Middle-East Journal of Scientific Research 24 (S1): 339-345, 2016
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2016.24.S1.10

Smart Phone Guided Vehicle for Industrial Material Transport

[1]R. Supritha, [2]K. Arun Venkatesh and [1]G. Gugapriya

[1]School of Electronics Engineering - SENSE, VIT University, India
[2]Product Development Division, Yashika Industries, Chennai, India

Abstract: The dominance of automation in the industrial sector is rapidly increasing. Industries always prefer highly reliable automated equipment. A range of vehicles are being used for industrial material transport which can be replaced by Automated Guided Vehicles (AGV). The design of AGV navigation system should comply with the challenges which will be faced within the closed premises of an industry. A design of smart phone guided vehicle to carry the materials within industrial premises is proposed in this paper. A serial command driven navigator algorithm is developed.

Key words: Automated Guided Vehicle (AGV) · Navigation System · LabVIEW · sbRIO 9626 · NI VISA

INTRODUCTION

An autonomous vehicle is becoming the need of the hour in several industrial premises. It is required to reduce the time taken in manufacturing a product in industries. The reduction in time is mainly due to autonomous behaviour of the vehicle. If a job has to be transported from one place to another in the industrial premise, the job can simply be placed on an autonomous vehicle and it will be delivered to the end position in the optimised path. If the job is heavy, it makes the transportation process tedious had it been done manually. The optimization of production in trade and service sector is achieved mainly by the automation of transportation. Special applications are provided by AGV's for handling this task. Mainly these systems are the centrally controlled automated guided vehicles (AGV). During the dawn of industrial automation, optical and inductive guidelines were used to guide the vehicle. One of the difficulties in guiding a vehicle automatically is the inflexibility in modifying the change in path The AGVs are designed mostly for a specific application. They are always designed for a particular application which makes it unique in its own way. In the proposed system LabVIEW FPGA (FIELD-PROGRAMMABLE GATE ARRAYS) based sbRIO 9626 has been used.The large number of gates in FPGAs has functions which are defined while programming them. The programming provides a function which interconnects these gates. The semiconductor switches are turned on and off dynamically with the help of wiring list in the FPGA programming. Hence, the connections are modifiable in the field. The demand of FPGA is always high because of its moderate cost, high precision and high operating speed and easily reconfigurable. Even though, designing and debugging FPGAs in classical methods does not make production quicker. The function of FPGAs' depends on the configuration of hardware. In this way the FPGA is different from microprocessors, whose programs are described in the fixed hardware structure. Writing programs on a FPGA is similar to designing hardware for a very unique application. Extensive knowledge in VHDL development and similar complex programming languages is required to use FPGA. This has eventually reduced several engineers to not step ahead with FPGA [1] [2]. The problem faced by engineers in using the conventional FPGA coding methods is overcome by using Graphical language LabVIEW FPGA. In LabVIEW FPGA, the block diagram is created and is downloaded in to the FPGA [3].

Navigation System for Agv with Serial Commands: A Navigator System in an automated vehicle guides it to travel in a prescribed path. In the early days navigation system used magnetic spot guidance methods. In this type of navigation, the vehicle moves along a magnetic strip [4]. After encountering a few problems due to reduced flexibility in changing the path dynamically and thus controlling the vehicle, several other technologies took over the existing systems. One such navigation

Corresponding Author: R. Supritha, School of Electronics Engineering - SENSE, VIT University.

system which has produced an efficient outcome is the vision based navigation system [5]. A novel navigator algorithm for an automated vehicle which shall be driven by serial commands is proposed in this paper. The user provides serial commands to the controller in sequence, which navigates the AGV. Proposed navigation system has a controller which is programmed by LabVIEW to navigate the vehicle. The advent of graphical programming is making the job easier for engineers. The controller usedis NI sbRIO 9626. NI sb RIO 9626 is of industrial standards. The controller has a built in FPGA and RT processor. It can be programmed in LabVIEW FPGA and LabVIEW RT. Unlike the conventional FPGA programming which requires a lot of learning process to program a system, the graphical programming like LabVIEW FPGAis much easier to learn. The LabVIEW FPGA program generates bitfiles which are dumped into the FPGA Chips in the NI sbRIO 9626. The program generated can be used as an application on a mobile phone. This is done by creating. exe (executable) file of LabVIEW application program.

