# فهرست مطالب


**فصل ۱: دستورات پایه در LabVIEW و پروژه‌های کاربردی** ۱۵

۱-۱- مقدمه ۱۶

۱-۲- کاربرد LabVIEW ۱۶

۱-۳- ماژول‌های جانبی LabVIEW ۱۸

۱-۳-۱- ماژول بلادرنگ ۱۸

۱-۳-۲- ماژول کنترل، نظارت و ثبت اطلاعات ۱۹

۱-۳-۳- ماژول پایگاه داده ۲۰

۱-۳-۴- ماژول تولید گزارش ۲۰

۱-۳-۵- ماژول Touch Panel ۲۰

۱-۳-۶- ماژول منطق فازی و کنترل تناسبی ۲۱

۱-۳-۷- ماژول FPGA ۲۱

۱-۳-۸- ماژول ارتباط از طریق اینترنت ۲۲

۱-۳-۹- ماژول روباتیک ۲۲

۱-۳-۱۰- ماژول میکروکنترلر ARM ۲۳

۱-۳-۱۱- ماژول کنترل و شبیه‌سازی ۲۳

۱-۳-۱۲- ماژول پردازش سیگنال ۲۴

۱-۳-۱۳- ماژول مولتی سیم ۲۵

۱-۴- نصب نرم‌افزار ۲۵

۱-۵- کرک نمودن برنامه LabVIEW ۳۲

۱-۶- شروع کار با برنامه ۳۲

۱-۷- ذخیره نمودن برنامه ۳۴

۱-۸- پنجره Front Panel ۳۵

۱-۸-۱- پالت کنترل (Controls Palette) ۳۵

۱-۸-۱-۱- Numeric ۳۷

۱-۸-۱-۲- Boolean ۴۰

۱-۸-۱-۳- String and Path Controls and Indicators ۴۵

۱-۸-۱-۴- دکوراسیون (Decoration) ۵۳

۱-۸-۱-۵- ظروف (Containers) ۵۳

۱-۸-۲- پنجره Block Diagram ۵۴

۱-۸-۳- نوار ابزار بلوک دیاگرام (Tool Bar) ۵۷

۱-۸-۴- پالت ابزار ۵۸

۱-۸-۵- نوار ابزار Front Panel ۶۰

۱-۹- ساختار برنامه‌نویسی در LabVIEW ۶۳

۱-۹-۱- حلقه While ۶۴

۱-۹-۱-۱- ساختار تونل در While ۶۵

۱-۹-۱-۲- ساختار مشاهده خطا در While ۶۶

۱-۹-۲- حلقه For ۶۸

۱-۹-۲-۱- اضافه نمودن زمان‌بندی به حلقه‌ها ۶۹

۱-۹-۳- حلقه Case ۷۰

۱-۹-۴- ساختار Case با چند فریم ۷۱

۱-۹-۵- ثبات انتقال Shift Register ۷۳

۱-۹-۶- تعریف رخداد (Event) ۷۷

۱-۹-۷- بررسی تابع پیش آمد ۷۹

۱-۹-۸- ساختار ترتیبی ۸۱

۱-۹-۹- ند فرموله شده (Formula Node) ۸۳

۱-۹-۱۰- مقایسه‌گرها (Comparison) ۸۴

۱-۹-۱۱- زمان‌بندی (Timing) ۸۵

۱-۱۰- پنجره‌های ارتباط با کاربر ۸۶

۱-۱۱- آرایه‌ها ۸۷

۱-۱۱-۱- توابع آرایه ۸۸

۱-۱۱-۱-۱- اندازه آرایه (Array Size) ۸۹

۱-۱۱-۱-۲- مقدار دهی اولیه آرایه (Initialize Array) ۸۹

۱-۱۱-۱-۳- اندیس آرایه (Index Array) ۹۰
۱-۱۱-۱-۴- ساختن آرایه (Build Array) ۹۱
۱-۱۱-۱-۵- انتخاب زیر مجموعه‌ای از آرایه‌ها (Array Subset Function) ۹۲
۱-۱۱-۱-۶- چرخاندن آرایه یک بعدی (Rotate 1D Array) ۹۲
۱-۱۱-۱-۷- تقسیم یک آرایه یک بعدی (Split 1D Array) ۹۲
۱-۱۱-۱-۸- مرتب کردن آرایه (Sort 1D Array) ۹۳
۱-۱۱-۱-۹- تعیین ماکزیمم و مینیمم آرایه (Array Max & Min) ۹۳
۱-۱۱-۱-۱۰- تبدیل آرایه به کلاستر (Array To Cluster) ۹۳
۱-۱۱-۱-۱۱- تبدیل کلاستر به آرایه (Cluster To Array Function) ۹۳
۱-۱۱-۱-۱۲- آرایه دوبعدی ۹۴
۱-۱۱-۲- آرایه ماتریسی (Array Matrix) ۹۴
۱-۱۲- کلاسترها ۹۵
۱-۱۲-۱- توابع کلاستر ۹۶
۱-۱۲-۱-۱- تابع جداکننده داده‌ها (Unbundle & Unbundle by Name) ۹۶
۱-۱۲-۱-۲- تابع دسته بندی داده‌ها (Bundle & Bundle by Name) ۹۸
۱-۱۲-۱-۳- تبدیل تابع آرایه به کلاستر و برعکس ۹۸
۱-۱۳- نمایش داده‌ها ۹۹
۱-۱۳-۱- نمودار شکل موج ۱۰۰
۱-۱۳-۱-۱- گزینه‌های منوی مرتبط با نمودار شکل موج (WaveForm Chart) ۱۰۱
۱-۱۳-۲- گراف شکل موج ۱۰۳
۱-۱۳-۳- گزینه‌های مهم در منوهای گراف ۱۰۴
۱-۱۳-۴- گراف XY ۱۰۶
۱-۱۳-۵- گراف شکل موج دیجیتال ۱۰۷
۱-۱۳-۶- گراف شکل موج دیجیتال-آنالوگ ۱۰۸
۱-۱۳-۷- گراف شکل موج با ورودی دکارتی یا کارتزین ۱۰۹
۱-۱۳-۸- نمایش سه بعدی به کمک گراف‌های Surface و Parametric (3D) ۱۰۹
۱-۱۳-۹- نمایش سه بعدی دیتا به کمک 3D Pie Graph ۱۱۱
۱-۱۳-۱۰- نمایش سه بعدی دیتا به کمک 3D Contour Graph ۱۱۲
۱-۱۳-۱۱- نمایش سه بعدی دیتا به کمک 3D Comet Graph ۱۱۲
۱-۱۳-۱۲- نمایش سه بعدی به کمک 3D Bar Graph ۱۱۳
۱-۱۴- ایجاد زیر برنامه (Sub VI) ۱۱۳
۱-۱۵- ایجاد کتابخانه اختصاصی در LabVIEW ۱۱۷
۱-۱۶- متغیرهای محلی، سراسری و اشتراکی ۱۲۰
۱-۱۶-۱- متغیرهای محلی (Local Variable) ۱۲۱
۱-۱۶-۲- متغیرهای سراسری (Global Variable) ۱۲۲
۱-۱۶-۳- متغیرهای اشتراکی (Shared Variable) ۱۲۳
۱-۱۷- گره‌های خصوصیت (Property Node) ۱۲۶
۱-۱۸- تنظیم مشخصات VI ۱۲۸
۱-۱۹- تنظیم کار با Domain Account Manager و تنظیمات امنیتی در LabVIEW ۱۳۰
۱-۲۰- گزینه Show VI Hierarchy ۱۳۶
۱-۲۱- چاپ برنامه ۱۳۷
۱-۲۲- تولید و پروسس سیگنال توسط LabVIEW ۱۳۷
۱-۲۲-۱- توابع اولیه ضربه ، شیب و پله ۱۳۸
۱-۲۲-۲- تواتر Chrip خطی ۱۳۹
۱-۲۳- دستیابی به فایل‌ها و ذخیره اطلاعات در فضای هارد دیسک ۱۴۰
۱-۲۳-۱- استفاده از File I\Oها ۱۴۰
۱-۲۳-۲- کار با Databaseهای Microsoft Access ۱۴۵
۱-۲۳-۳- بهره‌برداری از دیتابیس‌های دیگر توسط LabVIEW ۱۴۸
۱-۲۴- ماژول Report Generation ۱۵۵
۱-۲۵- تحلیل طیفی در LabVIEW ۱۶۳
۱-۲۶- فیلترهای دیجیتال در LabVIEW ۱۷۰
۱-۲۷- مدولاسیون عرض پالس یا مدولاسیون مدت زمان پالس ۱۷۶
۱-۲۸- بالا بردن نمونه‌های یک سیگنال ۱۸۱

