

استفاده از نرم‌افزار LabView برای ارتباط با دستگاه‌های سیستم PAC

یکی از ابزارهای قدرتمند جهت برقراری ارتباط بین کامپیوتر و دستگاه‌های سیستم PAC، استفاده از نرم‌افزار LabView می‌باشد. برای آماده‌سازی این نرم‌افزار جهت ارتباط با دستگاه‌های سیستم PAC، یک سری توابع (VI) تهیه شده است که در اختیار کاربران قرار می‌گیرد. در این Document، نحوه استفاده از این توابع در نرم‌افزار LabView توضیح داده می‌شود.

معرفی:

مجموعه توابع اصلی برای برقراری ارتباط بین نرم‌افزار LabView و دستگاه‌های سیستم PAC در یک Library با عنوان PACdriver.llb قرار گرفته‌اند. این توابع عبارتند از:

- Initial.vi •
- Close Serial Driver.vi •
- ReadPAC.vi •
- WritePAC.vi •
- StrToFloat.vi •
- FloatToStr.vi •
- Slave.vi •

علاوه بر توابع اصلی عنوان شده در بالا، چند تابع دیگر نیز برای استفاده در موارد خاص تهیه شده است که عبارتند از:

- AI.vi •
- Bus Inspector.vi •
- IOSample.vi •

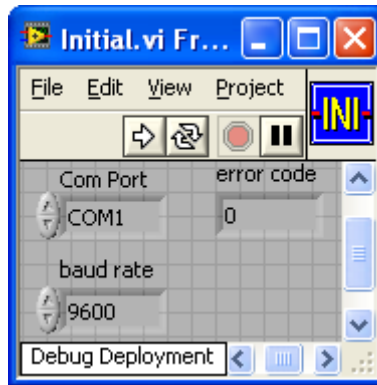
در ادامه نحوه استفاده از توابع فوق شرح داده می‌شود.

استفاده از توابع اصلی ارتباط با دستگاه‌های سیستم PAC

تابع Initial.vi:

از این تابع برای Initial نمودن پورت کامپیوتر جهت ارتباط با شبکه RS-485 استفاده می‌شود.

استفاده به صورت مستقیم:

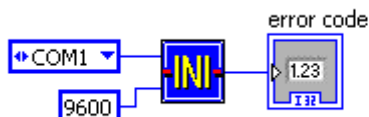


- در بخش Com Port، پورتی از کامپیوتر را که با استفاده از آن با شبکه RS-485 ارتباط برقرار می‌شود، انتخاب می‌کنیم.
- در بخش baud rate، سرعت انتقال اطلاعات بر روی شبکه را وارد می‌کنیم.
- با اجرای تابع، خطای احتمالی در بخش error code قرار می‌گیرد. در صورتی که پورت بدون خطا Initial شود، مقدار 0 در این بخش قرار می‌گیرد.

استفاده به عنوان SubVI:



مثال:

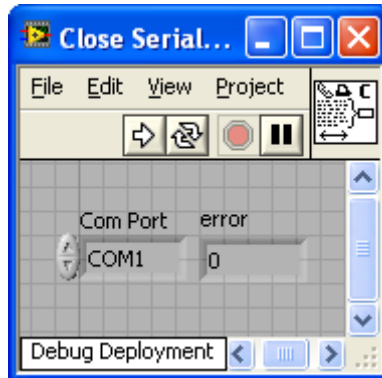


توجه: در صورتی که برای ارتباط بین کامپیوتر و شبکه RS-485 از مبدل RS-232 به RS-485 استفاده می‌کنید، برای ورودی پورت این تابع، COM1 را انتخاب نمایید. در صورتی که از مبدل USB به RS-485 استفاده می‌کنید، برای اطلاع از پورت صحیح، به بخش Device Manager/Ports در کامپیوتر خود مراجعه کنید.

تابع Close Serial Driver.vi

از این تابع برای بستن پورت سریالی که کامپیوتر از طریق آن با شبکه RS-485 ارتباط برقرار می‌کند، استفاده می‌شود.

استفاده به صورت مستقیم:

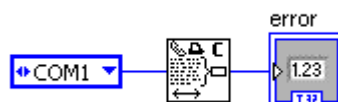


- در بخش Com Port، پورتهای از کامپیوتر را که با استفاده از آن با شبکه RS-485 ارتباط برقرار می‌شود، انتخاب می‌کنیم.
- با اجرای تابع، خطای احتمالی در بخش error code قرار می‌گیرد. در صورتی که پورت بدون خطا بسته (Close) شود، مقدار 0 در این بخش قرار می‌گیرد.

استفاده به عنوان SubVI



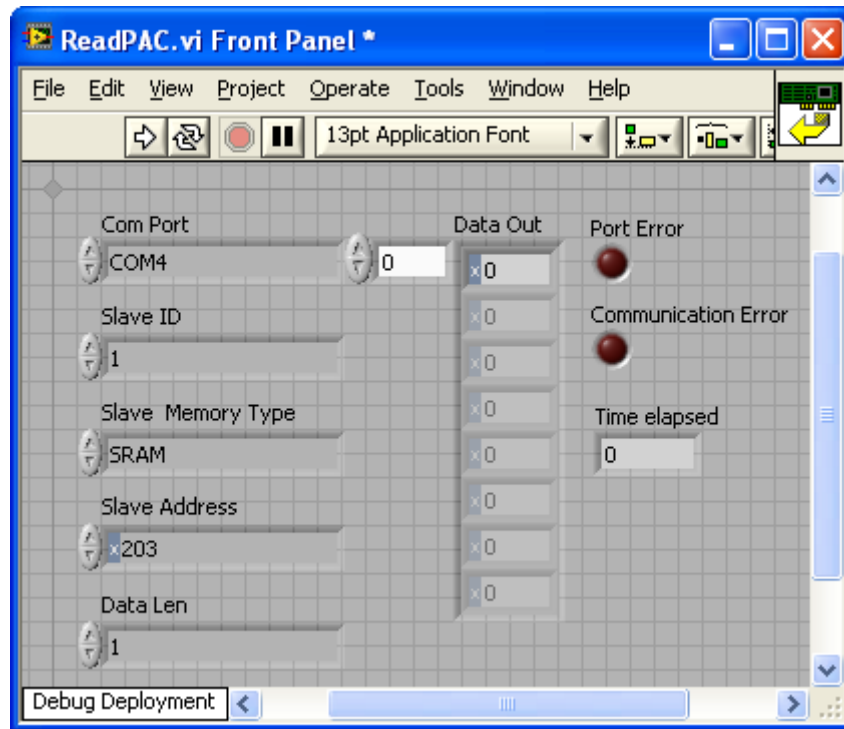
مثال:



تابع ReadPAC.vi

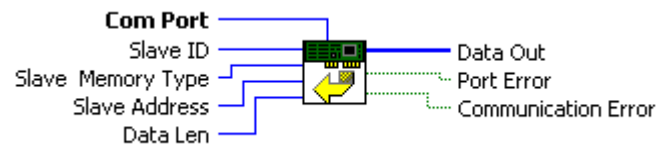
از این تابع برای خواندن تعداد مشخصی بایت داده از حافظه دستگاه PAC استفاده می‌شود.

