FPGA LabVIEW Programming, Monitoring and Remote Control

doi:10.3991/ijoe.v5i2.874

Doru Ursutiu[1], Marius Ghercioiu[2], Cornel Samoila1, Petru Cotfas[1]
[1] University "Transylvania" of Brasov, Romania
[2] National Instruments, USA

***Abstract*—NI ELVIS is an educational design and prototyping platform from National Instruments Corp. based on NI LabVIEW graphical system design software. The platform is used for teaching concepts in areas such as instrumentation, circuits, control, communication, and embedded design in a hands-on, interactive manner. NI ELVIS contains an integrated suite of the 12 most commonly used instruments – DMM, oscilloscope, function generator, MIO, Counter etc – in a compact, rugged, laboratory-friendly form factor.**

A new instrument, the Digital Electronics FPGA Board has been added to the NI-ELVIS platform in order to help educators teach concepts of FPGA programming. This paper presents an example of graphical FPGA programming and monitoring using the Digital Electronics FPGA Board and LabVIEW FPGA graphical programming.

***Index Terms*—FPGA programming, LabVIEW FPGA, Digital Electronics FPGA Board, PDA .**

I. INTRODUCTION

The Digital Electronics Trainer Board is a circuit development platform based on the Xilinx Spartan 3E FPGA. The Digital Electronics Trainer Board Key Components and Features are:

- Xilinx XC3S500E Spartan-3E FPGA
- Up to 232 user I/O
- 320-pin FPGA package
- Over 10,000 logic cells
- 4 Mbit Platform Flash configuration PROM
- 16 Mbits of SPI serial Flash (STMicro) for FPGA configuration storage
- On-board USB-based FPGA/CPLD download/debug interface
- 50 MHz clock oscillator
- 6 Digilent 12-pin expansion connectors (PMOD)
- 4-ch, SPI-based DAC (Digital-to-Analog Converter)
- 2-ch, SPI-based ADC (Analog-to-Digital Converter) with programmable-gain
- pre-amplifier
- 2-digit 7-segment LED display
- Rotary-encoder with push-button shaft
- Eight discrete LEDs
- Eight slide switches
- Four push-buttons
- FPGA Breadboard Area
- NI-Elvis Breadboard Area
- General Purpose Breadboard Area
- NI-ELVIS connector interface
- +15VDC power barrel plug

The Digital Electronics FPGA Board can be used as a plug-in card into the NI-ELVIS teaching platform. The board is powered from the +15V power line generated by NI-ELVIS and it is connected to the Host PC via a USB cable.

Figure 1. NI-ELVIS II platform with the Digital Electronics FPGA Board

The Digital Electronics FPGA Board can also be used as a stand-alone board that sits on the table, it is powered from and external 15V power adapter (min 500mA) and it is connected to the Host PC via a USB cable.

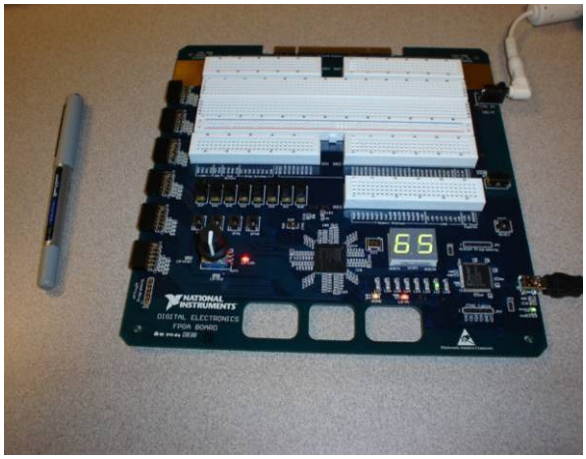


Figure 2. The Digital Electronics FPGA Board used in stand-alone mode

The new NI LabVIEW FPGA Module uses LabVIEW embedded technology to target field-programmable gate arrays (FPGAs). With the LabVIEW FPGA Module, we were able to program the Spartan-3E FPGA on the Digital Electronics FPGA board without low-level hardware description languages or board-level design.

II. GRAPHICAL FPGA PROGRAMMING

The application setup can be:

a) Digital Electronics FPGA Board in stand-alone mode connected to PC via USB cable, and powered from +15VDC power adapter.

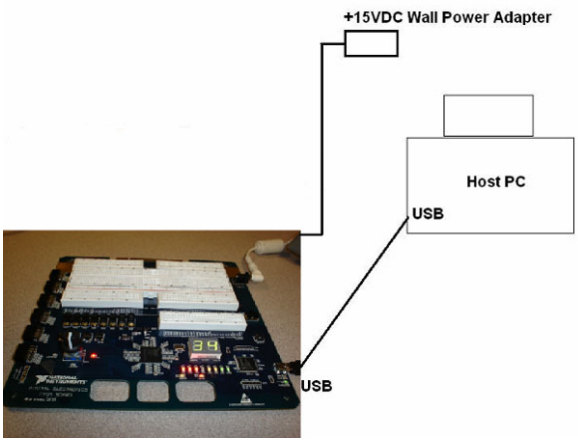


Figure 3. The Digital Electronics FPGA Board used in stand-alone mode

b) Digital Electronics FPGA Board in NI-Elvis mode mounted on the NI-Elvis platform and connected to PC via USB cable.