۱-۲۹- پایین آوردن تعداد نمونه‌های یک سیگنال ۱۸۳
۱-۳۰- کار با فایل های صوتی و ایجاد اصوات به کمک کارت صدای PC ۱۸۵
۱-۳۱- معرفی اجمالی نرم‌افزار Virtual Serial Ports Driver ۱۸۸
۱-۳۲- کار با پورت‌های PC ۱۹۱
۱-۳۲-۱- پورت Serial ۱۹۱
۱-۳۲-۲- VISA ۱۹۵
۱-۳۳- بررسی ارتباط اترنت یا شبکه TCP در LabVIEW ۱۹۷
۱-۳۳-۱- مثال ارتباط Server – Client بخش اول: Simple Data Server ۱۹۷
۱-۳۳-۲- فراهم آوردن شرایط تست عملی ارتباط TCP ۱۹۸
۱-۳۳-۳- مثال ارتباط Server – Client بخش دوم: Simple Data Client ۲۰۲
۱-۳۴- مثال ارتباط بررسی TCP Communicator - Active ۲۰۳
۱-۳۵- بررسی TCP Communicator – Passive ۲۰۵
۱-۳۶- اتوماسیون صنعتی و پروتکل Modbus RTU/ASCII ۲۰۷
۱-۳۶-۱- بررسی ساختار پروتکل Modbus ۲۰۷
۱-۳۶-۲- بررسی ساختار فریم در استاندارد Modbus ۲۱۰
۱-۳۶-۳- قدم به قدم با فریم های Modbus ASCII ۲۱۲
۱-۳۶-۴- معرفی اجمالی Modbus Plus ۲۱۶
۱-۳۶-۵- معرفی Modbus TCP ۲۱۶
۱-۳۶-۶- بررسی و تمرین Modbus با نرم‌افزار LabVIEW ۲۱۷
۱-۳۶-۶-۱- MB Serial Example Master.vi ۲۱۹
۱-۳۶-۶-۲- زیر برنامه MB master query ۲۲۳
۱-۳۶-۶-۳- MB Serial Example Slave.vi ۲۲۹
۱-۳۶-۶-۴- بررسی یک برنامه نمونه‌ای Modbus Ethernet Master در LabVIEW ۲۳۳
۱-۳۶-۶-۵- بررسی یک برنامه نمونه‌ای Ethernet Slave Modbus در LabVIEW ۲۳۵
۱-۳۷- استفاده از LabVIEW و S7\S5 OPC Server برای ارتباط با PLCهای زیمنسی ۲۳۷
۱-۳۷-۱- نصب و راه‌اندازی برنامه‌ی S7S5OPCServer ۲۳۷
۱-۳۷-۲- مثال کاربردی استفاده از LabVIEW به عنوان OPC Client ۲۴۴
۱-۳۷-۳- کار با بلوک های ارتباط با OPC Server در LabVIEW ۲۴۸
۱-۳۷-۴- نحوه پیدا کردن آدرس OPC Servers ۲۴۸
۱-۳۷-۵- استفاده از نرم‌افزار LabVIEW برای خواندن از PLC ۲۵۰
۱-۳۷-۶- استفاده از نرم‌افزار LabVIEW برای نوشتن در PLC ۲۵۱
۱-۳۸- ساخت DLL در Simulink و فراخوانی آن در LabVIEW ۲۵۲
۱-۳۹- گیجهای آنالوگ و استفاده از تابع Curve Fitting ۲۶۰
۱-۴۰- استفاده از تصاویر متحرک در LabVIEW ۲۶۲
۱-۴۱- دسترسی به فایل های اجرایی ۲۶۲
۱-۴۲- استفاده از Mouse و Keyboard ۲۶۳
۱-۴۳- بهره‌برداری از Registry ویندوز ۲۶۶
۱-۴۳-۱- شاخه‌های اصلی رجیستری ویندوز ۲۶۷
۱-۴۳-۲- مقادیر رجیستری ۲۶۷
۱-۴۴- ساخت فایل اجرایی یا EXE در LabVIEW ۲۶۹
۱-۴۵- ساخت فایل Setup یا Installer ۲۷۱

**فصل ۲: سیمولاتورها و سیستم‌های بلادرنگ ۲۸۱**

۲-۱- مقدمه ۲۸۲
۲-۲- علت استفاده از سیستم شبیه‌ساز یا سیمولاتور بلادرنگ ۲۸۲
۲-۳- تاریخچه سیستم‌های شبیه‌ساز بلادرنگ ۲۸۲
۲-۳-۱- Transient Network Analyzer (TNA) ۲۸۳
۲-۳-۲- سیمولاتور هیبرید (Analog and Hybrid Simulator) ۲۸۵
۲-۳-۳- Fully Digital off-line Simulator ۲۸۵
۲-۳-۴- سیمولاتورهای بلادرنگ (Real Time) ۲۸۵
۲-۴- علت استفاده از سیستم‌های بلادرنگ در شبیه‌سازی ۲۸۶
۲-۵- طراحی مبتنی بر مدل MBD (Model Based Design) ۲۸۸
۲-۶- پیاده سازی یک کنترلر بلادرنگ ۲۸۹
۲-۷- معرفی سیستم عامل‌های بلادرنگ ۲۹۰

۲-۸- مجموعه اصطلاحات فنی ۲۹۱
۲-۹- انتخاب گام زمانی مناسب در شبیه‌سازی ۲۹۲
۲-۱۰- مثال کاربردها و برنامه‌های بلادرنگ ۲۹۲
۲-۱۱- تفاوت سیستم عامل‌های بلادرنگ با سیستم عامل‌های عمومی ۲۹۳
۲-۱۲- معرفی سیمولاتورهای شناخته شده ۲۹۳
۲-۱۲-۱- سیمولاتور RTDS ۲۹۳
۲-۱۲-۲- سیمولاتور Opal RT ۲۹۸
۲-۱۲-۳- سیمولاتور SolvSim Power Station (SSPS) ۳۰۰
۲-۱۲-۴- سیمولاتورهای National Instrument ۳۰۲
۲-۱۲-۴-۱- معرفی اجمالی ماژول LabVIEW Real-Time ۳۰۲
۲-۱۲-۴-۲- اجزای سیستم بلادرنگ LabVIEW ۳۰۳
۲-۱۲-۴-۳- کامپیوتر Host ۳۰۳
۲-۱۲-۴-۴- نرم‌افزار LabVIEW ۳۰۴
۲-۱۲-۴-۵- موتور بلادرنگ یا RT Engine ۳۰۴
۲-۱۲-۴-۶- کامپیوتر بلادرنگ یا RT Target ۳۰۵
۲-۱۳- تفاوت عملکردی و ویژگی‌های خاص سیستم عامل‌ها ۳۰۸
۲-۱۴- تفاوت زمانی اجرای برنامه در محیط‌های مختلف ۳۰۹

