AN-1132
8 Channel Digital Output Data Acquisition Card using LABVIEW

Author: Bilal Ahmed

Date: November 2, 2016

Introduction

Data acquisition (DAQ) hardware acts as the interface between a computer and external signals. It transfers data (Digital and Analog) to and from a computer to devices installed in the field. DAQ cards are of fundamental importance for industrial, research and development, and many other applications.

This application note shows how Digital data is transferred from a computer (using LabVIEW) to a GreenPAK4 SLG46620V, which then drives relays with 220VAC / 10A contact rating. Featuring 8 Digital Output channels, this DAQ using GreenPAK4 SLG46620V can be completed for less than $10, which is efficient and low cost.

Also shown is an interface established between LabVIEW and GreenPAK4 SLG46620V by using a USB to TTL module (FT232RL). 8 relays are controlled via the LabVIEW GUI. The small size of GreenPAK4 SLG46620V provides a portable solution for DAQ cards.

Brief Introduction of LabVIEW and NI-VISA

LabVIEW

Labview software is widely known and used for a wide variety of industrial applications. LabVIEW is a highly productive development environment for creating custom applications that interact with real-world data or signals in different fields.

The benefits of using LabVIEW are shorter programming and debug time, graphical programming, wide contractor network, and high quality results.

Across different industries, the tools and components required for different projects vary widely, and it can be a challenging task to find and use all these disparate items together. LabVIEW is unique because it makes this wide variety of tools available in a single environment, ensuring that compatibility is as simple as drawing wires between functions.

LabVIEW is different from most other general-purpose programming languages in two major ways.

1. Programming is performed by wiring together graphical icons on a diagram, which is then compiled directly to machine code so the computer processors can execute it. While represented graphically instead of with text, it contains the same programming concepts found in most traditional languages.

2. The second main differentiator is that LabVIEW code executes according to the rules of data flow instead of sequential commands.

This distinction renders the data paths between different parts of the program to be the developer's main focus. Different functions in a LabVIEW program have inputs, process data, and produce outputs. Once all inputs contain valid data, the function executes its logic, produces output data, and passes that data to the next function in the dataflow path. A function that receives data from another function can execute only after the other function completes execution.

NI-VISA

The Virtual Instrument Software Architecture (VISA) is a standard for configuring, programming, and troubleshooting instrumentation systems comprising GPIB, VXI, PXI, Serial, Ethernet and USB interfaces. VISA provides the programming interface between the hardware and LabVIEW. NI-VISA is the National Instruments implementation of the VISA I/O standard. NI-VISA includes software libraries, interactive utilities and the VISA Interactive Control, and configuration programs through Measurement & Automation Explorer for all developmental needs.

DAQ Hardware design

The complete 8-Ch DO DAQ Hardware schematic is shown in Fig. 1.

Configuring the COM port

When the module is detected as COM port in the device Manager, it shall be configured through NI MAX.

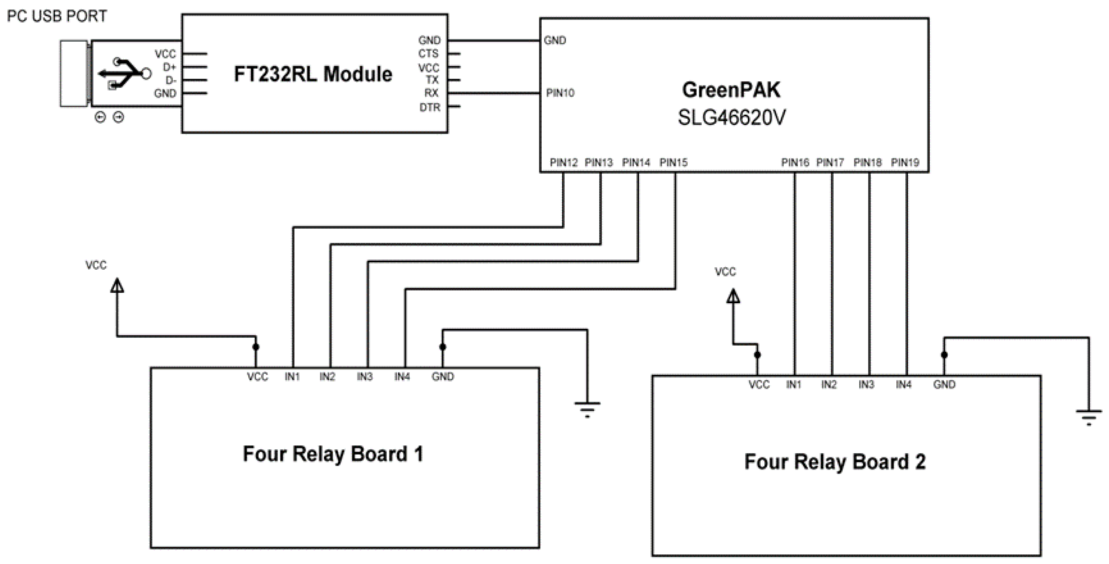


Figure 1. 8-Ch DO DAQ Schematic

FT232RL USB to TTL Module

FT232RL USB to Serial Breakout for the FT232RL is a small board with a built in USB to serial UART interface. This little breakout is built around the FT232RL IC from FTDI, with an internal oscillator.

The board has a jumper for setting 3.3V or 5V voltage level. We are using 5V, and connecting the Tx pin of this board to PIN 10 of the GreenPak Development kit. PIN 10 will be programmed as Rx pin in GreenPAK code to receive the serial data from LabVIEW.

When the module is inserted in the USB port of the PC, it gets detected as a COM port in the Device Manager.

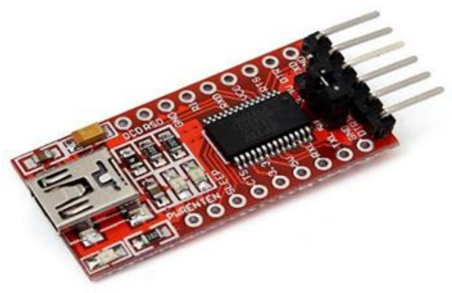


Figure 2. FT232RL USB to TTL Module

The configuration includes setting the baudrates, parity, number of data bits and Flow control. We are using the default parameters of the COM port. These parameters can also be changed later in the LabVIEW Programming.

Four Channel 5V Relay Module Board

Each module board has four Songle relays (SRD-05VDC-SL-C) and each relay is driven by an 817C optocoupler. The advantage is that ground isolation is available. An external 5V power supply is used for VCC and ground pins. SLG46620V digital pins interface at the 'IN' pins on the board.