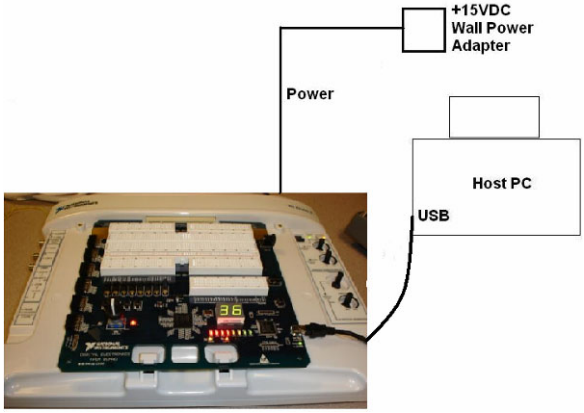


Figure 4. The Digital Electronics FPGA Board used in NI-Elvis mode

Digital Electronics FPGA Board has the following FPGA I/O peripherals:

- 6 Digilent 12-pin expansion connectors (PMOD)
- 4-ch, SPI-based DAC (Digital-to-Analog Converter)
- 2-ch, SPI-based ADC (Analog-to-Digital Converter) with programmable-gain pre-amplifier
- 2-digit 7-segment LED display
- Eight discrete LEDs
- Eight slide switches
- Four push-buttons
- FPGA Breadboard Area
- NI-Elvis Breadboard Area
- General Purpose Breadboard Area

LabVIEW FPGA contains graphical objects named Elemental I/O that represent physical FPGA I/O peripherals of the Digital Electronics FPGA Board in the application diagram (or code). An FPGA I/O object can be activated for a certain application by placing it in the project – right click on FPGA Target device:

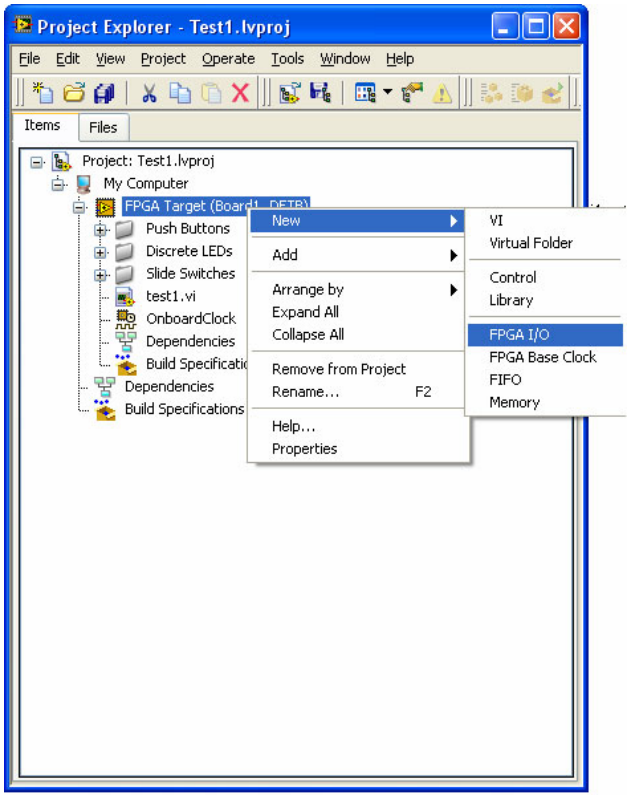


Figure 5. FPGA I/O Selection

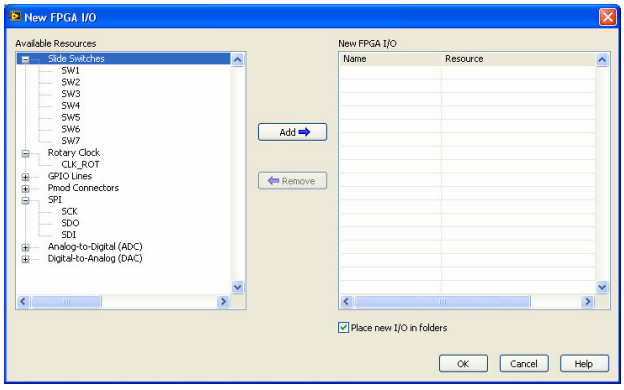


Figure 6. FPGA I/O Selection

For example, let us assume that the user wants the FPGA to monitor the status of buttons BTN0, BTN1, BTN2, and BTN3, and turn ON/OFF LED0, LED1, LED2, and LED3 to indicate buttons states. The diagram for implementing this task (Fig. 7) contains FPGA I/O objects representing buttons and LEDs that are connected to each other.