**فصل ۳: Ardence RTX در محیط سیستم عامل Windows XP** **۳۱۱**

۳-۱- مقدمه ۳۱۲
۳-۲- دلایل رغبت به سیستم‌های کنترل تحت ویندوز ۳۱۲
۳-۳- به چه سخت‌افزاری یک سیستم بلادرنگ می‌گویند؟ ۳۱۴
۳-۴- معماری RTX ۳۱۴
۳-۵- کار با Real-Time HAL ۳۱۵
۳-۶- بررسی عملکرد و کارایی نرم‌افزار ۳۱۶
۳-۷- ارزیابی صحت عملکرد بلادرنگ در محیط برنامه تحت RTX ۳۱۷
۳-۸- ساخت برنامه‌های عملی قابل اجرا در RTX به کمک Visual Studio ۳۱۷
۳-۹- انجام تحلیل‌های چند پروسسوری به کمک CPU ۳۱۹
۳-۱۰- راه‌اندازی زیر مجموعه بلادرنگ RTX در محیط Windows XP ۳۱۹
۳-۱۱- نحوه شناسایی و نصب سخت‌افزار در RTX ۳۲۴
۳-۱۲- تست و راه‌اندازی یک نرم‌افزار در محیط بلادرنگ RTX ۳۲۸
۳-۱۳- کاربرد نظریه دستگاه مرجع در مدل‌سازی ماشین‌های الکتریکی ۳۳۰
۳-۱۴- تبدیلات سه فاز ۳۳۰
۳-۱۴-۱- تبدیل کلارک ۳۳۱
۳-۱۴-۲- تبدیل پارک ۳۳۱
۳-۱۴-۳- دستگاه‌های مرجع متعارف ۳۳۳
۳-۱۵- مثال کاربردی: پیاده‌سازی ماشین آسنکرون با روتور قفس سنجابی در دستگاه مرجع اختیاری ۳۳۵
۳-۱۶- مدل‌سازی ماشین القایی سه فاز ۳۳۵
۳-۱۶-۱- معادلات اساسی در شبیه‌سازی ماشین القایی ۳۳۶
۳-۱۷- پیاده‌سازی مدل ماشین القایی سه فاز در LabVIEW ۳۴۱
۳-۱۸- بررسی نحوه محاسبه دیگر مقادیر مکانیکی و الکتریکی ۳۴۸
۳-۱۸-۱- گشتاور ۳۴۸
۳-۱۸-۲- سرعت ۳۴۹
۳-۱۸-۳- باکس $I_{dq}$ ۳۵۰
۳-۱۸-۴- تبدیل دستگاه‌ها (باکس $\mathbf{f_{qs}} \rightarrow \mathbf{f_{abc}}$) ۳۵۰
۳-۱۹- برنامه‌نویسی برای FPGA روی کارت NI PCI-7831R ۳۵۱
۳-۱۹-۱- FPGA چیست؟ ۳۵۲
۳-۱۹-۲- کاربرد FPGA ۳۵۲
۳-۱۹-۳- برنامه‌نویسی و طراحی با FPGA ۳۵۲
۳-۱۹-۴- کار با FPGA در LabVIEW ۳۵۳
۳-۱۹-۵- مفهوم Resolution در کارت‌های آنالوگ ۳۵۵
۳-۱۹-۶- برنامه‌نویسی برای FPGA کارت ۳۵۶
۳-۱۹-۷- برنامه‌نویسی سمت RTX ۳۵۸
۳-۱۹-۸- اجرا کردن برنامه و شبیه‌سازی ماشین آسنکرون ۳۵۸