استفاده به صورت مستقیم:

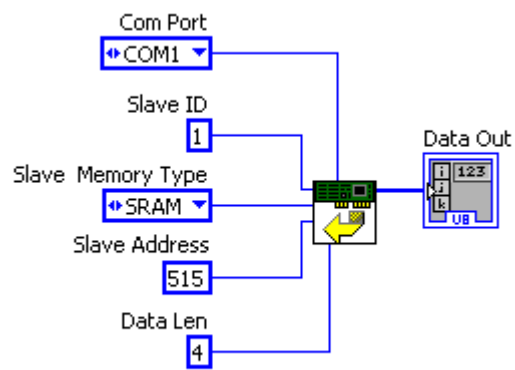


- در بخش Com Port، پورتی از کامپیوتر را که از آن برای ارتباط با شبکه RS-485 استفاده می‌شود، انتخاب می‌کنیم. لازم به ذکر است که پورت انتخاب شده باید قبلاً با استفاده از تابع «Initial.vi»، Initial شده باشد.
- در بخش Slave ID، شماره ID دستگاه PAC که می‌خواهیم با آن ارتباط برقرار کنیم را وارد می‌کنیم.
- در بخش Slave Memory Type، نوع حافظه‌ای از دستگاه PAC (SRAM, EEPROM)، که می‌خواهیم داده را از آن بخوانیم، انتخاب می‌کنیم.
- در بخش Slave Address، آدرس بایتی از حافظه دستگاه PAC که می‌خواهیم داده‌ها را از آن بخوانیم، تعیین می‌کنیم.
- در بخش Data Len تعیین می‌کنیم که چه تعداد بایت از حافظه را می‌خواهیم بخوانیم.
- با Run کردن تابع، داده مورد نظر از حافظه دستگاه PAC خوانده شده و در بخش Data Out قرار می‌گیرد.
- در صورتی که در حین ارتباط با دستگاه خطایی در انتقال یا پورت کامپیوتر پیش بیاید، LED مربوط به خطای مربوطه روشن می‌شود.
- در بخش Time elapsed مقدار زمانی که خواندن داده‌های تعیین شده طول کشیده است، بر حسب میلی‌ثانیه نشان داده می‌شود.

استفاده به عنوان SubVI



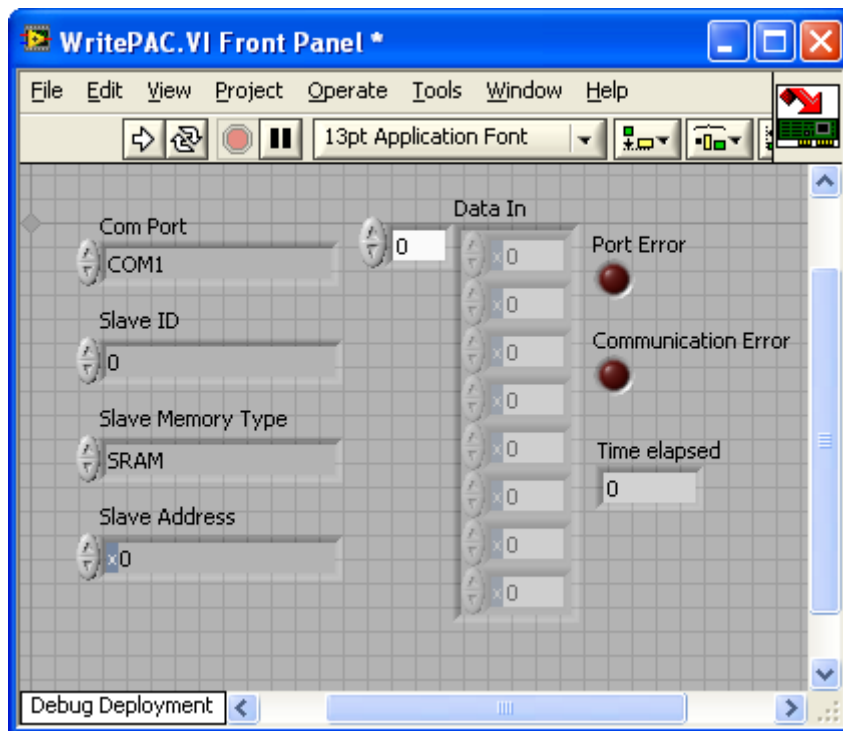
مثال:



تابع WritePAC.vi:

از این تابع برای نوشتن تعداد مشخصی بایت داده در حافظه دستگاه PAC استفاده می‌شود.

استفاده به صورت مستقیم:



- در بخش Com Port، پورتی از کامپیوتر را که از آن برای ارتباط با شبکه RS-485 استفاده می‌شود، انتخاب می‌کنیم. لازم به ذکر است که پورت انتخاب شده باید قبلاً با استفاده از تابع «Initial.vi»، Initial شده باشد.

- در بخش Slave ID، شماره ID دستگاه PAC که می‌خواهیم با آن ارتباط برقرار کنیم را وارد می‌کنیم.

- در بخش Slave Memory Type، نوع حافظه‌ای از دستگاه PAC (SRAM, EEPROM)، که می‌خواهیم داده را در آن بنویسیم، انتخاب می‌کنیم.

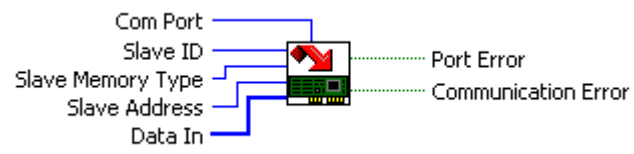
- در بخش Slave Address، آدرس بایتی از حافظه دستگاه PAC که می‌خواهیم داده‌ها را در آن بنویسیم، تعیین می‌کنیم.

- در بخش Data In، داده‌هایی را که می‌خواهیم در حافظه دستگاه PAC نوشته شوند وارد می‌کنیم. با Run کردن تابع، ارتباط برقرار داده شده و داده‌های وارد شده، در حافظه دستگاه PAC نوشته می‌شوند.