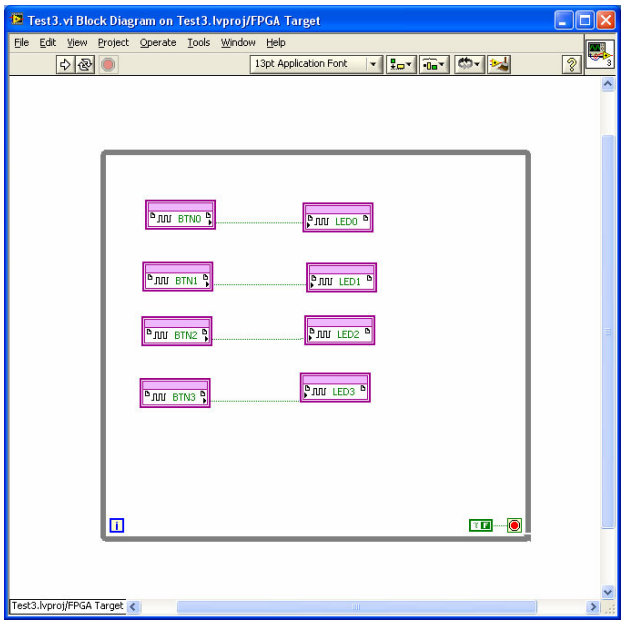


Figure 7. LabVIEW FPGA diagram using FPGA I/O objects

LabVIEW FPGA uses a resources file to associates FPGA I/O symbols with FPGA lines that are connected to the physical objects represented by the FPGA I/O symbols. The resources file, named DETB.ucf in this example, contains definitions that associate object names to FPGA lines.

```
Net "BTN0" LOC="C13" | IOSTANDARD = LVCMOS33;
Net "BTN1" LOC="D12" | IOSTANDARD = LVCMOS33;
Net "BTN2" LOC="C12" | IOSTANDARD = LVCMOS33;
Net "BTN3" LOC="C10" | IOSTANDARD = LVCMOS33;
Net "LED0" LOC="C11" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8;
Net "LED1" LOC="D11" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8;
Net "LED2" LOC="B11" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8;
Net "LED3" LOC="A12" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8;
```

This application written in LabVIEW FPGA is equivalent with the following C code based sequence written WebPack (Xilinx ISE 10.1):

```
entity main is
   Port (
        btn : in  STD_LOGIC_VECTOR (3 downto 0);
        led : out  STD_LOGIC_VECTOR (3 downto 0));
end main;

architecture Behavioral of main is

begin

led(0) <= btn(0);
led(1) <= btn(1);
led(2) <= btn(2);
led(3) <= btn(3);

end Behavioral;
```

Both programming environments use a resources file to link software defined resources with FPGA lines that control the resources.

III. LED CONTROL APPLICATION DESCRIPTION

The Digital Electronics FPGA Board has eight LEDs and eight switches. We created a LabVIEW application that generates patterns of blinking LEDs based on the position of switch SW0.

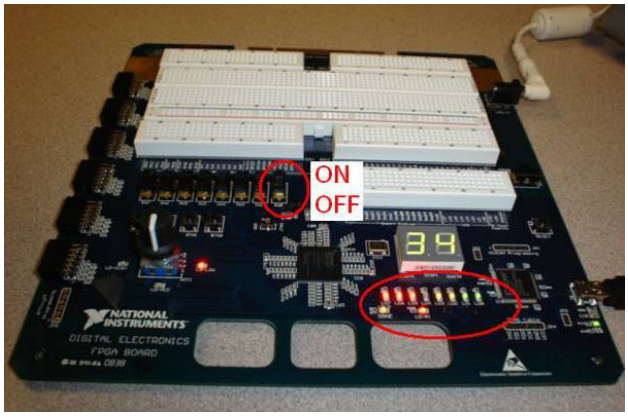


Figure 8. LEDs and SW0 on the Digital Electronics FPGA Board

The application panel displays LED blinking patterns and also the position (ON/OFF) of witch SW0.

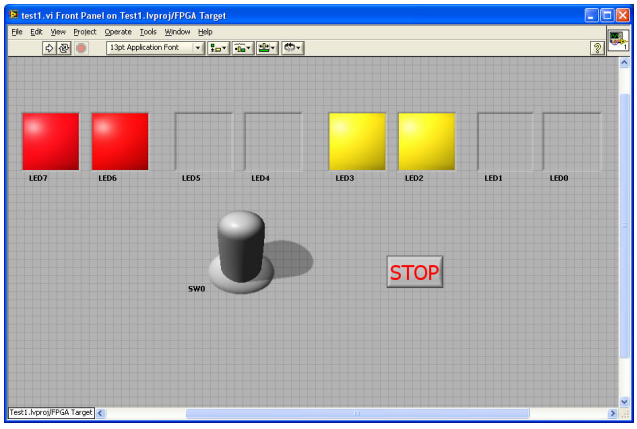


Figure 9. Application Panel

The application diagram uses FPGA I/O objects to build cases that associate different LED blinking patterns with the position of switch SW0.