Bock Diagram: Navigation System provides an aid to travel in the path in which an AGV has to travel. In the proposed algorithm for navigating the AGV from one position to another, the AGV moves in a pre-defined path. The user provides serial commands to the controller in sequence, which hence navigates the AGV dynamically. The controller used is NI sbRIO 9626. NI sb RIO 9626 is of industrial standards. The controller has a built in FPGA and RT processor. It can be programmed in LabVIEW FPGA and LabVIEW RT. The advent of graphical programming is making the job easier for engineers. Unlike the FPGA programming which requires a lot of learning process to program a system, the graphical programming like LabVIEW FPGA is much easier to learn. The LabVIEW FPGA program generates bitfiles which are dumped into the FPGA Chips in the NI sbRIO 9626. The program generated can be used as an application on the mobile phone. Various pre defined paths inside an industry is fed into the system. The system receives start node and end node. This path is provided to the user with the corresponding distance to travel. The system is for now designed to transmit around a pre-defined path. Fig. 1 shows the data flow diagram of the serial command based navigator system.

Architecture of sbRIO Applications: FPGA based platforms developed by National Instruments are compatible with the industrial standard measurements.

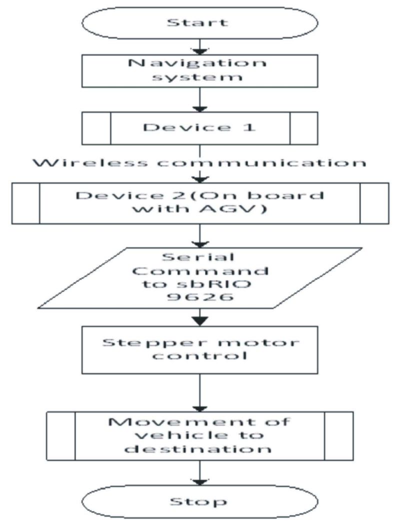

Fig. 1: Flow Diagram of Navigation System for AGV using LabVIEW

The measurements in the real time can be obtained with a higher precision. There are three levels of programming in LabVIEW-FPGA, RT and windows. The availability programming pallets with different functionalities are more in the LabVIEW windows. The LabVIEW RT has comparatively less pallets. The LabVIEW FPGA has only the most important palettes. The communication between RT and FPGA occurs via the bitfile. The bitfile is generated while the FPGA program is deployed into the system. The bitfile generated by LabVIEW FPGA program can be either given as a reference in host VI or can be directly downloaded to On board flash memory of sbRIO. The FPGA compilation and deployment types are shown in Fig. 2(a) and 2(b) respectively. The bitfile is downloaded to sbRIO when there is no requirement of data logging in the application. The bitfile is also known as personality. The size of flash memory in NI sbRIO 9626 is 256 MB.

The Script based Navigation System: The script based Navigation System for AGV consists of a sequence of commands to direct the system to move along a prescribed path. The data acquisition and control is done in sbRIO [6]. Basically there are three vehicle movements