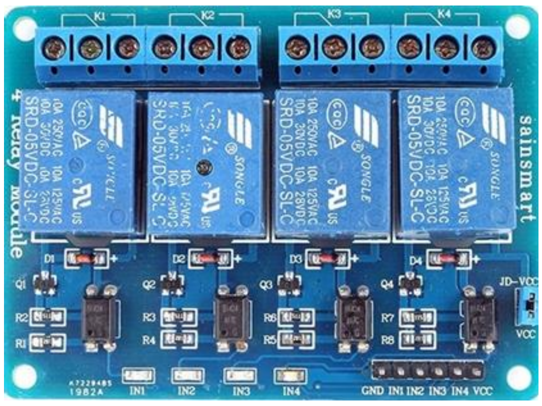


Figure 3. Four Channel 5V Relay Module Board

GreenPAK Design code

The serial data written on the COM port is received by PIN 10 of the GreenPAK4 SLG46620V, the data being sent to the SPI block MOSI input.

In GreenPAK designer, the SPI block Mode property is set to S2P, CPHA and CPOL property are both set to '0'. Byte selection property is [7:0] and serial data is set to be from PIN 10.

Now the serial data (10 bits total, one start bit, 8 data bits and one stop bit) which reaches the SPI block needs to be converted into parallel bits. For this purpose we need to set the SCLK of the SPI block so that it matches the baudrates of 9600 bps (which was set in the LabVIEW). SCLK needs to be started when the serial data is transmitted by LabVIEW.

Also, start and stop bits need to be separated from data bits. The Start bit is first in the packet, and Stop bit is last received by SPI. Moreover, when no data is received there is a continuous high signal at the SPI.

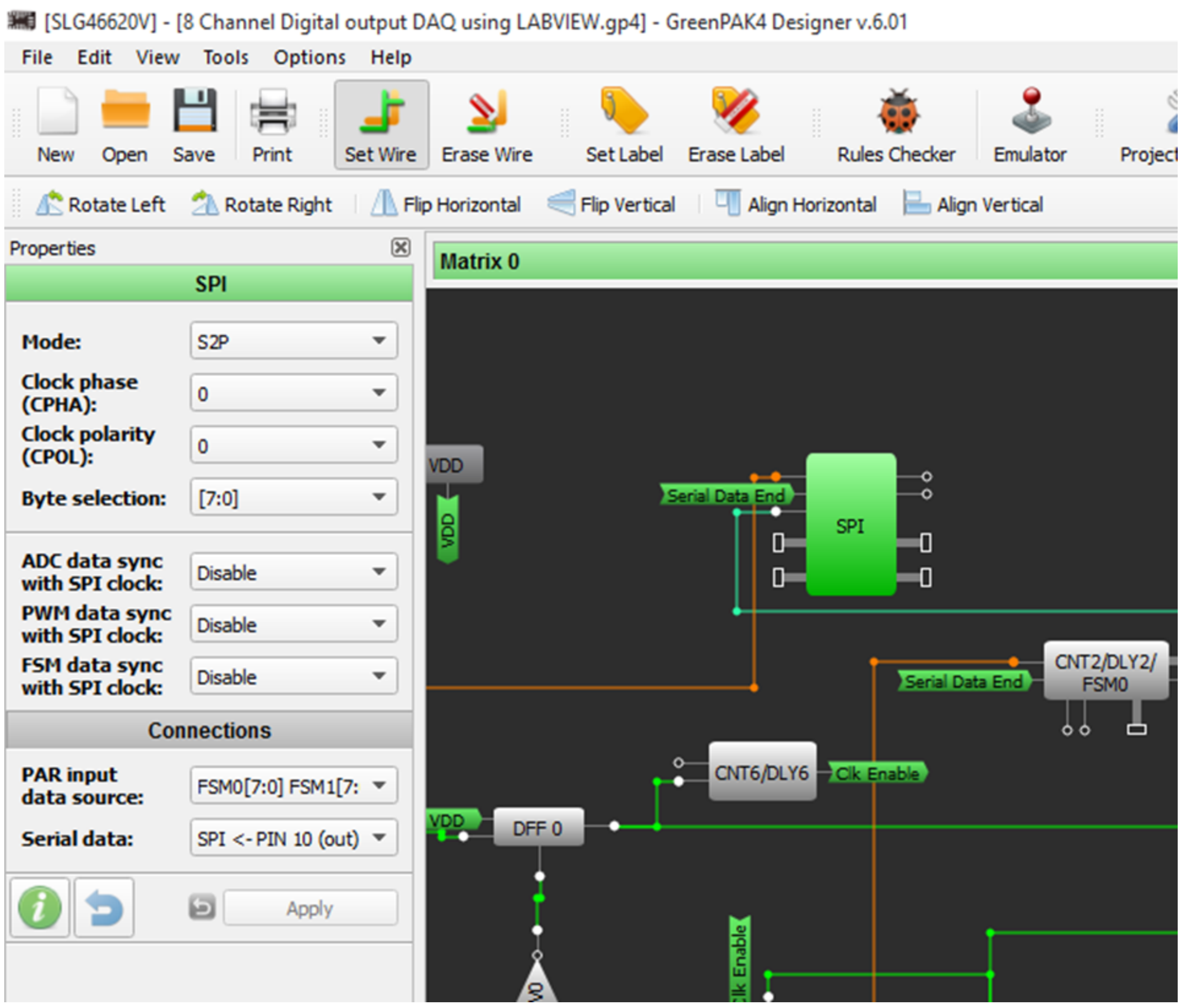
Catching the start bit of serial data

Whenever the data is sent to the serial port from LabVIEW, the start bit which is at the start of the packet is LOW, arriving first. So we can detect a new packet arriving by detecting the falling edge of the incoming data at PIN 10. This is done by inputting PIN 10 to a P DLY0 block, which is configured to detect Falling edge of the input. So whenever the data packet arrives, the falling edge (of start bit) is detected by the P DLY0 block and it sets the output HIGH for one cycle.

The next step is to latch the pulse at P DLY0 output. The signal is sent to CLK input of DFF 0. The D input is set high by VDD signal, so when the positive edge of clock (from P DLY0 output) arrives it stores the HIGH at D input to the output of DFF 0.

Resetting the Latch for storing Start bit of next cycle

For resetting the DFF 0 before the second cycle of the incoming data, the DFF 0 latch output is sent to the input of time delay CNT5/DLY5. The block is configured as a delay for a time of 924 microseconds. Each cycle of 9600 bps corresponds to 104 microseconds, so after almost 9 cycles of the incoming data, the delay block resets the DFF 0.