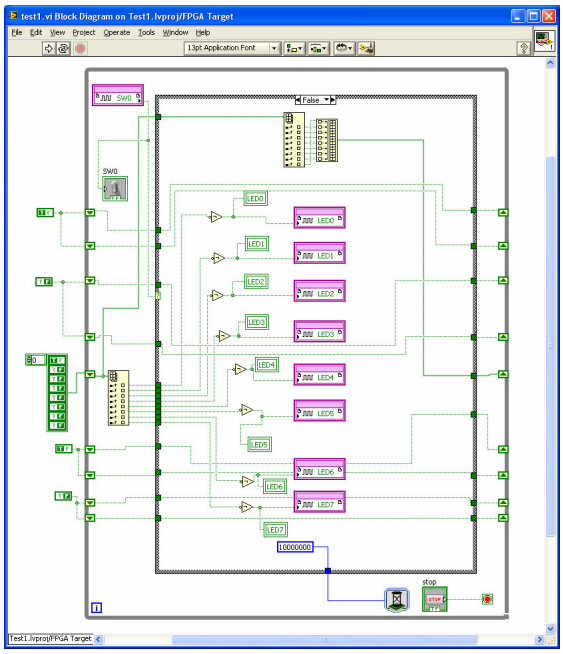


Figure 10. Application Diagram

When you click the Run button LabVIEW compiles the VI for FPGA, then generates, compiles and downloads VHDL code into FPGA memory for execution on the board.

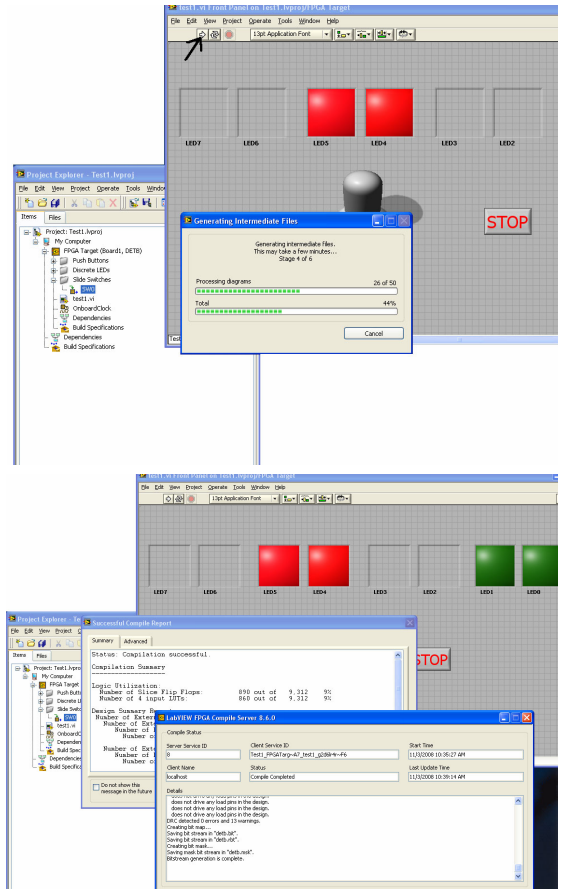


Figure 11. LabVIEW FPGA VI compilation

The Digital Electronics FPGA Board is connected to the PC via USB cable. This setup is mobile to the extent of being able to connect the board to any PC. The USB cable is needed for downloading code to the FPGA. Application monitoring can be extended from the host PC to a mobile computing platform by using a PDA.

IV. REMOTE DC MOTOR CONTROL VIA FPGA LINES

The PmodHB5 module is an interface module for robotics and other applications where logic signals are used to drive small to medium-sized DC motors. PmodH5 module features include:

- a 2A H-bridge circuit for voltages up to12V
- a JST 6-pin connector for direct connection of Digilent motor-gearboxes
- a 2-channel quadrature encoder with Hall-effect sensors to detect motor speed small form factor (0.8" x 1.30")

Figure 12. PmodHB5 module with motor attached

Let us connect the PmodHB5 board and a DC motor to the Digital Electronics FPGA Board using one of the PMOD connectors located on the far left side of the board. PMOD lines are connected to the FPGA.

We want to implement an application that does the followings:

- Code download from PC to FPGA to locally control motor via PMOD connector and PmodHB5 I/O board attached to the Digital Electronics FPGA Board.
- Control and monitor motor application running on the Digital Electronics FPGA Board, from the PC via USB cable.

Remotely control and monitor motor application running on the Digital Electronics FPGA Board, via application running on the PC, from a PDA via TCP/IP.