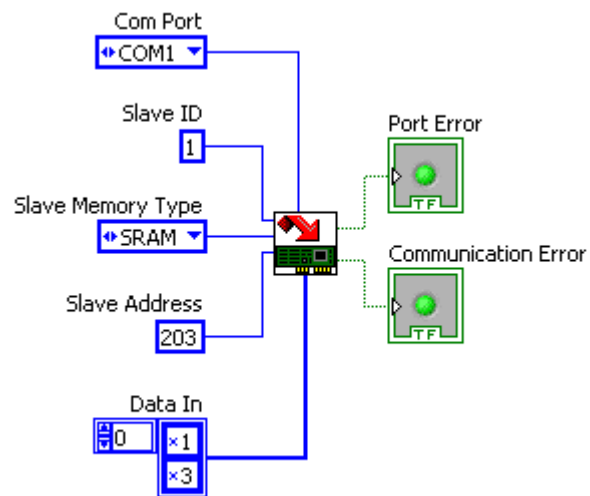
- در صورتی که در حین ارتباط با دستگاه خطایی در انتقال یا پورت کامپیوتر پیش بیاید، LED مربوط به خطای مربوطه روشن می‌شود.

- در بخش Time elapsed مقدار زمانی که نوشتن داده‌های وارد شده طول کشیده است، بر حسب میلی‌ثانیه نشان داده می‌شود.

استفاده به عنوان SubVI



مثال:



تابع StrToFloat:

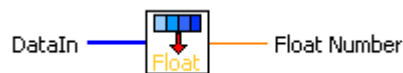
این تابع زمانی استفاده می‌شود که بخواهیم از حافظه یک دستگاه PAC، مقدار یک داده از جنس Float را به نرم‌افزار LabView منتقل کنیم. نحوه خواندن یک داده Float از روی دستگاه PAC به شرح زیر است.

استفاده به عنوان SubVI:

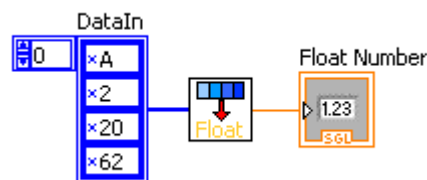
می‌دانیم که داده Float در حافظه دستگاه PAC از ۴ بایت تشکیل شده است. برای انتقال مقدار یک داده Float از دستگاه PAC نمی‌توانیم این ۴ بایت را مستقیماً به صورت Float بخوانیم؛ برای این کار ابتدا باید ۴ بایت مربوط به داده Float را به وسیله تابع ReadPAC.vi به برنامه LabView منتقل کنیم.

در این صورت داده‌های منتقل شده به برنامه LabView، به صورت یک آرایه با ۴ المان بایتی می‌باشد. برای تبدیل آرایه مذکور به یک مقدار Float برای استفاده در نرم‌افزار LabView، می‌توانیم از تابع StrToFloat.vi استفاده کنیم.

برای استفاده از این تابع کافی است آرایه ۴ بایتی انتقال داده شده از دستگاه را به ورودی آن متصل کنیم، با اجرای این تابع، مقدار معادل Float آن در خروجی تابع قرار می‌گیرد.



مثال:



تابع FloatToStr:

این تابع زمانی استفاده می‌شود که بخواهیم با استفاده از نرم‌افزار LabView، یک داده از جنس Float را در حافظه یک دستگاه PAC بنویسیم. نحوه نوشتن این داده Float در دستگاه PAC به شرح زیر است.

استفاده به عنوان SubVI:

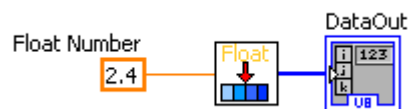
می‌دانیم که داده Float در حافظه دستگاه PAC از ۴ بایت تشکیل شده است. برای انتقال مقدار یک داده Float از نرم‌افزار LabView به یک دستگاه PAC نمی‌توانیم این ۴ بایت را مستقیماً به صورت Float در آن بنویسیم؛ برای این کار باید ۴ بایت معادل داده Float را به وسیله تابع WritePAC.vi به حافظه دستگاه منتقل کنیم.

برای به دست آوردن ۴ بایت معادل یک داده Float می‌توانیم از تابع FloatToStr استفاده کنیم. در صورتی که مقدار داده Float مورد نظر را در ورودی این تابع قرار دهیم، خروجی تابع آرایه ۴ بایتی مورد نظر را به دست می‌دهد.

با انتقال این ۴ بایت به دستگاه PAC و قرار دادن آنها در ۴ بایت معادل یک داده Float، مقدار Float مورد نظر در حافظه دستگاه PAC قرار می‌گیرد.

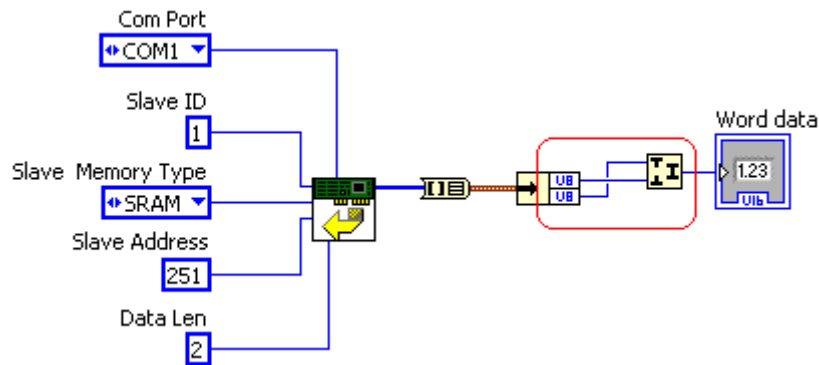


مثال:



خواندن داده Word از حافظه دستگاه PAC:

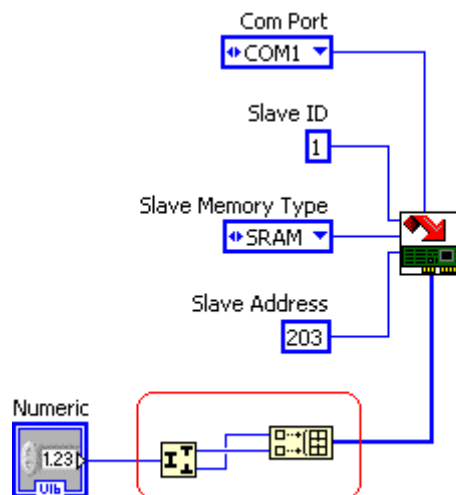
از آنجا که خواندن داده از حافظه دستگاه PAC با استفاده از تابع ReadPAC به صورت بایتی انجام می‌گیرد، برای انتقال داده از جنس Word، باید ابتدا بایت‌های آن را با استفاده از تابع ReadPAC بخوانیم و سپس مقدار معادل آن را محاسبه کنیم. برای محاسبه مقدار یک داده Word از روی بایت‌های منتقل شده به نرم‌افزار LabView به صورت زیر عمل می‌کنیم.