**فصل ۴: پیکربندی نرم و سخت‌افزاری سیستم بلادرنگ PXI** ۳۶۵
۴-۱- مقدمه ۳۶۶
۴-۲- مزایای PXI Express ۳۶۶
۴-۳- سخت‌افزار پروسسور NI PXIe-8133 ۳۶۶
۴-۳-۱- توصیف ۳۶۶
۴-۳-۲- مروری بر عملکرد NI PXIe-8133 ۳۶۷
۴-۳-۳- نرم‌افزار های قابل نصب روی پروسسور NI PXIe-8133 ۳۶۸
۴-۴- نصب و پیکربندی سیستم PXIe ۳۶۹
۴-۵- ارتباط با فیلد ۳۶۹
۴-۶- نصب NIMAX ۳۷۰
۴-۶-۱- تنظیمات IP در NI MAX ۳۷۴
۴-۶-۲- مشاهده و تغییر مشخصات تجهیزات نصب شده بر روی شاسی ۳۷۶
۴-۶-۳- تنظیمات زمان و پروتکل‌های همزمان سازی تجهیزات ۳۷۷
۴-۶-۴- نصب نرم‌افزار های مرتبط بر روی سیستم PXI ۳۷۹
۴-۶-۵- تعریف سطح دسترسی برای کاربران ۳۸۲
۴-۷- اطلاعات بخش‌های مختلف پروسسور PXIe-8133 ۳۸۶
۴-۷-۱- پورت DVI ۳۸۷
۴-۷-۲- پورت COM1 ۳۸۸
۴-۷-۳- پورت Ethernet ۳۸۹
۴-۷-۴- پورت Parallel ۳۹۰
۴-۷-۵- پورت USB ۳۹۱
۴-۷-۶- Trigger ۳۹۲
۴-۷-۷- پورت GPIB (IEEE 488.2) ۳۹۳
۴-۷-۷-۱- توابعی برای GPIB ۳۹۴
۴-۷-۸- اسلات ExpressCard/34 ۳۹۵
۴-۸- ویژگی‌های پنل PXI ۳۹۵
۴-۹- کیس NI PXIe-1062Q و کارت‌های PXI ارتباطی و DAQ ۳۹۶
۴-۹-۱- نصب chassis در نرم‌افزار MAX ۳۹۷
۴-۹-۲- کارت NI_PXI_7833R ۳۹۹
۴-۹-۳- کارت NI_PXI_8430 (RS232) ۴۰۰
۴-۹-۴- کارت NI_PXI_8431 (RS485/RS422) ۴۰۱
۴-۹-۵- کارت NI_PXI_8231 ۴۰۱
۴-۱۰- استفاده از LabVIEW برای برنامه‌نویسی PXI ۴۰۲
۴-۱۰-۱- استفاده از درگاه های سریال RS232 و RS485 ۴۰۳
۴-۱۰-۲- راه‌اندازی Modbus Server بر روی PXI ۴۰۵
**فصل ۵: ارتباط LabVIEW2012 با نرم‌افزار سیمولینک MATLAB با استفاده از Simulation Interface Toolkit** ۴۱۱
۵-۱- مقدمه ۴۱۲
۵-۲- نصب برنامه‌های مرتبط ۴۱۲
۵-۳- ایجاد رابط کاربری LabVIEW برای ارتباط مستقیم با مدل درون سیمولینک ۴۱۴
۵-۴- تبدیل یک مدل سیمولینک به فایل DLL ۴۱۵
۵-۵- ایجاد یک مدل سیمولینک در MATLAB ۴۱۶
۵-۶- تنظیمات پارامترهای شبیه‌سازی در MATLAB ۴۱۸
۵-۷- ایجاد رابط کاربری در LabVIEW ۴۱۸
۵-۸- اختصاص مدل MATLAB به رابط کاربری LabVIEW ۴۱۹
۵-۹- کتابخانه ارتباط غیر مستقیم یا ساخت DLL ۴۲۴
۵-۱۰- کتابخانه دینامیک یا DLL چیست؟ ۴۲۴
۵-۱۱- تبدیل مدل سیمولینک به مدل DLL ۴۲۵
۵-۱۲- کاربرد نحوه باز خوانی فایل DLL در محیط LabVIEW ۴۲۷
۵-۱۳- برقراری ارتباط LabVIEW با مدل DLL ۴۲۸
**فصل ۶: شبیه‌سازی ماشین سنکرون، سیستم Exciter استاتیک و شبکه قدرت در سیمولینک** ۴۳۱
۶-۱- مقدمه ۴۳۲
۶-۲- نظریه ماشین‌های سنکرون و معادلات حاکم بر آن ۴۳۲
۶-۳- معادلات ولتاژ در متغیرهای دستگاه مرجع روتور- معادلات پارک ۴۳۳

۶-۴- معادله گشتاور بر حسب شارهای پیوندی ۴۳۵
۶-۵- محاسبه سرعت لغزشی ۴۳۶