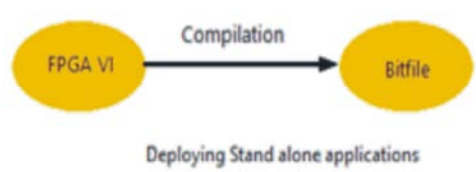

Fig. 2(a): FPGA Deployment

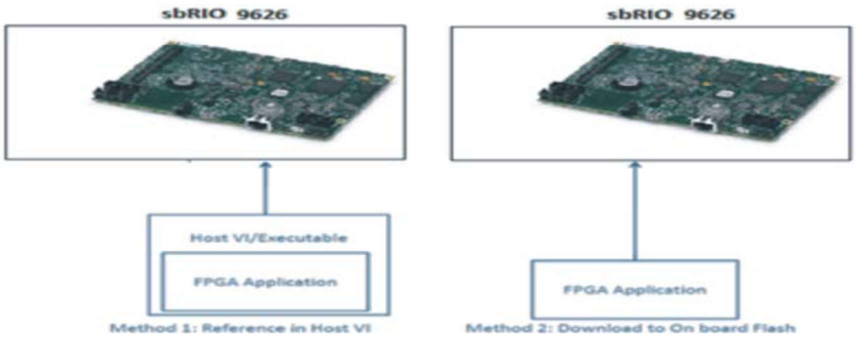

Fig. 2(b): FPGA deployment

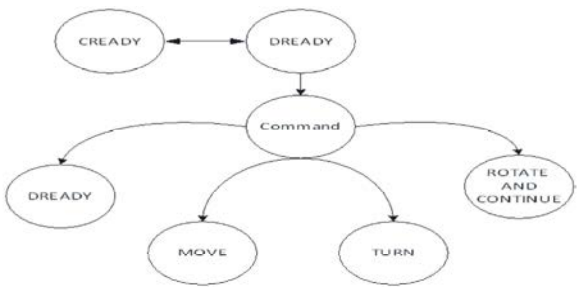

Fig. 3: State Diagram for Serial commands between controller and LabVIEW application

which the AGV will perform in the proposed navigator system which are: move forward, move backward and turn. CREADY (Controller Ready) and DREADY (Device Ready) are like handshake commands between the controller and the device to start the communication. The device is the one which is providing the serial commands to the controller to traverse along the optimized path. Fig 3 describes the state diagram for serial commands used in the system.AGV motion is controlled by few common serialcommands through the serial port in controller. The instructions include TURN, REVERSE, FORWARD and RESET. The angles to rotate and the distance to move are specified along with the command. The system is to be programmed to move the AGV corresponding to the serial commands. When there is no command provided, the system should automatically reset. The NI stepper motor is to be programmed to act according to the incoming serial commands. Each stepper motor has an encoder which provides the rpm. This rpm can be changed to navigate the two wheels. The direction of rotation can also be changed in the NI ISM stepper motor driver.

Distance Calculation Using Pi, Diameter and Angle of Two Stepper Motors: The stepper motors are digital devices that are operated by pulse inputs. The motion control of a stepper motor is free from drift and accumulative error. In the proposed application, two stepper motors (NI ISM 7402 E) have been used to navigate the AGV. The two motors are controlled in FPGA, which provides the real time data accurately [7]. The signals to these motors are controlled by a stepper motor driver which provides input frequencies corresponding to the required steps. Real time data comes from NI ISM 7402E. This value is compared with the value given by the user in the serial command. Even though the steps are driven by the stepper motor driver in NI ISM 7402E unit [8], the encoder data increases the synchronisation and precision. The calculations to move and turn the AGV as per the incoming serial command are discussed below.

- Calculations to move the AGV along a linear path:

a. $Circumference = 3.14 \times Diamet$

b. $\frac{Distance}{Circumference} = Rotati$

c. $(Max\ Machine\ Speed)\frac{Inch}{Minute} \times (Stepper\ motor)\frac{Step}{Inch} \times \frac{1\ Minute}{60\ Second} = \frac{Step}{Second}\ (H$

d. $Rotation \times \frac{Step}{Second} = Frequency\ of\ motor\ puls$

The same frequency is given to the two motor pulses of two respective motors in the AGV. The direction of movement is also controlled.

Calculations to Turn the AGV:

a. $Circumference = 3.14 \times Diamet$

b $Circumference\ of\ robot\ turning\ both\ wheels = Distance\ between\ two\ wheels \times 3.$

c. $\frac{Arc\ Segment}{Wheel\ Circumference} = Rotati$

$$d \quad Rotation \times \frac{Step}{Second} = Frequency\ of\ motor\ puls\cdot$$

To turn the AGV in a particular degree, the direction of rotation of two wheels should be opposite to each other. The motor pulses are respectively given to the two motors while they are rotating in the opposite directions [9]. The angle to turn the AGV should be given along with the "Turn command". Now when the angle is given, the corresponding differential speed between the wheels is to be calculated for the given time. If the AGV has to travel straight or reverse distance, the distance to travel in meters should also be specified along with "Move command".