با توجه به اینکه در حافظه دستگاه PAC، بایت کم ارزش یک داده Word قبل از بایت پرارزش آن قرار دارد، باید جای بایت‌های خوانده شده از حافظه دستگاه را باهم عوض کنیم تا همان مقدار صحیح در خروجی قرار گیرد.

نوشتن داده Word در حافظه دستگاه PAC:

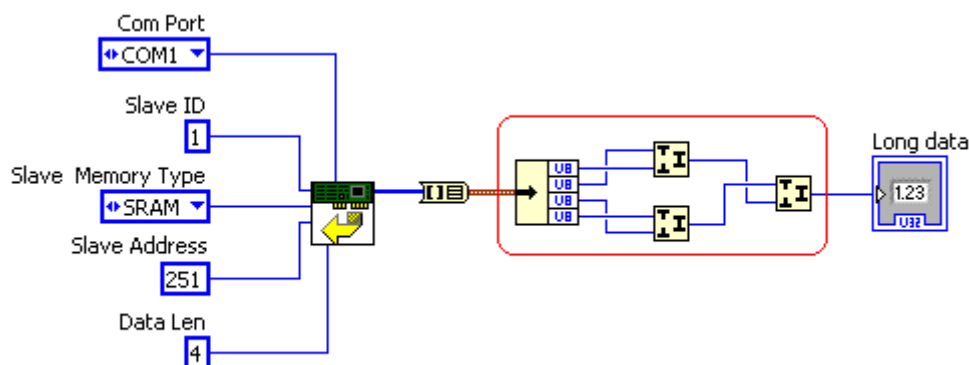
از آنجا که نوشتن داده در حافظه دستگاه PAC با استفاده از تابع WritePAC.vi به صورت بایتی انجام می‌گیرد، برای انتقال داده از جنس Word، باید ابتدا آن را به صورت یک آرایه ۲ بایتی تبدیل کنیم و سپس بایت‌های آن را با استفاده از تابع WritePAC.vi به حافظه دستگاه PAC منتقل کنیم. برای محاسبه آرایه ۲ بایتی معادل یک داده Word در نرم‌افزار LabView به صورت زیر عمل می‌کنیم.



با توجه به اینکه در حافظه دستگاه PAC، بایت کم ارزش یک داده Word قبل از بایت پرارزش آن قرار دارد، باید جای بایت‌های داده مورد نظر را باهم عوض کنیم تا همان مقدار صحیح در حافظه Word دستگاه قرار گیرد.

خواندن داده Long از حافظه دستگاه PAC:

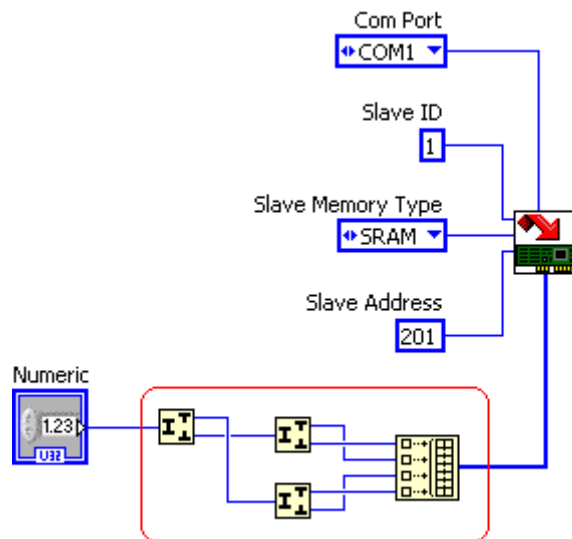
از آنجا که خواندن داده از حافظه دستگاه PAC با استفاده از تابع ReadPAC به صورت بایتی انجام می‌گیرد، برای انتقال داده از جنس Long، باید ابتدا بایت‌های آن را با استفاده از تابع ReadPAC بخوانیم و سپس مقدار معادل آن را محاسبه کنیم. برای محاسبه مقدار یک داده Long از روی بایت‌های منتقل شده به نرم‌افزار LabView به صورت زیر عمل می‌کنیم.



با توجه به اینکه در حافظه دستگاه PAC، بایت‌های کم ارزش یک داده Long قبل از بایت‌های پرارزش آن قرار دارند، باید جای بایت‌های خوانده شده از حافظه دستگاه را عوض کنیم (ترتیب آنها را برعکس کنیم) تا همان مقدار صحیح در خروجی قرار گیرد.

نوشتن داده Long در حافظه دستگاه PAC:

از آنجا که نوشتن داده در حافظه دستگاه PAC با استفاده از تابع WritePAC.vi به صورت بایتی انجام می‌گیرد، برای انتقال داده از جنس Long، باید ابتدا آن را به صورت یک آرایه ۴ بایتی تبدیل کنیم و سپس بایت‌های آن را با استفاده از تابع WritePAC.vi به حافظه دستگاه PAC منتقل کنیم. برای محاسبه آرایه ۴ بایتی معادل یک داده Long در نرم‌افزار LabView به صورت زیر عمل می‌کنیم.



با توجه به اینکه در حافظه دستگاه PAC، بایت‌های کم ارزش یک داده Long قبل از بایت‌های پر ارزش آن قرار دارد، باید جای بایت‌های داده مورد نظر را باهم عوض کنیم (ترتیب آنها را برعکس کنیم) تا همان مقدار صحیح در حافظه Long دستگاه قرار گیرد.

تابع Slave:

با استفاده از این تابع می‌توان نرم‌افزار LabView را به عنوان یک دستگاه Slave در یک شبکه RS-485 قرار داد. به عبارت دیگر، در صورتی که این تابع در یک برنامه LabView به صورت مداوم اجرا شود، کامپیوتر می‌تواند به عنوان یکی از دستگاه‌های متصل به شبکه RS-485 با سایر دستگاه‌های شبکه ارتباط برقرار کند. باید در نظر داشت که کامپیوتر در این حالت تنها می‌تواند یک Slave در شبکه مذکور باشد.