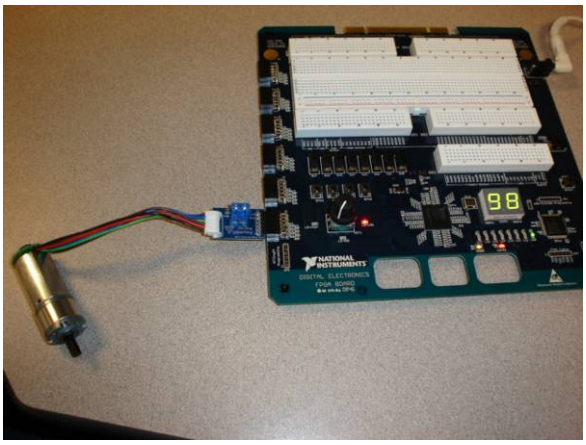


Figure 13. PmodHB5 module inserted into a PMOD connector on the Digital Electronics FPGA Board

Let us create an application project named PmodHB5.prj. Project target is FPGA Target (Board1, DETB). We will to add to the project the FPGA lines that are connected to the PMOD connector. As seen in the example before physical FPGA lines are represented in the application by Elemental I/O objects that define PMO_1 lines: J1_IO1, J1_IO2, J1_IO3, and J1_IO4. These objects need to be added to the project.

The project will contain two types of VIs:

1) A local VI that controls the motor. This VI is downloaded on the FPGA board, and it will run continuously on the FPGA board.
2) A host VI that is running on the Host Computer and allows the user to interface with the VI that is running on the FPGA board.

A third VI will be created to talk to the Host VI via TCP/IP from a PDA. This VI will allow remote/mobile motor monitoring and control via TCP/IP to Host PC and further via USB to FPGA board and further via 4-wire communication bus to motor attached to connector on the FPGA board.

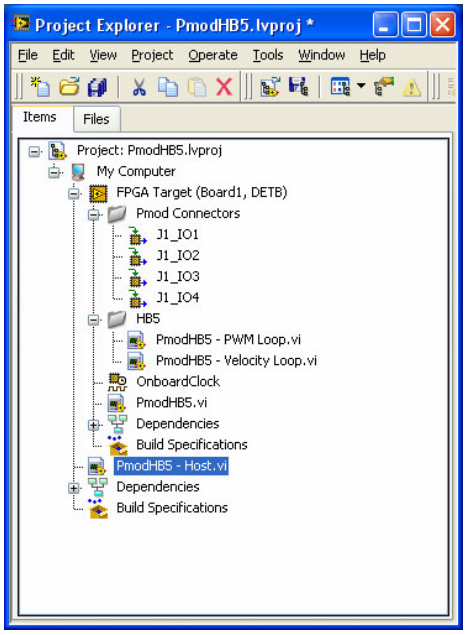


Figure 14. PmodHB5.prj explorer window

Next we need to create the local VI which controls the motor by combining the functionalities of the PWM loop and velocity loop. The local VI is named PmodHB5.vi.

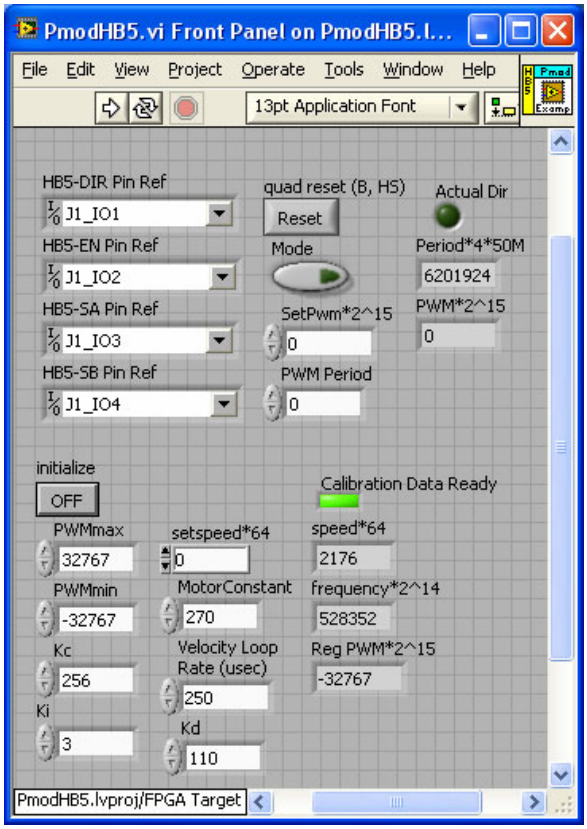


Figure 15. PmodHB5.vi Panel

Controls

HB5-DIR Pin Ref - The pin reference that corresponds to the DIR pin (pin 0) on the port that the HB5 board is plugged into.

HB5-EN Pin Ref - The pin reference that corresponds to the EN pin (pin 1) on the port that the HB5 board is plugged into.

HB5-SA Pin Ref - The pin reference that corresponds to the SA pin (pin 2) on the port that the HB5 board is plugged into.

HB5-SB Pin Ref - The pin reference that corresponds to the SB pin (pin 3) on the port that the HB5 board is plugged into.