۶-۶- پیاده‌سازی معادلات در MATLAB ۴۳۷
۶-۷- پیاده‌سازی روابط ریاضی ماشین و شبکه در نرم‌افزار سیمولینک ۴۳۸
۶-۸- پیاده‌سازی معادلات ماشین سنکرون ۴۳۹
۶-۸-۱- ورودی‌ها ۴۳۹
۶-۸-۲- مبدل‌ها ۴۴۰
۶-۸-۳- روابط و معادلات ریاضی ماشین ۴۴۱
۶-۸-۴- خروجی‌ها ۴۴۳
۶-۹- پیاده‌سازی شبکه در نرم‌افزار سیمولینک ۴۴۴
۶-۱۰- پیاده‌سازی سیستم تحریک و پایدارساز قدرت ۴۴۵
۶-۱۰-۱- سیستم تحریک ۴۴۶
۶-۱۰-۱-۱- مبدل ولتاژ و مدار جبران بار ۴۴۶
۶-۱۰-۱-۲- تنظیم کننده ۴۴۶
۶-۱۰-۱-۳- تحریک گر DC ۴۴۷
۶-۱۰-۲- پایدارساز سیستم قدرت ۴۴۷
۶-۱۱- اجرای سناریوهای مختلف در MATLAB ۴۵۱
۶-۱۱-۱- سناریو ۱ ۴۵۱
۶-۱۱-۲- سناریو ۲ ۴۵۳
۶-۱۱-۳- سناریو ۳ ۴۵۵
۶-۱۲- پیاده‌سازی سیستم در LabVIEW ۴۵۷
۶-۱۲-۱- اجرای برنامه مستقل از محیط RT و در واسط کاربری درون PC ۴۷۰
۶-۱۲-۲- بخش نمونه‌برداری و ذخیره اطلاعات توسط TDMS ۴۷۴
۶-۱۳- ایجاد ارتباط با محیط واقعی ۴۷۶
۶-۱۳-۱- معرفی کارت‌های موجود روی سخت‌افزار PXI به LabVIEW ۴۷۷
**فصل ۷: آماده‌سازی سیستم PC Real-Time ۴۹۱**
۷-۱- مقدمه ۴۹۲
۷-۲- پیاده سازی سیستم LabVIEW Real-Time 2015 در VirtualBox ۴۹۲
۷-۲-۱- مرحله اول، ساخت درایو PC Desktop Utility ۴۹۲
۷-۲-۲- مرحله دوم، مشاهده شماره disk حافظه فلش ۴۹۳
۷-۲-۳- مرحله سوم، ساخت فایل حافظه سخت مجازی از درایو حافظه فلش ۴۹۴
۷-۲-۴- مرحله چهارم، ساخت یک ماشین مجازی ۴۹۵
۷-۲-۵- مرحله پنجم، اعمال تنظیمات ضروری بر روی Virtual Machine ۴۹۷
۷-۲-۶- مرحله ششم، اجرای Virtual Machine ۵۰۰
۷-۳- مرحله هفتم، برقراری ارتباط Ethernet با Target ۵۰۲
۷-۳-۱- مرحله هشتم، نصب LabVIEW Real-Time از طریق NI MAX ۵۰۵
۷-۴- تبدیل یک کامپیوتر معمولی به Target بلادرنگ LabVIEW ۵۰۷
۷-۴-۱- نحوه آزمون یک PC برای بررسی امکان استفاده به عنوان Target بلادرنگ ۵۰۷
۷-۴-۲- الزامات ساختاری یک کامپیوتر معمولی برای تبدیل شدن به یک Target بلادرنگ ۵۰۹
۷-۴-۳- نرم‌افزارهای مورد نیاز برای پیاده سازی یک Target بلادرنگ ۵۰۹
۷-۴-۴- سخت‌افزارهای مورد نیاز برای پیاده سازی یک Target بلادرنگ ۵۰۹
۷-۴-۴-۱- پروسسورها ۵۰۹
۷-۴-۴-۲- کارت‌های شبکه ۵۱۰
۷-۴-۴-۳- حافظه‌های سخت ۵۱۳
۷-۴-۴-۴- حافظه RAM ۵۱۳
۷-۴-۴-۵- بورد اصلی یا مادربورد ۵۱۴
۷-۴-۴-۶- کارت گرافیکی ۵۱۴
۷-۵- پیاده سازی LabVIEW Real-Time 2015 ۵۱۵
۷-۵-۱- مثال کاربردی اول، ارتباط اترنت بلادرنگ ۵۱۵
۷-۵-۱-۱- تنظیمات ارتباط شبکه بلادرنگ ۵۱۶
۷-۵-۱-۲- بررسی برنامه سمت کامپیوتر RT Target ۵۱۷
۷-۵-۱-۳- بررسی برنامه سمت کامپیوتر HOST ۵۲۱
۷-۵-۲- مثال کاربردی دوم، شبیه‌سازی موتور القایی در PC Real-Time ۵۲۳