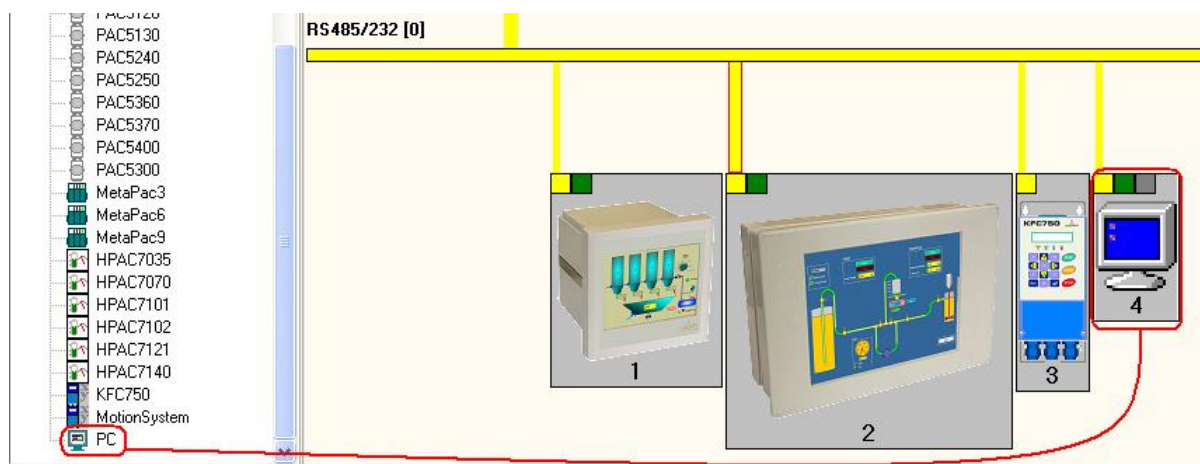
برای قرار دادن کامپیوتر به عنوان یک Slave در شبکه به صورت زیر عمل می‌کنیم.

ابتدا پروژه مورد نظر خود را در نرم‌افزار FBD Editor ایجاد می‌کنیم.

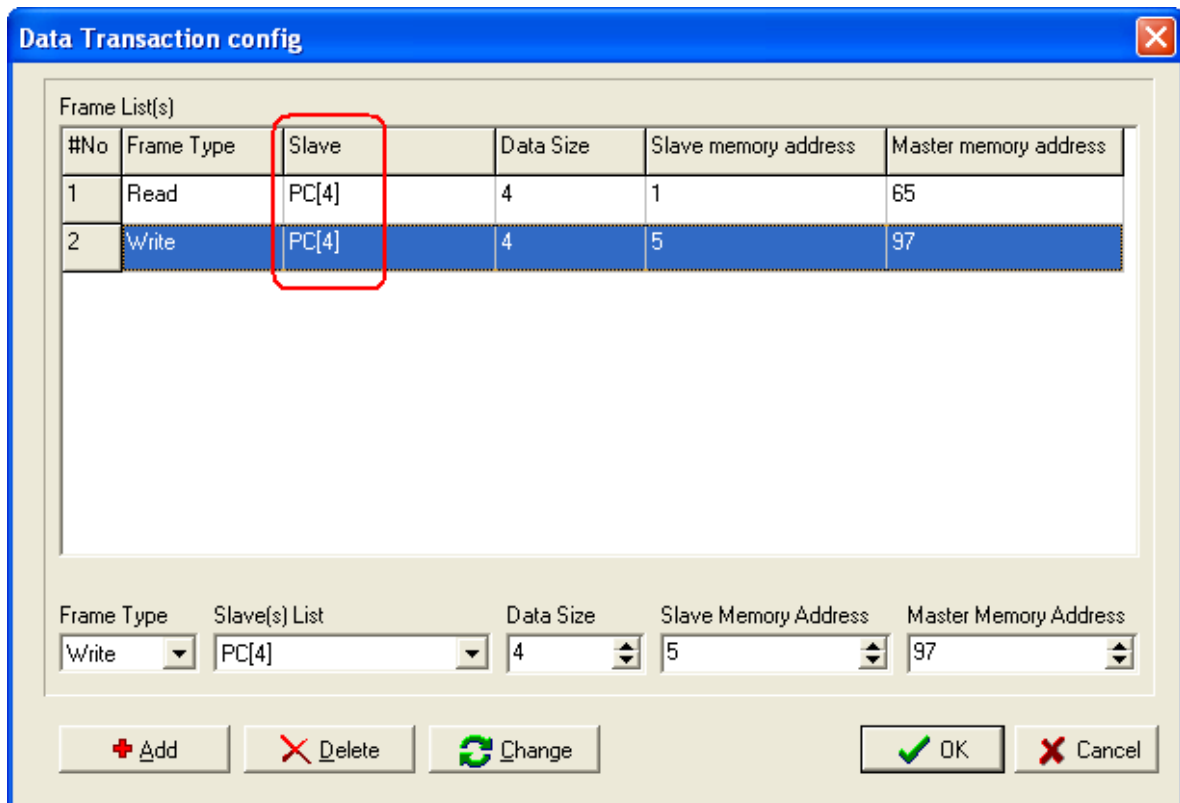
در بخش Hardware Config، دستگاه‌های مورد نیاز را به پروژه اضافه نموده و آنها را با هم شبکه می‌کنیم.

از مجموعه Devices در بخش Object Explorer، المان «PC» را انتخاب کرده و آن را در کنار سایر دستگاه‌ها در بخش Hardware Config قرار می‌دهیم.