[SLG46620V] - [8 Channel Digital output DAQ using LABVIEW.gp4] - GreenPAK4 Designer v.6.01
File Edit View Tools Options Help
New Open Save Print Set Wire Erase Wire Set Label Erase Label Rules Checker Emulator Project
Rotate Left Rotate Right Flip Horizontal Flip Vertical Align Horizontal Align Vertical
Properties
Matrix 0
VDD
Serial Data End
SPI
CNT2/DLY2/ FSM0
CNT6/DLY6
Clk Enable
DFF 0
Apply

SPI

Mode: S2P

Clock phase (CPHA): 0

Clock polarity (CPOL): 0

Byte selection: [7:0]

ADC data sync with SPI clock: Disable

PWM data sync with SPI clock: Disable

FSM data sync with SPI clock: Disable

Connections

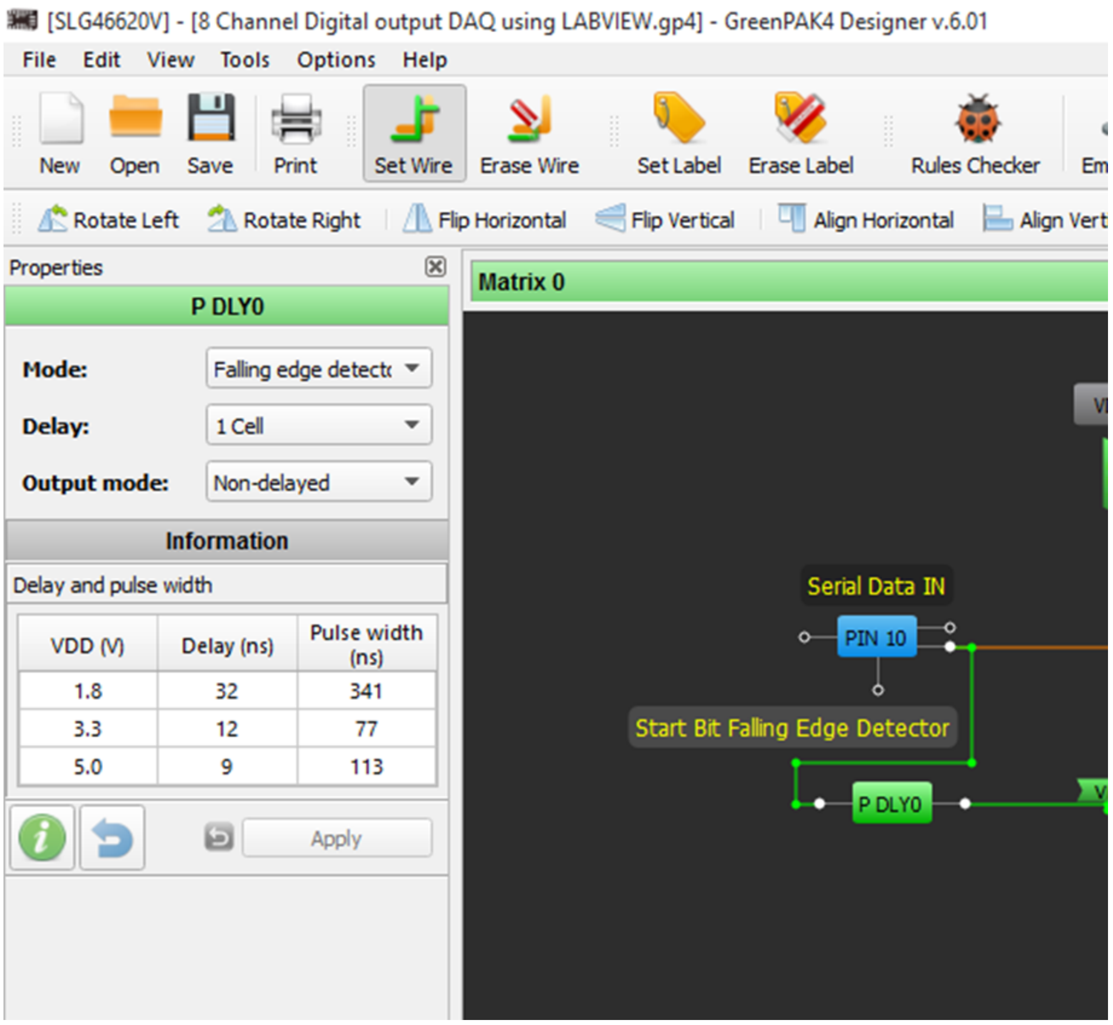
PAR input data source: FSM0[7:0] FSM1[7:

Serial data: SPI <- PIN 10 (out)

Figure 4. SPI block Properties

The CNT2/DLY2 and the DFF 0 latch both are reset by same signal (Serial Data End). The Pipe Delay 0 module gives an additional delay of 3 clock cycles of 2MHz clock before resetting of DFF 0, this is just to make sure that the 9th data bit is loaded in the SPI block before resetting CNT2/DLY2 counter.

The Serial Data End signal LOW shall enable the SPI block nCSB input, then after 9th cycle of incoming data, the signal goes high and SPI block is disabled.



P DLY0

Mode: Falling edge detect

Delay: 1 Cell

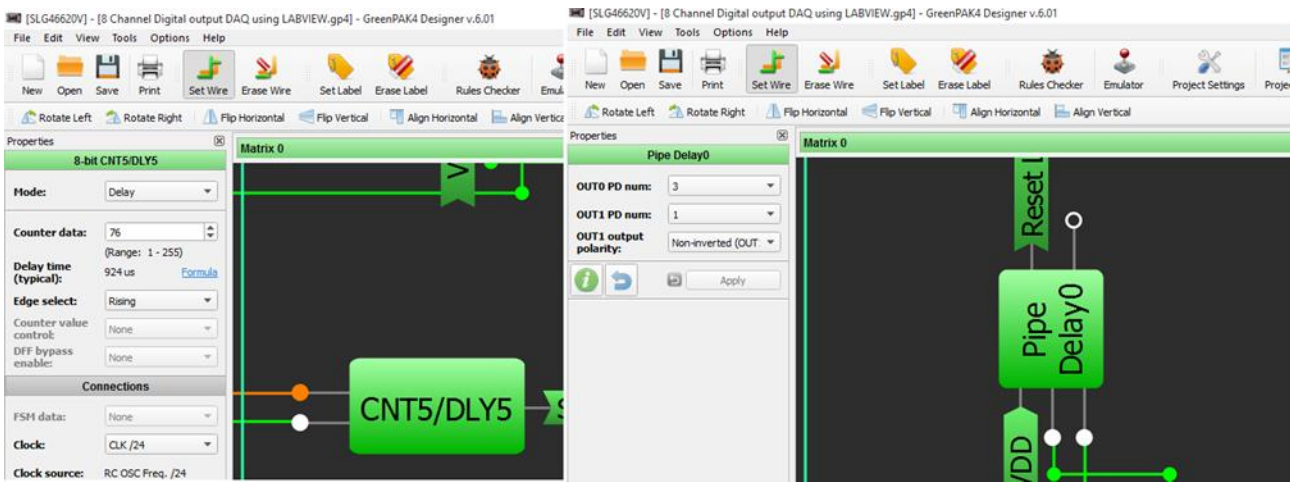
Output mode: Non-delayed

Information

Delay and pulse width

VDD (V)	Delay (ns)	Pulse width (ns)
1.8	32	341
3.3	12	77
5.0	9	113

Figure 5. P DLY0 block Properties



8-bit CNT5/DLY5

Mode: Delay

Counter data: 76

(Range: 1 - 255)