Quad reset - Resets the counter used in the quadrature encoder reader

Mode - Selects the PWM input of the PWM Loop VI between SetPWM*2^15 (Mode = False) and RegPWM*2^15 (Mode = True).

SetPWM*2^15 - Motor speed on a scale of -32768 to 32767. 32767 represents full speed forward -32768 represents full speed backwards.

PWM Period - The period of the PWM signal in ticks.

Motor Constant - Motor constant as defined by the manufacturer of the motor.

Setspeed*64 - Speed to set the motor in RPM

Velocity Loop Rate - Sampling interval in microseconds

Initialize - Initializes the PID control VI

PWMmax - The maximum PWM to be used by the PID VI.

PWMmin - The minimum PWM to be used by the PID VI.

Kc - Proportional gain for the PID VI.

Ki - Integral gain for the PID VI.

Kd - Derivative gain for the PID VI.

Indicators

Actual Dir - Actual direction that the motor is turning.

Actual Dir = 1 means the motor is turning forward and

Actual Dir = 0 means the motor is turning backwards.

Period*4*50M - Actual period of the PWM. Used to determine speed feedback for the Velocity Loop VI.

PWM*2^15 - Actual PWM signal chosen between SetPWM (control) and RegPWM (indicator).

Frequency*2^14 - Frequency of the PWM signal

Speed*64 - Speed of the motor in RPM

Reg PWM*2^15 - Regulated motor speed on a scale of -32768 to 32767. 32767 represents full speed forward and -32768 represents full speed backwards.

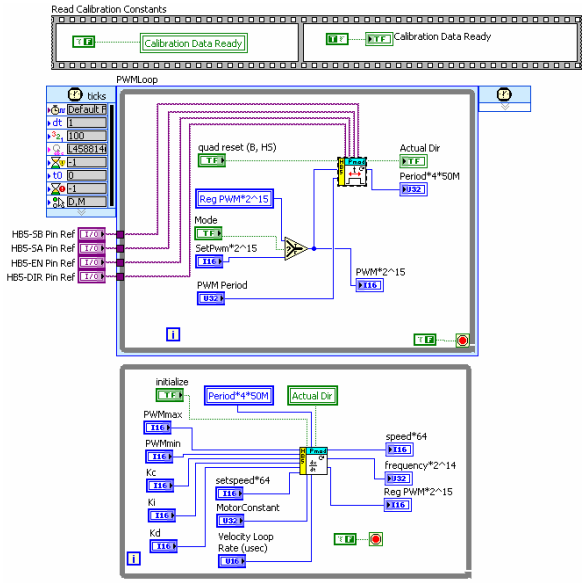


Figure 16. PmodHB5.vi Diagram

The Host vi is named PmodHB5 - Host.vi

This VI is used on the host computer to control PmodHB5.vi using intuitive controls and graphs of the PWM and speed over time.

Controls

Set Speed - Speed to set the motor in RPM

Set PWM - Speed to set the motor on a scale of -32768 to 32767. 32767 represents full speed forward -32768 represents full speed backwards.

Mode - Selects the PWM input of the PWM Loop VI between SetPWM*2^15 (Mode = False) and RegPWM*2^15 (Mode = True).

Sampling Interval - Sampling interval in microseconds

Proportional Coeff Kc - Proportional gain for the PID VI.

Integral Coeff Ki - Derivative gain for the PID VI.

Derivative Coeff Kd - Integral gain for the PID VI.

Pulses Per Rotation - Number of pulses per rotation produced by the Quadrature Encoder on the motor