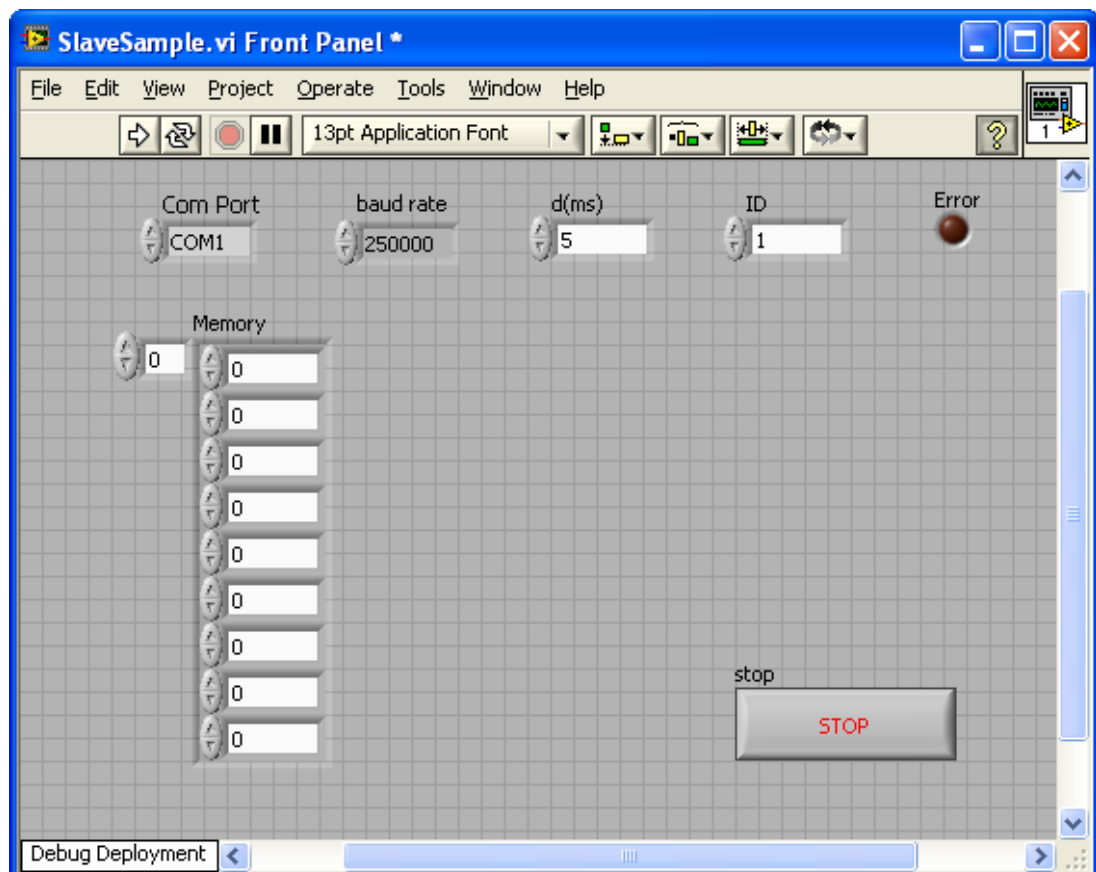
پورت «PC» را همانند سایر دستگاه‌ها به باس شبکه متصل می‌کنیم.

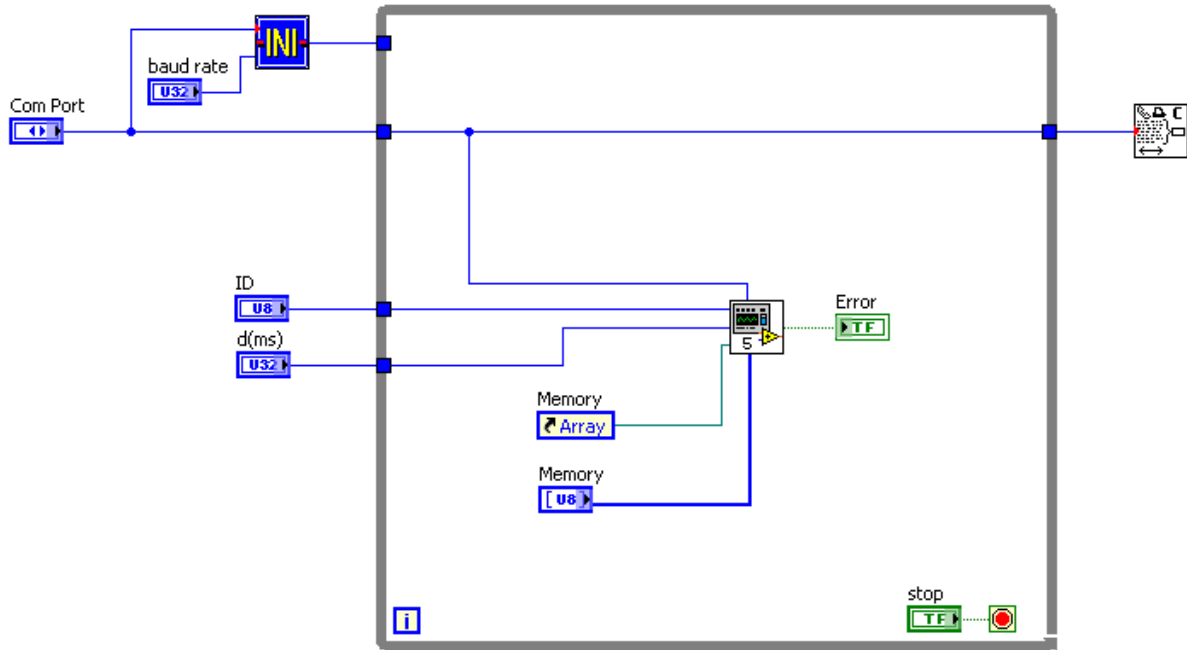


پس از اینکه المان «PC» در شبکه قرار گرفت می‌توان در بخش Data Transaction شبکه، فریم انتقال اطلاعات را بین Master در شبکه و «PC» ایجاد نمود.



تابع Slave.vi را در یک برنامه LabView به صورت زیر استفاده می‌کنیم:





- در بخش Com Port، شماره پورتهی را که کامپیوتر از طریق آن با شبکه RS-485 ارتباط برقرار می کند، تعیین می کنیم.
 - در بخش baud rate، سرعت ارتباط کامپیوتر با شبکه RS-485 را تعیین می کنیم. باید توجه داشت که baud rate انتخاب شده در این بخش باید همان baud rate باشد که در نرم افزار FBD Editor برای شبکه تعیین شده است.
 - در بخش ID، شماره ID المان «PC» در شبکه را تعیین می کنیم. این ID باید همان ID باشد که در نرم افزار FBD Editor برای المان «PC» قرار داده شده است.
 - در بخش d(ms) باید تأخیر (delay) ارسال و دریافت فریم در شبکه را بر حسب میلی ثانیه تعیین کنیم.
 - اگر بین نرم افزار LabView و ارتباط آن با شبکه خطایی رخ دهد، LED مربوط به Error روشن می شود.
- در انتقال اطلاعات در شبکه RS-485، داده ها در حافظه های دستگاه های موجود در شبکه نوشته می شوند و از آنها خوانده می شوند. در تابع Slave.vi، یکی از ورودی های تابع، یک آرایه با عنوان Memory می باشد. این آرایه معادل یک حافظه مجازی برای دستگاه «PC»، در نرم افزار LabView است.
- به عنوان مثال اگر بخواهیم یک داده Long را به نرم افزار LabView روی «PC» انتقال دهیم، فریم زیر را در شبکه ایجاد می کنیم:

Frame List(s)					
#No	Frame Type	Slave	Data Size	Slave memory address	Master memory address
1	Read	PC[4]	4	1	65
2	Write	PC[4]	4	5	97

بدین ترتیب، یک داده Long از آدرس 25Long دستگاه Master، در بایت‌های 5 تا 8 آرایه Memory در نرم‌افزار LabView قرار می‌گیرد.