Delay time (typical): 924 us Formula

Edge select: Rising

Counter value control: None

DFF bypass enable: None

Connections

FSM data: None

Clock: CLK /24

Clock source: RC OSC Freq. /24

Pipe Delay0

OUT0 PD num: 3

OUT1 PD num: 1

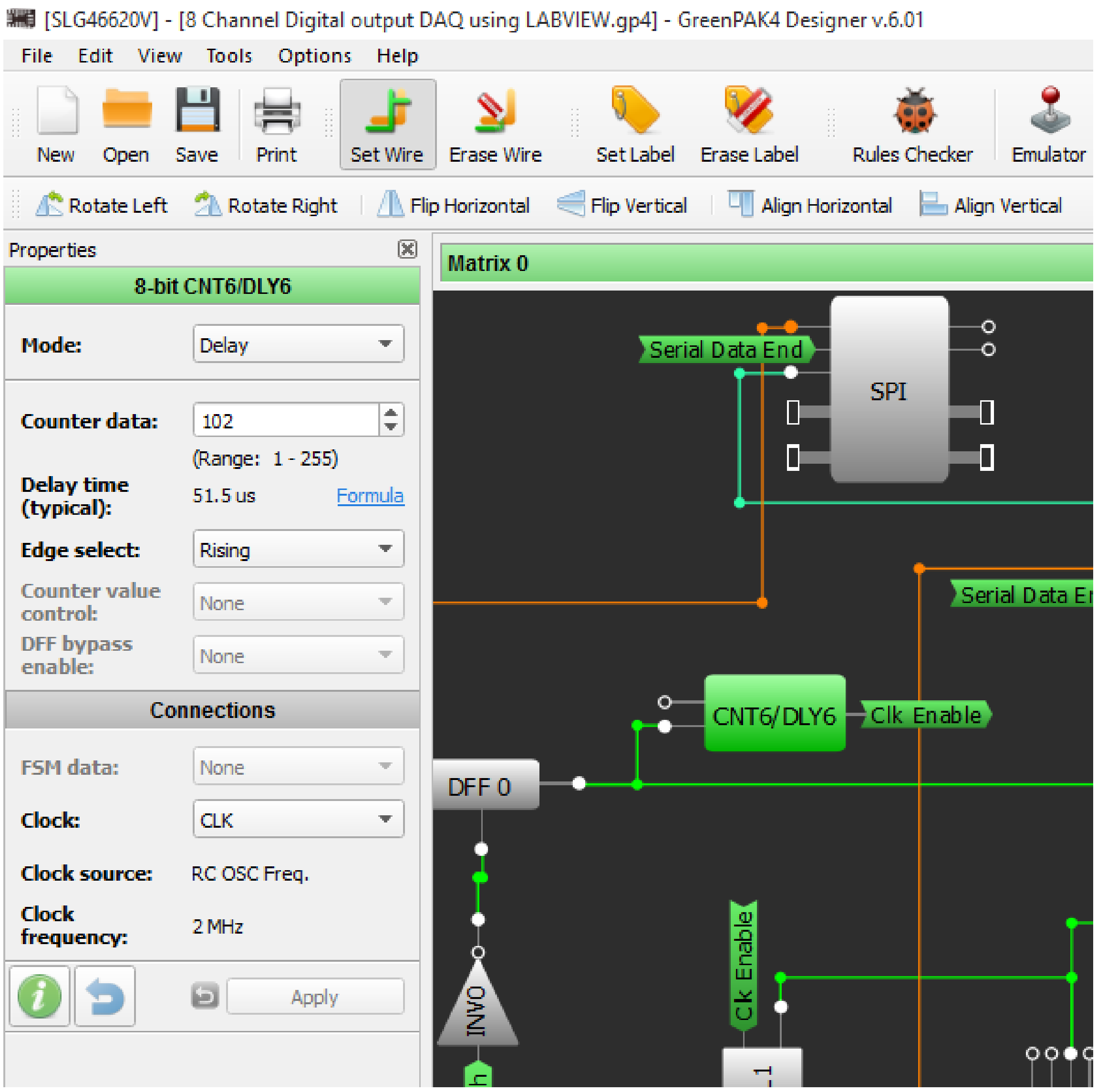
OUT1 output polarity: Non-inverted (OUT

Figure 6. CNT5/DLY5 and Pipe Delay 0 properties

Loading serial data in the SPI block

The latch (DFF 0) high output is also used to load serial data at the MOSI input of the SPI block. For storing serial data at the middle of SCLK, it is necessary to input a delay of half cycle of 9600 bps for SCLK. The time corresponding to half cycle of 9600 bps is approximately 52 microseconds. This time delay is achieved by passing the output of DFF 0 through CNT6/DLY6.

The output of the block is ANDed with the Oscillator clock out1 signal, and sent to the clock signal of CNT2/DLY2 block. The Clock Enable signal makes sure that counter CNT2/DLY2 shall not start counting until the start bit arrives. The CNT2/DLY2 output is then the clock signal for the SPI block (SCLK). It helps load the serial data to SPI block, and once the 10 bit cycle complete, the SPI block converts the serial data to the SPI parallel output block ports.


[SLG46620V] - [8 Channel Digital output DAQ using LABVIEW.gp4] - GreenPAK4 Designer v.6.01

Properties

8-bit CNT6/DLY6

Mode: Delay

Counter data: 102

(Range: 1 - 255)