۷-۵-۲-۱- نحوه کنترل یک موتور الکتریکی ..... ۵۲۵
مراجع ..... ۵۳۶

## ۱-۱- مقدمه فصل اول

اگر علاقه دارید بدون تایپ و تنها به کمک موس برنامه‌نویسی کنید، نرم‌افزار LabVIEW بهترین گزینه برای شما می‌باشد. کاملاً مشهود است که تسریع در عملکرد مهندسین و دانشمندان در مسیر حرکت از ایده تا واقعیت از جمله مزیت‌های برنامه‌نویسی با زبان گرافیکی LabVIEW است. به طورکلی نقاط قوت این برنامه شامل طراحی آسان واسط کاربری، سرعت در اقدام به جمع‌آوری و تحلیل اطلاعات می‌باشد. همچنین با معرفی ابزارهای سودمند نرم‌افزاری جهت تولید سیگنال و معرفی کارت‌های فوق‌العاده برای ارتباط با دنیای بیرون، این نرم‌افزار گزینه مناسبی برای اهداف پژوهشی دانشگاه‌ها گردیده است. در کنار تمام این موارد امکان تحلیل دیتای یک سیستم صنعتی، ذخیره‌سازی دیتا در دیتابیس‌های استاندارد، کدنویسی C در درون برنامه و کار در سیستم عامل‌های متنوع، این برنامه را محبوب صنعت نیز نموده است. در این فصل، با معرفی اجزای گوناگون نرم‌افزار LabVIEW و ماژول‌های آن، سعی شده تا بخش‌های مختلف برنامه توضیح داده شود. همچنین برخی توابع تحلیل سیگنال، کار با توابع کنترلی، ارتباط با دیتابیس‌های رایج و پیاده‌سازی پروتکل‌های مرسوم ارتباطی از دیگر موارد مورد بحث این فصل می‌باشند. برای درک بهتر موضوعات، مثال‌های متنوع و کاربردی آماده و در DVD همراه کتاب قرار داده شده است.

## ۲-۱- مقدمه فصل دوم

استفاده از سیستم‌های بلادرنگ هر روز در حال گسترش می‌باشد. به عنوان مثال امروزه در سیستم‌های کنترل ترافیک هوایی، سیستم‌های مخابراتی و رسانه‌ایی، صنعت خودرو، فرآیندهای پیچیده کنترلی صنایع حساس و شبیه‌سازها و امولاتورهای بلادرنگ از سیستم‌های Real-Time استفاده می‌شود. سیستم‌های بلادرنگ می‌توانند پروسسورها یا کامپیوترهایی باشند که وظیفه نظارت، پاسخ و یا کنترل یک سیستم را بر عهده دارند. سیستم تحت نظارت نیز به کمک سنسورها، محرک‌ها و یا دیگر رابط‌های کاربری با پروسسور یا کامپیوتر در ارتباط است. نکته مهم در این سیستم‌ها دقت و قطعیت در زمان اجرای برنامه و قابل پیش‌بینی بودن عملکرد آنها می‌باشد.