برای خواندن داده از آرایه مذکور نیز به همین صورت می‌توانیم یک فریم Read در شبکه ایجاد کنیم.

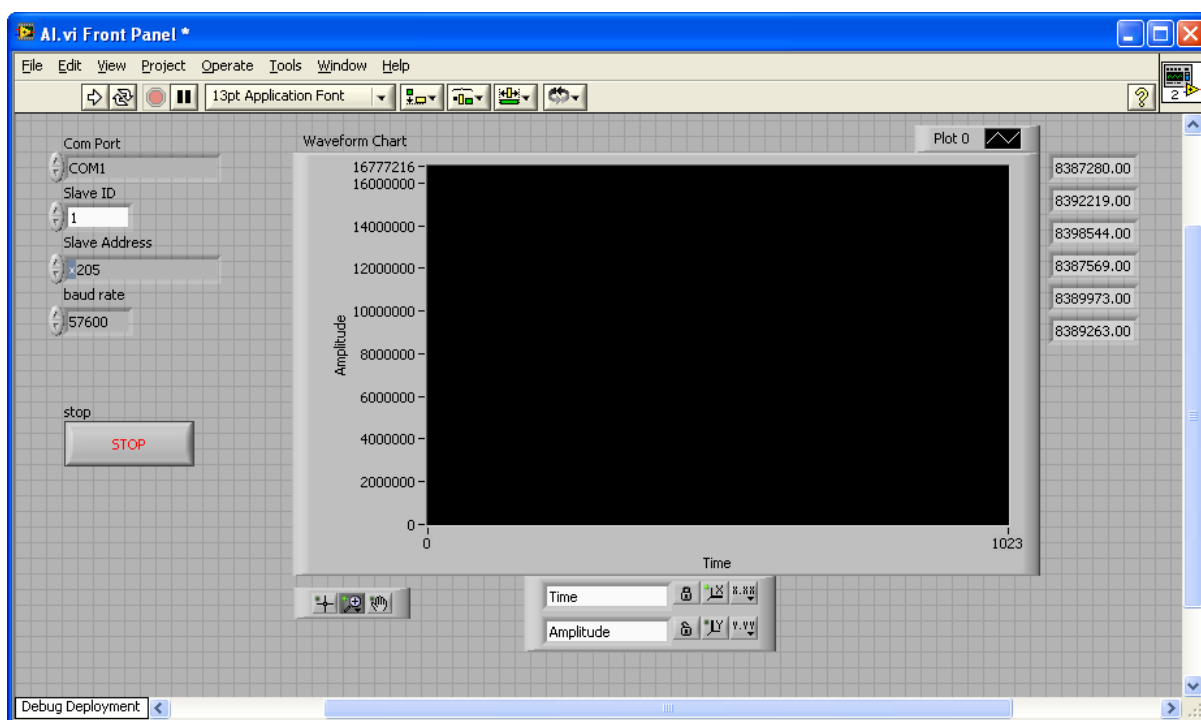
توجه: در صورت انتقال و استفاده از داده‌های از جنس Word و Long و نیز Float در نرم‌افزار LabView، به نکات گفته شده در بخش خواندن و نوشتن داده‌های Word و Long و استفاده از توابع FloatToStr.vi و StrToFloat.vi توجه کنید.

استفاده از توابع کمکی ارتباط با دستگاه‌های سیستم PAC

تابع AI.vi

از این تابع برای خواندن حافظه مربوط به ورودی آنالوگ یکی از دستگاه‌های PAC و رسم نمودار تغییرات آن استفاده می‌شود.

نحوه استفاده از آن به صورت زیر است:

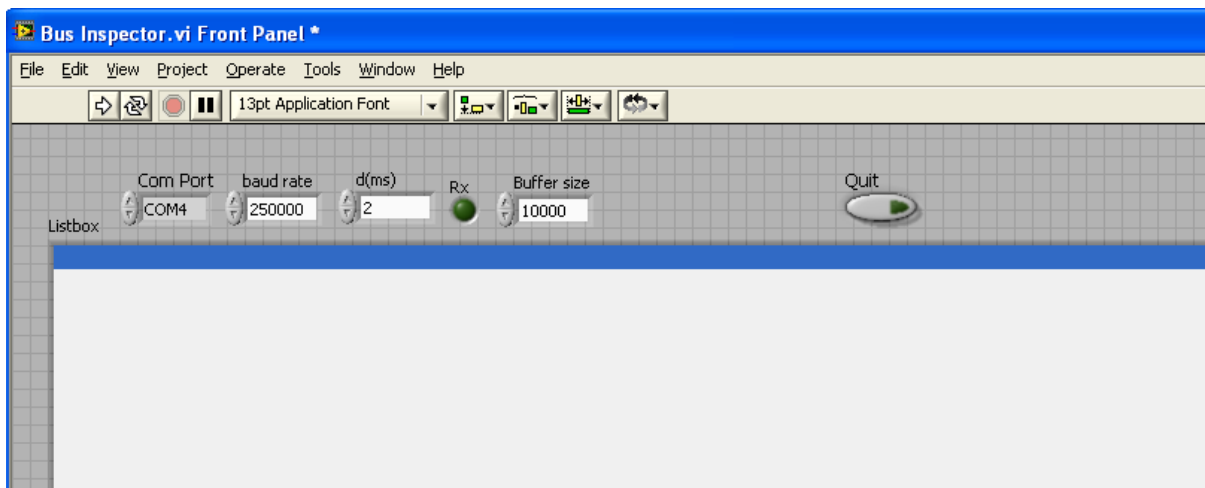


- در بخش Com Port، پورتی از کامپیوتر را که با استفاده از آن به شبکه RS-485 متصل می‌شویم تعیین می‌کنیم.
- در بخش Slave ID، شماره ID دستگاهی که مقدار آنالوگ را می‌خواهیم از آن بخوانیم تعیین می‌کنیم.
- در بخش Slave Address، آدرس بیتی از حافظه دستگاه PAC را که مقدار آنالوگ ورودی در آن بایت و بایت‌های بعدی آن قرار دارد، وارد می‌کنیم.
- در بخش baud rate، سرعت انتقال کامپیوتر با شبکه RS-485 را تعیین می‌کنیم. این baud rate باید با baud rate تعیین شده در دستگاه PAC یکسان باشد.
- با اجرای تابع، مقدار آنالوگ ورودی دستگاه توسط نرم‌افزار LabView خوانده شده و نمودار تغییرات آن رسم می‌گردد.