Delay time (typical): 51.5 us

Edge select: Rising

Counter value control: None

DFF bypass enable: None

Connections

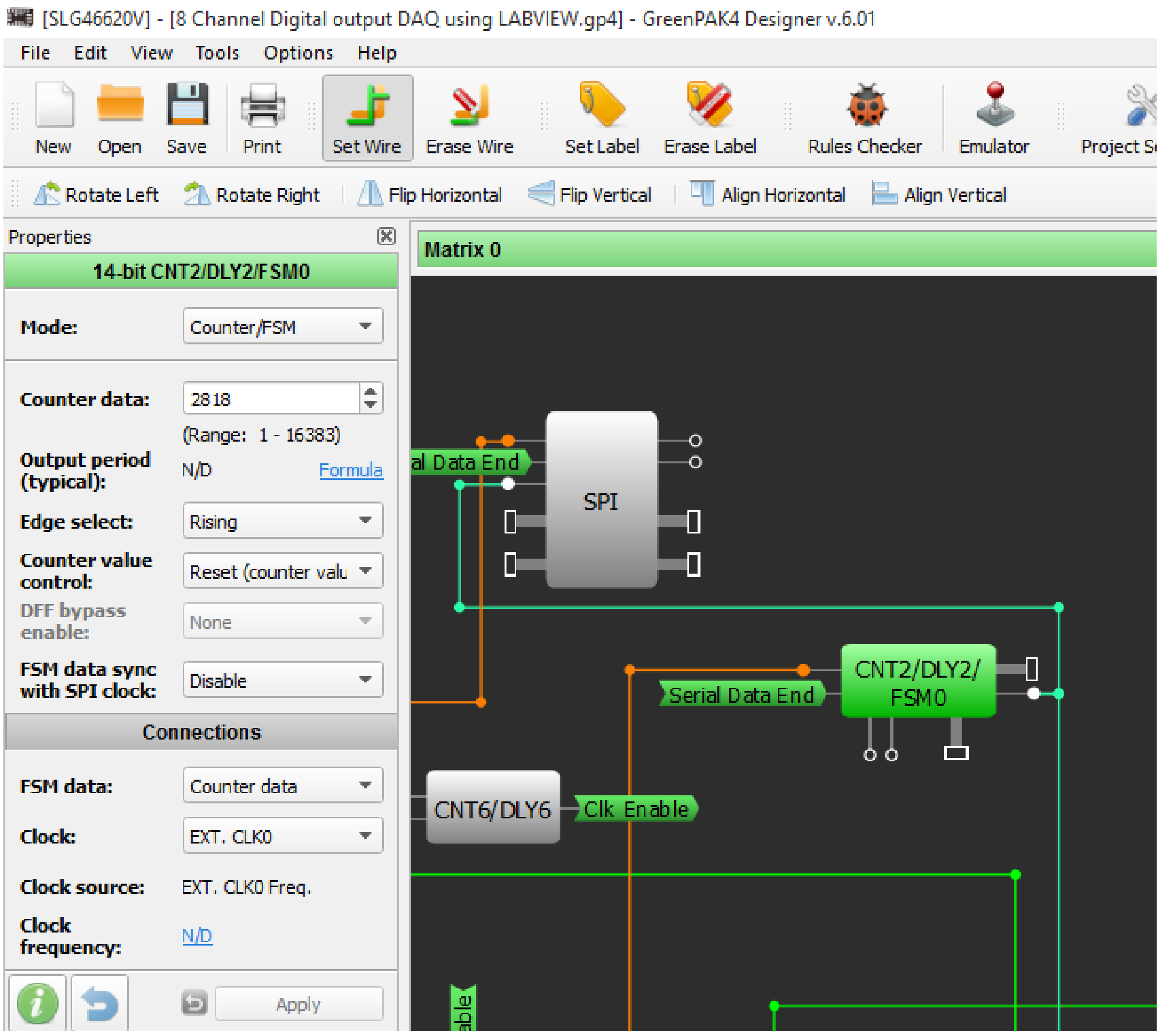
FSM data: None

Clock: CLK

Clock source: RC OSC Freq.

Clock frequency: 2 MHz

Figure 7. CNT6/DLY6 block properties


[SLG46620V] - [8 Channel Digital output DAQ using LABVIEW.gp4] - GreenPAK4 Designer v.6.01

Properties

14-bit CNT2/DLY2/FSM0

Mode: Counter/FSM

Counter data: 2818 (Range: 1 - 16383)

Output period (typical): N/D Formula

Edge select: Rising

Counter value control: Reset (counter valu

DFF bypass enable: None

FSM data sync with SPI clock: Disable

Connections

FSM data: Counter data

Clock: EXT. CLK0

Clock source: EXT. CLK0 Freq.

Clock frequency: N/D

Figure 8. CNT2/DLY2 block properties

Serial Port Data at Digital Output Pins

The serial 8 bit data (D1, D2…D8 buttons of LabVIEW) will appear on the PAR OUT pins of the SPI Parallel Output block according to their weighted order. The four data bits are sent directly to the Digital output pins, the other four are interlocked with the Digital input pins through inverter and 'AND' gate.

So an interlock can be set with these four Digital inputs and the interlock condition will override the corresponding digital output commands from LabVIEW. For example, we start a relay by push button 'DO 1' in LabVIEW, this will make the PIN 12 low. When we stop the relay from the same push button in LabVIEW GUI, PIN 12 goes high. But this is possible only when the Digital Input PIN 2 stays Low. If PIN 12 is high (due to some process condition like high temperature etc.), the DO 1 command from the LabVIEW will not be executed.

Running the LabVIEW GUI

Now insert the FT232RL Module and go to front panel. Click drop down menu of select COM port, press refresh and select the COM port. Its COM6 here (It may be some other number in your case).

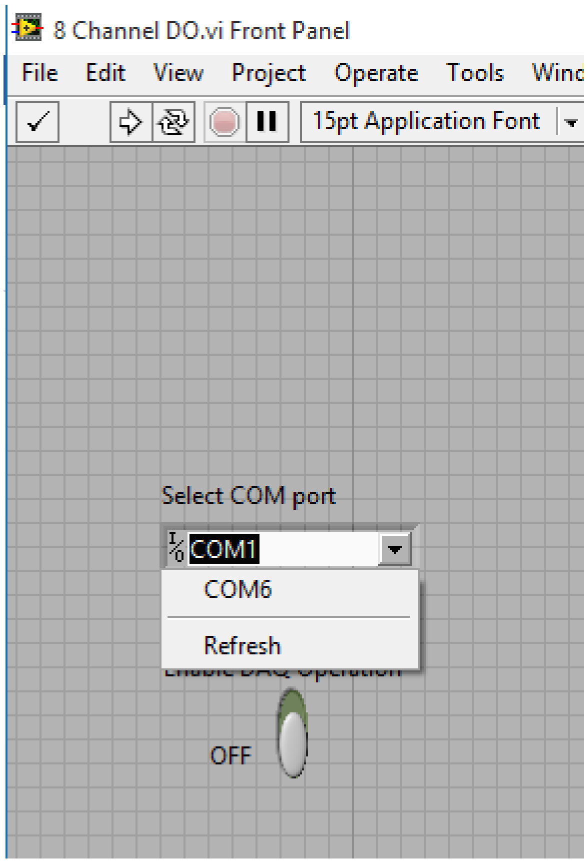


Figure 9. Select COM port in LabVIEW

After selecting the COM port, press Run Continuously. It is in the top right side of the panel.

It must be noted that the COM port must be selected before pressing Run Continuously, otherwise the program will misbehave and then it must be closed from Task Manager.

Also at that point the Tx pin of the module shall be connected to PIN 10 of the GreenPAK development kit. By clicking different push buttons (DO 1, DO 2, …, DO 8) from the Front panel, the corresponding LED's of the GreenPAK development kit will switch ON/OFF.

Example

Heavy AC loads can be controlled by connecting them on the contact side of the 5v relay. The relay contact will behave as a 'Switch' for the load, controlled from LabVIEW GUI.