Gear Ratio - Gear Ration of the motor

Stop - Stops the VI from running

Indicators

PWM - Current actual PWM from -1 to 1

Actual Speed - Current actual speed of the motor in RPM

Speed Chart - Plot of actual speed over time

PWM Chart - Plot of actual PWM over time

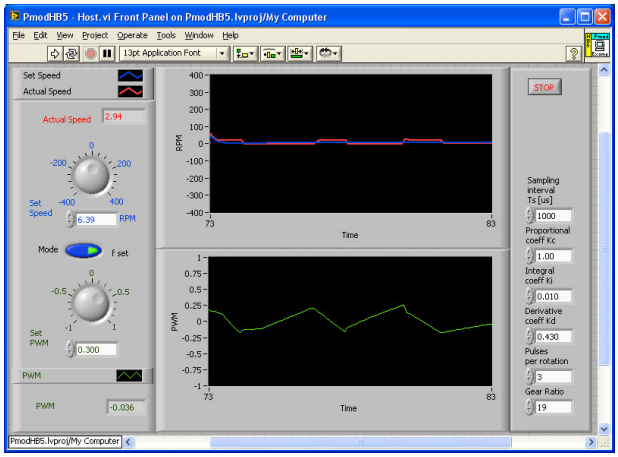


Figure 17. PmodHB5 - Host.vi Panel

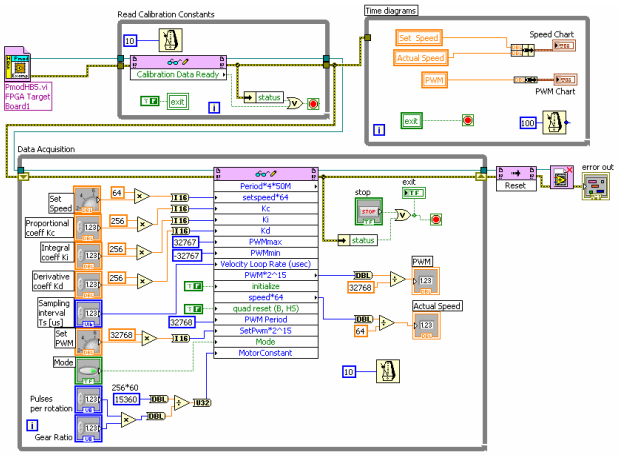


Figure 18. PmodHB5 - Host.vi Diagram

We will first run the local application, PmodHB5.vi in order to compile and download to FPGA memory on the Digital Electronics FPGA Board.

Then we run PmodHB5 - Host.vi on the host computer, in order to control the motor via USB. The third step is to start the remote application on the PDA and modify to the host application to get commands and send data via TCP/IP from/to the remote PDA VI panel.

V. PDA MOBILE MONITORING AND CONTROL (UNDER DEVELOPMENT)

Mobile monitoring and control of applications running on a host PC can be done by using the LabVIEW PDA Module.

LabVIEW PDA extends the LabVIEW development environment to applications for PDA handheld devices that are connected to the host PC via USB, Serial, or Wireless (ex: Bluetooth, WiFi). We have developed a remote application that is running on a PDA device and communicates via TCP/IP with and Access Point that communicates via Ethernet with the host application running on the PC which communicated via USB with the Digital Electronics FPGA Board which communicates via a PMOD interface with the I/O attachment board which is connected to the motor via a 6-wire power and communication bus.

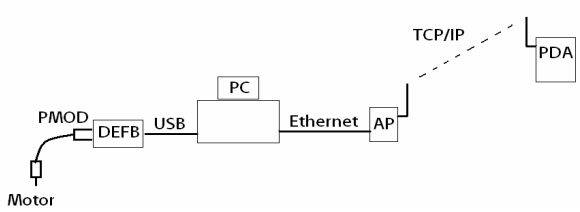


Figure 19. Remote DC Motor control Application Setup

The PDA client application connects to the host server application by using the host computer Server IP address.

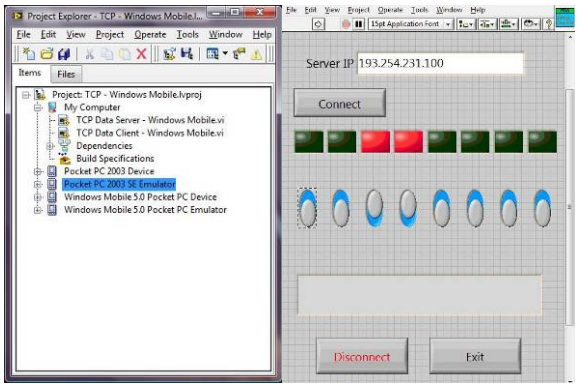


Figure 20. PDA Application Panel

We have incorporated a TCP Data Server VI in PmodHB5 - Host.vi which runs on the host computer. Then PDA application is running a TCP Data Client VI which uses the IP address of the host (server) machine to communicate with the data server.

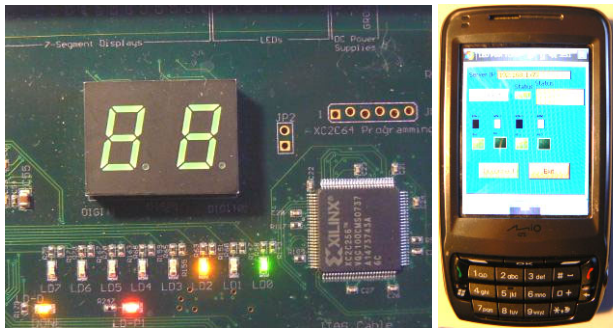
In Fig. 21 we present the application corresponding with the task presented in Fig. 7.

For this application we prezent in the Fig. 22 the Client-Server LabVIEW development, who was used in all our examples.