در این فصل ابتداً به تاریخچه سیستم‌ها و سیمولاتورهای بلادرنگ اشاره می‌کنیم و با ارائه مثال به برخی سیستم‌های شناخته شده اشاره‌ایی خواهیم داشت. سپس با ذکر دلیل به تحلیل زوایای لزوم پیاده‌سازی سیستم Real-Time می‌پردازیم. در ادامه با شرح خصوصیات سیستم‌های بلادرنگ به تفاوت‌های آنها با سیستم عامل‌های جنرال اشاره می‌کنیم. در پایان نیز نگاهی اجمالی به سیستم‌های Real-Time شرکت NI خواهیم داشت.

## ۳-۱- مقدمه فصل سوم

راحتی، تجربه و سرعت بالای ما در کار با سیستم عامل Windows سبب شده است که با وجود آگاهی از مشکلات این سیستم‌عامل، همچنان به کار با آن حتی در پروسه‌های حساس علاقه‌مند باشیم. در این فصل با معرفی یک زیرسیستم بلادرنگ به نام RTX این امکان فراهم آمده است تا کاربر در محیط ویندوز XP بتواند پروسه‌های حساس صنعتی را کنترل نموده و یا به ساخت سیمولاتورهای پیچیده صنعتی بپردازد. در ضمن برای سهولت در امر آموزش و تجربه عملی کار، به کمک VirtualBox، شرایط تست و کار با LabVIEW Real-Time RTX Module فراهم شده است. در پایان به صورت پایه ای بعد از شرح معادلات ماشین القایی قفس سنجابی، این ماشین را در محیط LabVIEW پیاده‌سازی و برای ارتباط با دنیای بیرون آماده می‌کنیم.

## ۴-۱- مقدمه فصل چهارم

تکنولوژی و سخت‌افزار PXI[1] (که در حقیقت توسعه یافته باس PCI برای کاربردهای سیستم کنترل و ابزار دقیق می‌باشد)، پس از انجام موفقیت آمیز تست‌های مختلف، به سرعت جای خود را در سیستم‌های کنترل و اندازه‌گیری باز کرد و از سال ۱۹۹۸ به صورت رسمی به بازار عرضه شد. یکی از المان‌های مهم در مقبول واقع شدن PXI، استفاده از PCI در بک پلین ارتباطی[2] می‌باشد. این مشابه همان سیاستی است که سازندگان کامپیوترهای صنعتی با معرفی PCI Express در کنار PCI دنبال کردند تا به راحتی PCI Express را تجاری نمایند. با در نظر گرفتن مزایای تکنولوژی PCI_Express در بک پلین[3]، و دنباله‌روی از آن تکنیک، PXI_Express توانست تا پهنای باند قابل استفاده در PXI را از 132MB/s به 8GB/s، یعنی تا ۶۰ برابر بهبود ببخشد.

در این فصل سعی شده تا با معرفی معماری داخلی سیستم PXI دید مناسبی به علاقه‌مندان در راستای شناخت این‌گونه تجهیزات داده شود و به صورت عملی پیکربندی اجزای یک سیستم بلادرنگ شرکت National Instrument که در کیس NI PXIe-1062Q جمع‌آوری شده است، بررسی و به کمک نرم‌افزار NI- Max پیاده‌سازی گردد. اجزای این سیستم بلادرنگ شامل CPUی NI PXIe-8133، به همراه دو عدد کارت DAQ، کارت سریال RS232، RS485 و اترنت از محصولات NI می‌باشد. در پایان نیز با ارائه توضیحات اجمالی نحوه برنامه‌نویسی و استفاده از LabVIEW برای استفاده عملی از سیستم PXI در پروژه‌های پژوهشی و صنعتی توضیح داده شده است. در ضمن با ارائه چند مثال عملی نحوه بکارگیری پورت‌های آن سیستم چه به صورت عمومی و چه به صورت ساختار یافته، در قالب پروتکل استاندارد Modbus بیان شده است.

1. PCI eXtensions for Instrumentation
2. Communication Backplane
3. Backplane

## ۵-۱- مقدمه فصل پنجم

ابزار رابط شبیه‌سازی یا (Simulation Interface Toolkit) که به اختصار SIT نامیده می‌شود، به منظور ایجاد یک رابط کاربری برای ارتباط La-VIEW2012 با سیمولینک استفاده می‌شود. این رابط به دو روش با نرم‌افزار سیمولینک ارتباط برقرار می‌نماید.

در روش اول با افزودن یک ابزار جانبی در ®MathWorks MATLAB امکان ارتباط زنده و دینامیکی این نرم‌افزار با LabVIEW2012 را برقرار می‌کند. در این وضعیت مدل‌ها و کدهای درون سیمولینک به صورت پویا قابل اجرا از LabVIEW2012، کنترل و تبادل دیتا با آن خواهند بود.

در روش دوم با تبدیل مدل سیمولینک به یک فایل DLL (Dynamic Link Library) این ارتباط صورت می‌پذیرد. در این حالت دیگر نیازی به اجرای MATLAB و ابزار سیمولینک نمی‌باشد. تبدیل مدل به DLL در درون سیمولینک و به کمک قسمت شبیه‌سازی بلادرنگ (Real Time) و ابزار SIT افزوده شده در آن صورت می‌گیرد.