Software Architecture of the Navigation Sytem Using Serial Commands: The software used in the proposed navigator system is LabVIEW. LabVIEW is virtual instrumentation platform with graphical user interface. Virtual instrumentation provides highly sophisticated instruments which can be used in control and automation applications. It integrates a sensor, actuator, hardware and software and provides a flexible virtual instrument. The IO ports in sbRIO 9626 are controlled and monitored by LabVIEW virtual instrument programming. The IO list for the proposed navigator system, in sbRIO 9626 is given below.

DIO List for the Navigation System in sbRIO 9626

MOTOR 1	
Input	Type
DIO 0	Step Signal
DIO 1	Direction
DIO 2	Enable
MOTOR 2	
Input	Type
DIO 3	Step Signal
DIO 4	Direction
DIO 5	Enable
MOTOR 1 Encoder Output	
DIO 9	Encoder Output (B)
DIO 10	Encoder Output (A)
DIO 11	Encoder Output (index (z))
MOTOR 2 Encoder Output	
DIO 12	Encoder Output (B)
DIO 13	Encoder Output (A)
DIO 14	Encoder Output(index (z)) Inputs
DIO 24	Move
DIO 25	Forward/Reverse
DIO 26	Emergency Stop Button

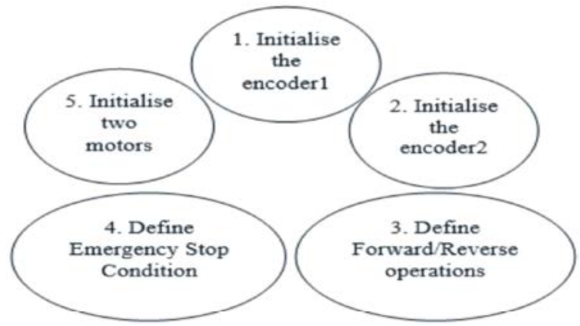

Fig. 4: Modules in Lab VIEW FPGA Program for the navigation system

FPGA Programming for the Navigation System: The digital controls and indicators, for various digital IO's listed above, are assigned in Lab VIEW FPGA program [10]. The direction of motor rotation is controlled in the FPGA. The index value of encoder is used in the navigation of the system by dead reckoning method. The value to travel is fed into the system and corresponding action is taken in the FPGA. Encoder values are given as a reference to the RT VI. When emergency button is pressed the whole circuit becomes an open circuit momentarily and hence the system should halt. For the encoder, a counter counts the number of pulses in channel A, B and Z (index). In RT VI the real time values of distance travelled by AGV is compared with the index pulses coming from encoder. The program modules present in Lab VIEW FPGA program is represented in Fig. 4.

If the required measurement is reached, the AGV stops moving and waits for the next serial command. The minimum speed required to move the vehicle, the steps per inch are all calibrated and initialized in the FPGA VI. Fig. 5 shows a sample FPGA code for encoder.

RT Programming for the Navigation System:

- The RT VI consists of following four cases:

The cases in Lab VIEW RT program is shown in Fig. 6. The first state i.e., "Init", initializes FPGA controls and indicators to default values. The motor pulses of two wheels are initialized to zero. The position of the two wheels is also initialized to zero. The motor operations like turn, turn reverse, forward/reverse and motor start/stop are all initialized to Boolean FALSE. The next case is "Poll". Poll is the default case. It checks the status of device continuously. The poll case and start case keeps on running until the controller sends the serial commands.