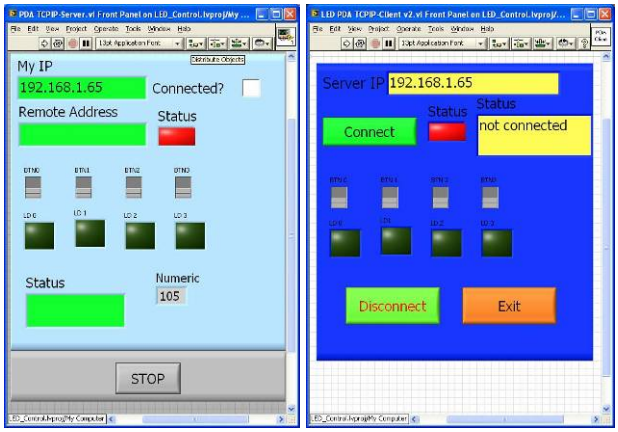
This development was done using a LabVIEW Project who incorporates all the applications and the necessary targets:

- PC TCP-IP Server
- PDA TCP-IP Client
- Pocket PC Emulator
- Pocket PC Device
- FPGA Target

By this Client-Server structure we can easy do Remote Control of this FPGA board using any PDA or SmartPhone.

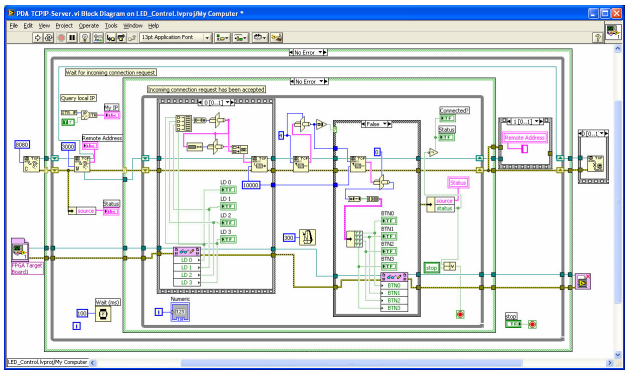


(a) The LED Monitorig using one MIO PDA- DigiWolker

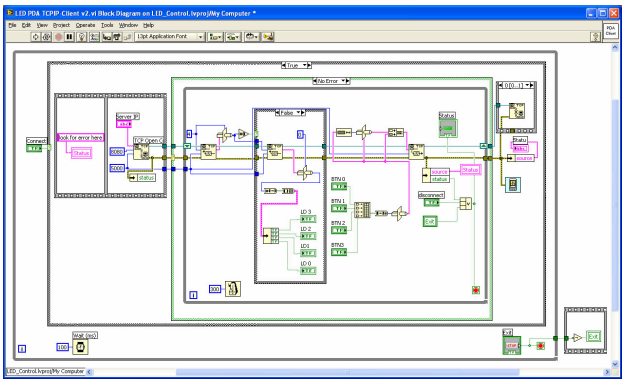


(b) Server (c) PDA Client

Figure 21. The Client-Server PDA control of FPGA board



(a) Server (on PC) Diagram



(b) Client (on PDA) diagram

Figure 22. The Diagrams for Server/Client Application

VI. CONCLUSIONS

This paper introduces the concept of graphical FPGA design and control using the Digital Electronics FPGA Board developed by National Instruments for the NI-ELVIS platform.

The idea to combine DMM, Oscilloscope, Function Generator, etc., capabilities with the programming and routing capabilities of a 1M gates FPGA is supported by the need in training laboratories for Lifelong Learning activities to cover complex applications and also give students access to new technologies.

FPGA's allow systems to be configured and reconfigured for many applications making the Digital Electronics Trainer Board a versatile tool for monitoring and controlling local and remote I/O devices.

REFERENCES

[1] LabVIEW FPGA, www.ni.com

[2] Doru Ursuțiu: "Inițiere in Lab VIEW", Lux Libris, 2001

[3] Hans van t Spijker, Petru Cotfas, Doru Ursuțiu "NI ELVIS Development of a Low-cost Potentiostat for Research Applications and e-Lerning", Second Int. Simposium on Remote Engineering and Virtual Instrumentation REV2005, Brasov – Romania, 30 June -1 July

[4] Jinga Vlad, Oros Ramona Georgiana: License Degree Paper "Laborator virtual de Telecomunicații", presented in June 2008 at the Electrical Engineering and Computing Science Faculty – Transilvania Univ. Brașov

[5] Doru Ursutiu, Cornel Samoila, Marius Ghercioiu, Graphical FPGA Programming and Monitoring, 4-th Int. Conf. on Interactive Mobile and Computer Aided Leraning, IMCL 2009, 22-24 April, Amman – Jordan, 2009

ACKNOWLEDGMENT

We would like to thank National Instruments for the NI Digital Electronics FPGA Board (DE FPGA) and Digilent Inc. for their Pmod boards offered for the DE FPGA board applications.

AUTHORS

Ursuțiu Doru (udoru@unitbv.ro) – Transilvania University, Technical Engineering Faculty, Braşov, România

Marius Ghercioiu (marius.ghercioiu@ni.com) – National Instrumensts – USA

Cornel Samoila , (csam@unitbv.ro) – Transilvania University, Technical Engineering Faculty, Braşov, România

Petru Cotfas (pcotfas@unitbv.ro) – Transilvania University, Technical Engineering Faculty, Braşov, România

Manuscript received 22 March 2009. Published as submitted by the authors.