The digital output of SLG 46620V, corresponding to the DO 'x' push button in the LabVIEW, will pull down and sink the current from the optocoupler on the relay board. Hence no VDD current is drawn from the DO of GreenPAK chip. The optocoupler is switched ON (the Corresponding LED on GreenPAK development board is OFF) and the LED corresponding to optocoupler of the relay board is ON. After the optocoupler is switched ON, the corresponding 5V relay is driven by an external power source. Do not use VDD of the GreenPAK chip to drive the relay directly since maximum current and voltage ratings will likely be exceeded.

The contact side of the relay can be used to drive the AC loads up to 220V and 10A. If even larger loads need to be driven, then connect the contact side to drive the coil of a much larger relay as desired.

A video is available showing how eight 5v relays are switched ON/OFF from the LabVIEW GUI by pressing push buttons D0 1, D0 2 ….DO 8.

Limitations

This design offers a low cost DAQ solution limited to DO channels, and the DO update rate is limited to the COM port baud rate of 9600 bps (approx. 10kHz).

The GreenPAK outputs cannot directly drive these relays due to limited current driving capability of the Digital pins. They require an optocoupler or relay driver, and an external power supply to drive the relay.

Conclusion

This application note demonstrated the usefulness of dedicated DO channel only for industrial controls. While it cannot update faster than 10kHz, Windows applications usually require 200 milliseconds to update anyway. So the COM port baudrate of 9600 bps is comparatively adequate.

The cost advantage of this design makes it an attractive choice for dedicated controls, allowing many more installations. Also, the design using the SLG46620V can be made quite compact, which allows placement where generic DAQ simply would not fit.

References

1. AN-1120 Bluetooth-Controlled Car/Robot (for serial data receive)

AN-1120 Bluetooth-Controlled Car/Robot

2. Advantages of Using LabVIEW

http://www.ni.com/white-paper/8534/en/#toc7

3. National Instruments VISA

https://www.ni.com/visa/

Appendix

LabVIEW Programming Guide (for this application note reference)

After creating a blank VI, go to NI LabVIEW front panel and right click, the Control window opens. Select Modern >> I/O >> VISA Resource.