توجه نمایید که با ترکیب توانایی‌های سیمولینک و LabVIEW2012 و با استفاده از ابزار SIT کاربر می‌تواند مدل‌های موجود در کتابخانه سیمولینک را به نرم‌افزار LabVIEW2012 وارد نماید و بیشتر امکانات پیشرفته سیمولینک، مثال‌های موجود و حتی دیگر کتابخانه‌های آن همچون SimPowe-Systems را در اختیار بگیرد.

## ۶-۱- مقدمه فصل ششم

طبق یک تعریف مدل یک سیستم عبارتست از معادلات ریاضی، ابزار و حتی تجهیزاتی که از آن برای کسب اطلاعات از یک سیستم، بدون بکارگیری مستقیم از آن استفاده می‌شود. اصولاً برای آگاهی از رفتار سیستم در همان لحظه و یا زمان‌های آینده و حتی شبیه‌سازی سیستم‌های واقعی، بهره‌برداری از مدل‌های دینامیکی می‌توانند بسیار موثر باشند. در علوم مهندسی، مدل‌ها همواره برای طراحی یا بهبود عملکرد تجهیزات، فرآیندهای جدید، تحلیل و تنظیم فرآیندهای موجود مورد استفاده قرار می‌گیرند. از ابتدا در کاربردهای صنعتی روش تبدیل مدل به سیستم‌های معادلات دیفرانسیل معمولی یا ODE؛ سپس نوشتن کد برای انتگرال‌گیری از معادلات دیفرانسیل، به دلیل سازگاری بالا با معماری پروسسورها از جمله محبوبترین روش‌ها بوده است.

برای استفاده عملی و تجاری از مدل‌ها، همچنین ساخت دستگاه‌های صنعتی یا سیستم‌های پیچیده و پرهزینه، شبیه‌سازی سخت‌افزار در حلقه یا HIL اولین قدم می‌باشد. امولاتورهای HIL یکی از متداول‌ترین روش‌هایی هستند که برای توسعه و تست عملکرد سیستم‌های پیچیده بصورت تحلیل عملکرد در زمان واقعی قبل از ساخت سیستم حقیقی مورد توجه قرار می‌گیرند. این فرآیند به عنوان یک روش مقرون به صرفه در طراحی، توسعه، بهینه‌سازی و عیب‌یابی در سیستم‌های صنعتی بکار می‌رود.

ژنراتور سنکرون تاریخچه‌ای بیش از صد سال دارد و اولین تحولات ژنراتور سنکرون در دهه ۸۰ قرن نوزدهم رخ داده است. توجه داشته باشید که ماشین سنکرون همواره یکی از مهمترین عناصر شبکه قدرت بوده و نقش کلیدی در تولید انرژی الکتریکی و کاربردهای خاص دیگر ایفاء کرده است. در این فصل در ابتدا به بررسی نظریه ماشین‌های سنکرون پرداخته شده است. سپس شبیه‌سازی کامپیوتری ژنراتور سنکرون و دیگر ادوات ضروری جهت تولید توان، در محیط سیمولینک صورت گرفته است. در ادامه به کمک آنچه در فصل قبل آموختیم، با ساخت یک کتابخانه دینامیکی، مدل را به LabVIEW انتقال داده و در نهایت با تفکیک سیستم Exciter و پیاده‌سازی مدل ریاضی-کنترلی IEEE Type, ST1 در EZDSP320F2812 اقدام به راه‌اندازی یک سیستم HIL شده است. شایان ذکر است که سرعت اجرای حلقه اصلی امولاتور برابر با ۳۰۰ میکروثانیه می‌باشد.

## ۷-۱- مقدمه فصل هفتم

معمولاً سخت‌افزارهای ساخت شرکت National Instrument قیمت بالایی دارند و برای اهداف آموزشی و استفاده در سیستم‌های کنترل کوچک و آزمایشگاه‌های کنترل و ابزار دقیق شرکت‌های کوچک مناسب نمی‌باشند. البته بسیاری از کامپیوترهای غیرصنعتی می‌توانند به عنوان سیستم کنترل و یا سیستم تست با قطعیت قابل اطمینان اجرای حلقه مورد استفاده قرار گیرند. ترکیب این کامپیوترهای تجاری با کارت‌های DAQ شرکت NI و سیستم عامل بلادرنگ Pharlap، به راحتی یک Target بلادرنگ اقتصادی را تشکیل می‌دهند که برای کاربردهای غیرحساس و شبیه‌سازها ایده‌آل می‌باشند. برنامه‌ها و امکانات قابل استفاده در این سیستم‌ها کاملاً مشابه سیستم گران قیمت PXI است.

در این فصل در ابتدا سعی شده تا با پیاده‌سازی LabVIEW Real-Time در VirtualBox به ساده‌ترین روش ممکن امکان آموزش و کار با سیستم‌های بلادرنگ شرکت LabVIEW فراهم گردد. در ادامه نحوه تبدیل کردن کامپیوترهای تجاری موجود در بازار به Target بلادرنگ توضیح داده می‌شود که شرکت NI از این محصولات با نام PC RIO یاد می‌کند. در پایان نیز به دو مثال می پردازیم. اول: شبکه Ethernet بلادرنگ، دوم: پیاده‌سازی ماشین القایی سه فاز در محیطی شبیه به سیمولینک با نام Control Design and Simulation Module. به کمک این دو مثال سعی می‌کنیم تا با جزئیات به بررسی نحوه پیاده‌سازی و برنامه‌نویسی در سیستم‌های بلادرنگ تحت PC بپردازیم.