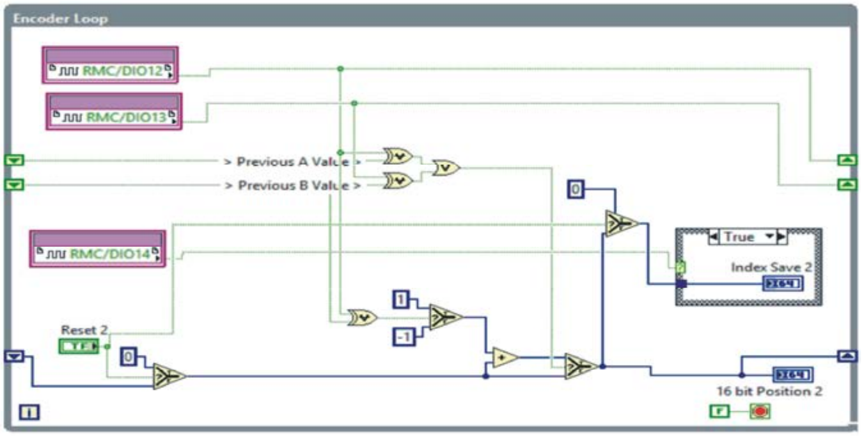

Fig. 5: Encoder loop in FPGA VI

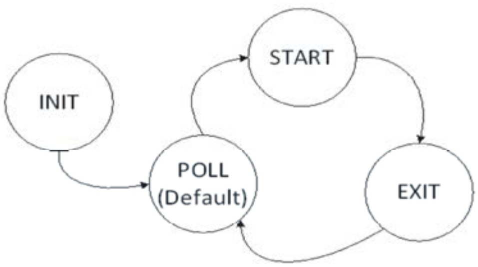

Fig. 6: States in RT VI

If the controller does not send any command the "Exit" case is executed where the application is stopped. If the digital input 'start' is true, then the "Start" case gets the input about which COM PORT will be used for the serial communication between the two pre-defined serial ports in NI sbRIO 9626. If the digital input start is false, the motor operations like turn, turn reverse, forward/reverse and motor start/stop are all initialized to Boolean FALSE i.e., the application stopped. The Start case has the serial parser sub VI. A sample code in Lab VIEW RT is given in Fig. 7.

The Serial Parser sub VI: The serial parser sub VI is written in Lab VIEW RT. It does the main operation in serial communication. The serial commands coming from the RS232 ports are sorted out and the corresponding sequence of operations are done to navigate the system. The cases in this VI are: Dready, Command, Move, Turn, Reset, Rotate and Continue. The command case is the default case in case structure. The controller sends CREADY (Controller Ready) via serial communication. When this VI receives Cready, Dready (Device Ready)