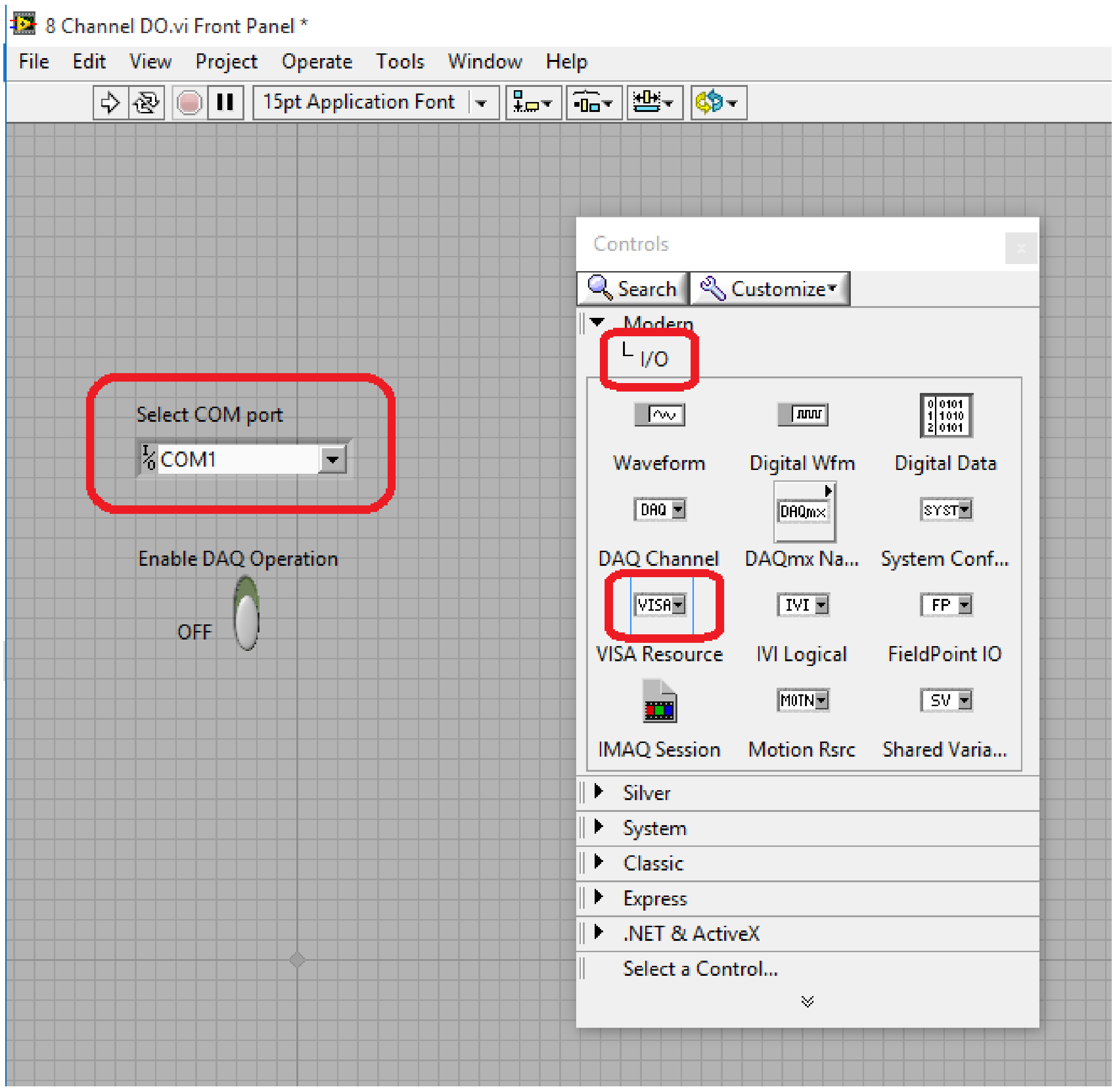


Figure 10. Adding COM port in LabVIEW Front Panel

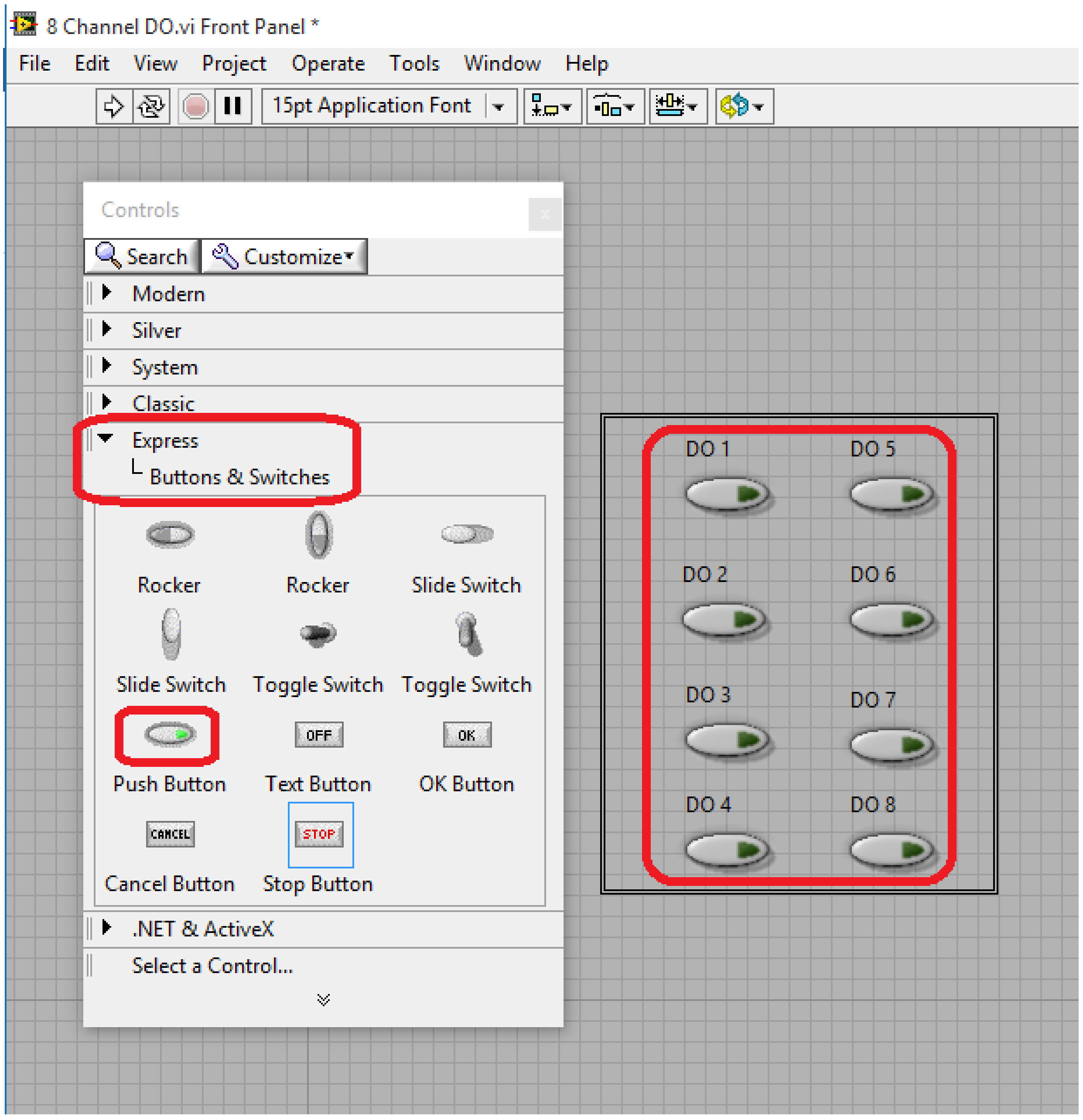


Figure 11. Adding Push Buttons in LabVIEW Front Panel

Now for adding 8 push buttons for 8 Digital outputs, right click again, Select Express >> Button & Switches >> Push Button. Repeat this process 8 times to add 8 buttons.

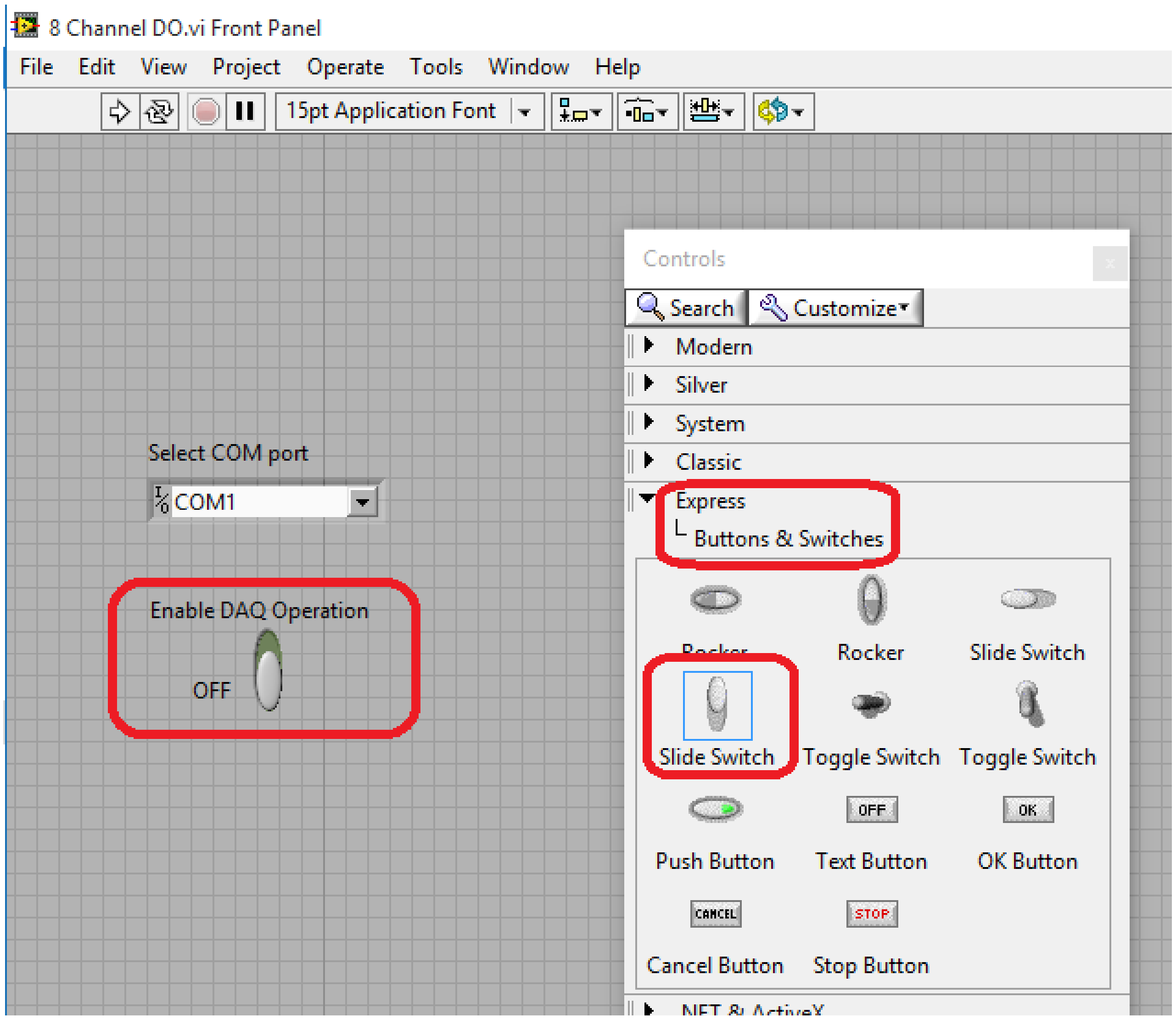


Figure 12. Adding Slide switch in LabVIEW Front Panel

For adding Enable write operation, again right click in LabVIEW Front panel, Select Express >> Button & Switches >> Slide switch.