تابع Bus Inspector

اگر کامپیوتر به وسیله یک مبدل به شبکه RS-485 متصل باشد، به وسیله این تابع می توان فریم های داده ای را که توسط سایر دستگاه های متصل به شبکه، بر روی باس شبکه فرستاده می شود، مشاهده نمود.

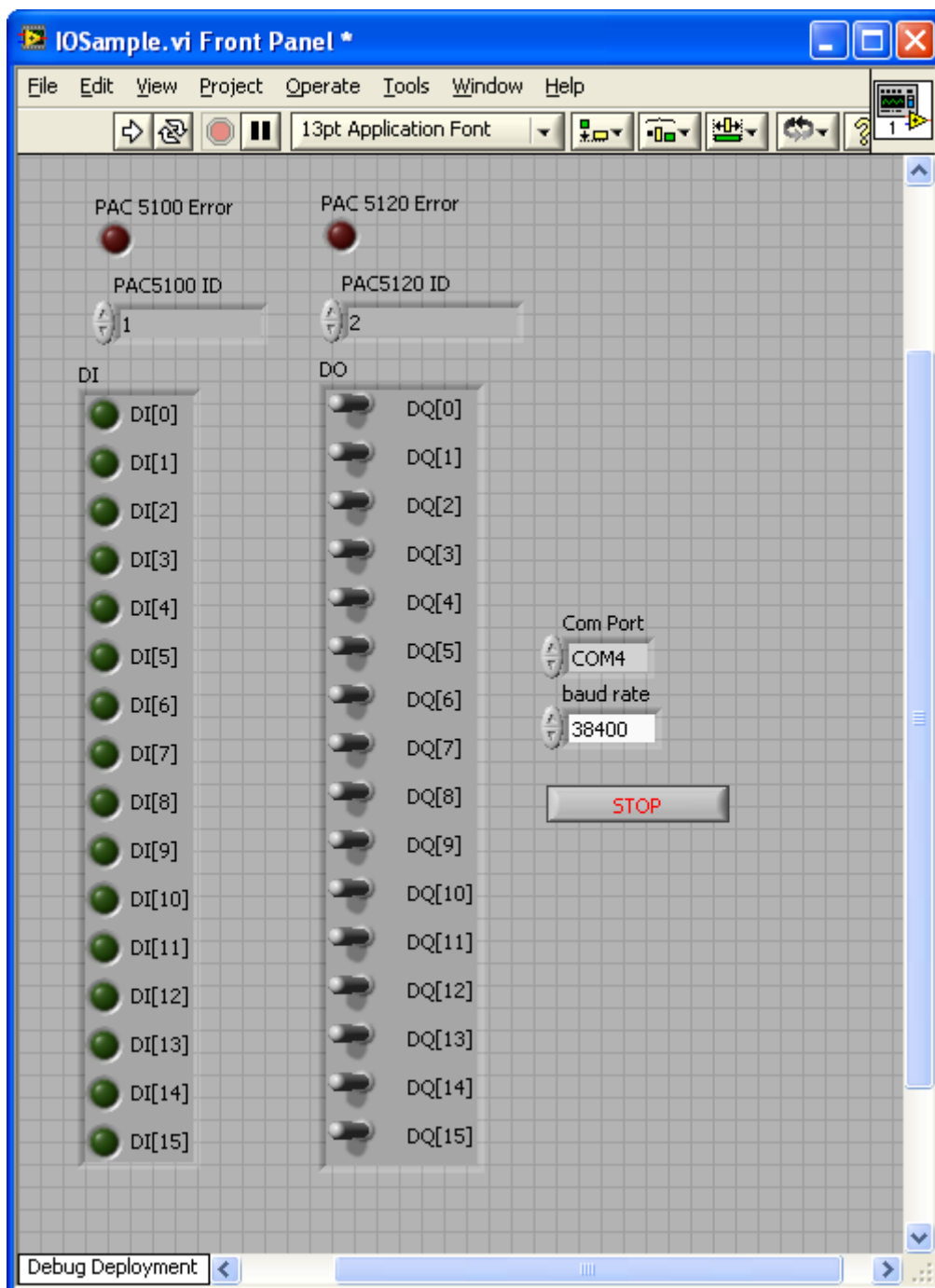
در چنین حالتی کامپیوتر هیچ داده ای را بر روی شبکه نمی فرستد و تنها داده های فرستاده شده بر روی شبکه را دریافت می کند.



- در بخش Com Port، پورتی از کامپیوتر را که با استفاده از آن به شبکه RS-485 متصل می شویم تعیین می کنیم.
- در بخش baud rate، سرعت انتقال کامپیوتر با شبکه RS-485 را تعیین می کنیم. این baud rate باید با baud rate تعیین شده در دستگاه های PAC یکسان باشد.
- در بخش d(ms)، تأخیر دریافت بایت های یک فریم داده را تعیین می کنیم.
- در صورت دریافت هر فریم توسط نرم افزار LabView، LED مربوط به Rx یکبار روشن و خاموش می شود.
- در بخش Buffer size، تعداد فریم هایی را که در فیلد پنجره برنامه LabView نگه داشته می شود، تعیین می کنیم.
- با زدن کلید Quit، عملیات دریافت داده های روی باس شبکه متوقف می شود.
- با اجرای تابع، فریم های ارسال شده بر روی باس شبکه، در فیلد زیرین پنجره این برنامه نشان داده می شوند.

تابع IOsample:

از این تابع برای نمایش وضعیت ورودی‌های دیجیتال دستگاه PAC5100 و فرمان دادن به خروجی‌های دیجیتال دستگاه PAC5120 استفاده می‌شود.



- در بخش Com Port، پورتی از کامپیوتر را که با استفاده از آن به شبکه RS-485 متصل می‌شویم تعیین می‌کنیم.
- در بخش baud rate، سرعت انتقال کامپیوتر با شبکه RS-485 را تعیین می‌کنیم. این baud rate باید با baud rate تعیین شده در دستگاه‌های PAC یکسان باشد.
- در بخش ID، شماره ID مربوط به دستگاه مورد نظر در شبکه را قرار می‌دهیم.
- با اجرای برنامه دستگاه، وضعیت ورودی‌های دیجیتال دستگاه PAC5100 خوانده شده و در صورت 1 بودن هر ورودی دیجیتال، LED مربوط به آن روشن می‌شود.
- همچنین وضعیت اعمال شده به سوئیچ‌های قرار گرفته بر روی برنامه، به خروجی‌های دیجیتال دستگاه PAC5120 اعمال (Force) می‌شود.
- در صورتی که در ارتباط با یکی از دستگاه‌های فوق خطایی پیش آید، LED مربوط به آن روشن می‌شود.