The text of these buttons, switches and COM port can be edited by double clicking.

Next step in the LabVIEW programming is to join these components to transfer serial data to GreenPAK4 SLG46620V. For this go to Block Diagram, (Press CTRL + E when in Front Panel, CTRL + E toggles the screen between Front Panel and Block Diagram).

It shows 8 buttons, a slide switch, and a select COM Port window. Add numeric constants, multiply, Boolean to integer, case structure, VISA Serial, VISA write, VISA Close, Compound Arithmetic (Addition), Build Array and Byte Array to String in the Block Diagram. These components can be searched in the search box in the right top corner and then added one by one in the Block Diagram.

First of all COM port is configured as shown in the figure below. The wiring can be done by simply hovering mouse over the connection and the left clicking and dragging the mouse towards the target connection.

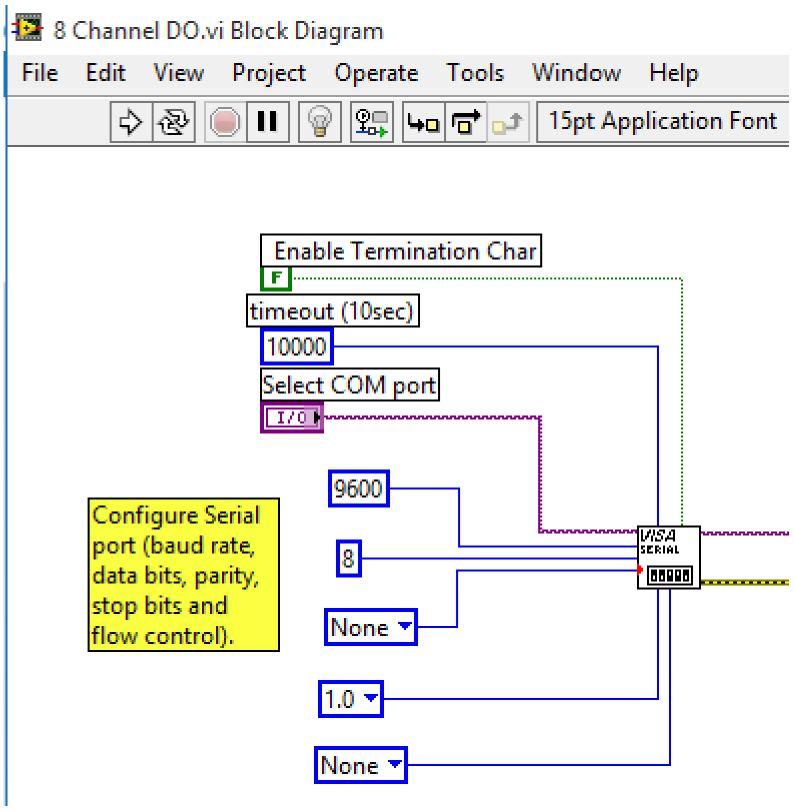

8 Channel DO.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Enable Termination Char
F
timeout (10sec)
10000
Select COM port
I/O
9600
8
None
1.0
None
VISA SERIAL
Configure Serial port (baud rate, data bits, parity, stop bits and flow control).

Figure 13. COM port configuration in LabVIEW

Here we can change the default COM port settings set in the NI MAX.

The Enable DAQ Operation button is used to select the CASE structure. Next step is to configure the TRUE window of CASE structure. When the connection of the COM port is established the control is transferred to this window and the data is written on COM port by clicking the DO buttons.

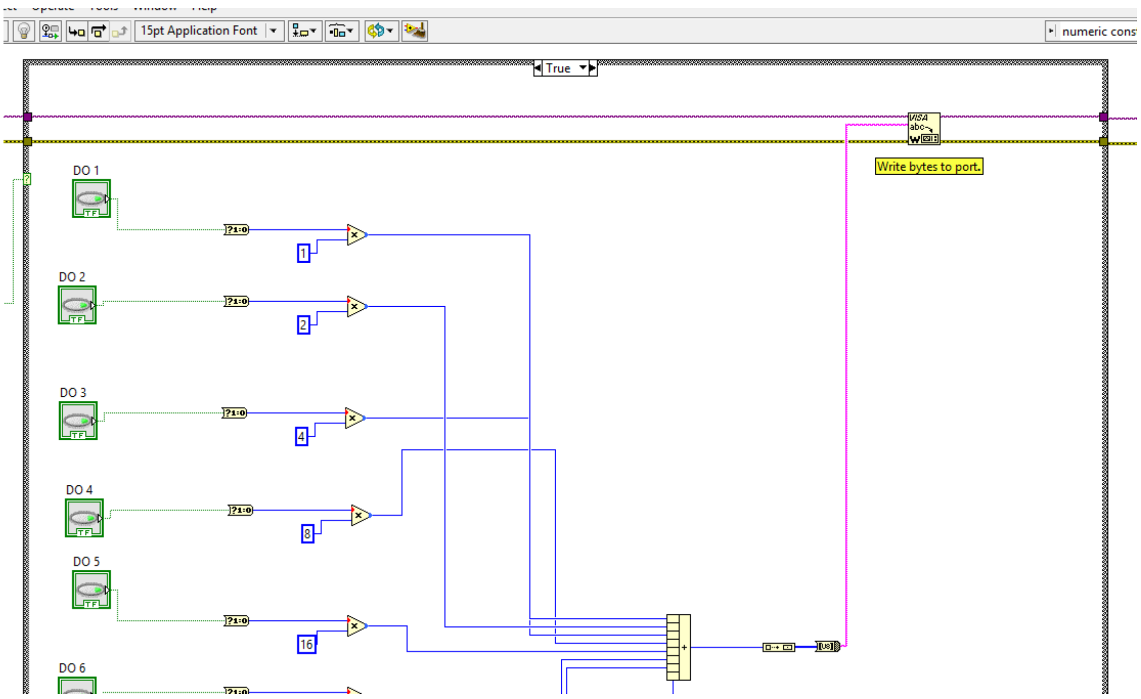


Figure 14. COM port configuration in LabVIEW

Here each button when pressed ON (From LabVIEW GUI during run operation) gives Boolean 1 and when pressed OFF (From LabVIEW GUI during run operation) gives Boolean zero. This 1 or zero is converted to integer and then multiplied by a weighted number (correspond to its location in a byte). Here DO 1 is LSB and DO 8 is MSB.

The product of these eight multiplications is added. This number is then used as input to build an array block and then the result is converted in to string by passing through byte array to string function, which is then input to VISA write block. Reason for converting the integer data to string is that VISA write only accepts string data at its input.

After writing data to COM port, the COM port is closed so that it can be used by other applications without quitting LabVIEW.

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.