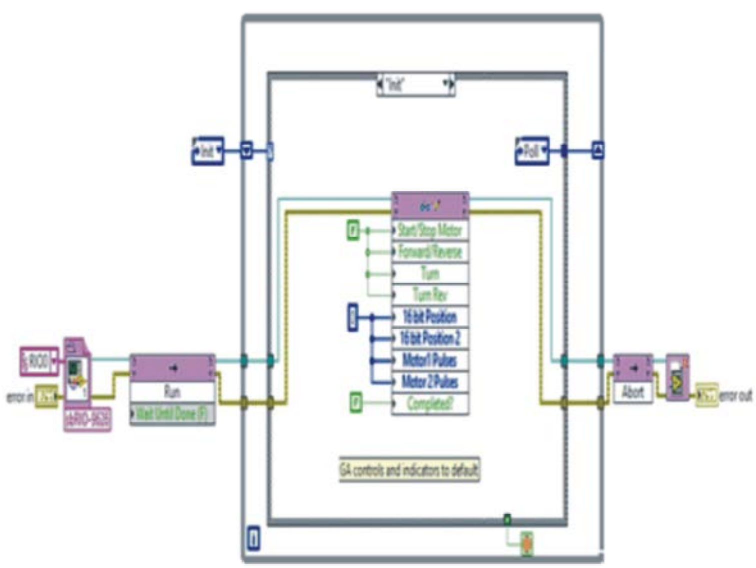

Fig. 7: INIT case in RT VI

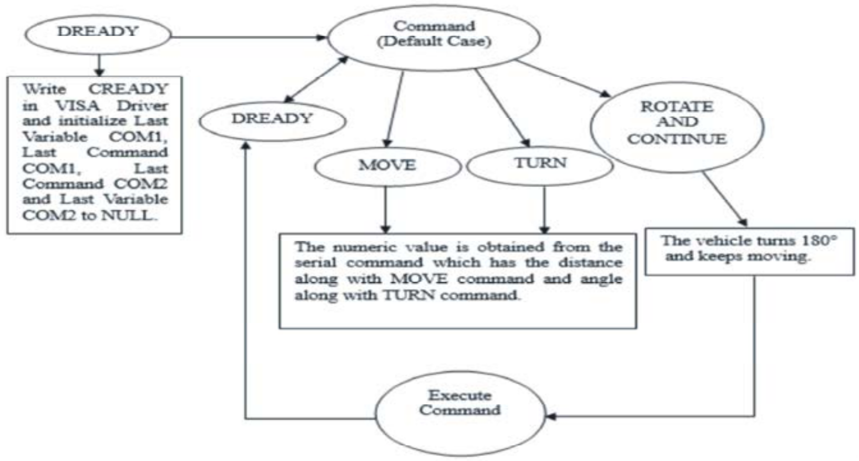

Fig. 8: States in serial parser VI

is sent to controller as an acknowledgement. The communication between controller and application begins after the controller receives DREADY. The user sends serial commands like Move 300, Turn 90, Move 40 etc. After executing a command, the CREADY command is again sent to the device until the next command is serially passed from the application. Fig. 8 shows the cases in serial parser VI.

VISA Driver sub VI: The VISA Driver sub VI consists of the open, read, write and close operations of serial communication.

CONCLUSIONS AND FUTURE WORKS

The system designed is controlled by user with serial commands. The system can be advantageous in applications where the vehicle has to travel in a pre defined path.

The device1 sends data to device on board wirelessly. The data from device2 which is on board is connected serially to NI sbRIO 9626. The data received is manipulated and the vehicle starts moving as per the guidance of user.RMC connectors are soldered with the controller and required power supply units are also provided. The two NI ISM 7402E modules are connected to two opposite wheels. The Module is placed beneath sbRIO. The serial communication between the controller and AGV has been established till now and the wireless communication between another device and the tablet on board is yet to be established. The system has to be updated with an autonomous path planning system which will allow the system to evade and take an alternative path during encounter with an obstacle. Fig. 9 and Fig. 10 shows the final AGV with a closed chassis.

Fig. 9: Top View of the vehicle inside a chassis

Fig. 10: Top View of the vehicle with the chassis covered

Table 1: Result of the distance travelled by the vehicle in Real Time

Input Distance to vehicle by user(in m)	Distance travelled by vehicle in Real Time(in m)	
	Iteration 1	Iteration2
30	29.9	30
100	100.5	99.8
50	50	49.5

Fig. 11: AGV with a smart phone in the holder navigating it.

Table 1 below shows the result of the comparison between given input and obtained real time output.

ACKNOWLEDGEMENT

The author would like to thank and appreciate all those who have been a helping hand towards the completion of the work.

REFERENCES

1. Pedro Ponce-Cruz, Arturo Molina and Brian MacCleery, 2016. "Fuzzy Logic Type 1 and Type 2 Lab VIEW FPGA Toolkit", © Springer International Publishing Switzerland 2016 P. Ponce-Cruz *et al*., Fuzzy Logic Type 1 and Type 2 Based on Lab VIEW™ FPGA, Studies in Fuzziness and Soft Computing, DOI 10.1007/978-3-319-26656-5_2.
2. Pedro Ponce-Cruz, Arturo Molina and Brian MacCleery, 2016. "Lab VIEW™ FPGA", © Springer International Publishing Switzerland 2016 P. Ponce-Cruz *et al*., Fuzzy Logic Type 1 and Type 2 Based on Lab VIEW™ FPGA, Studies in Fuzziness and Soft Computing, DOI 10.1007/978-3-319-26656-5_2.
3. Sumathi, S. and P. Surekha, 2007. "Introduction to Virtual Instrumentation", Lab VIEW based Advanced Instrumentation Systems-Springer Berlin Heidelberg.
4. Sok-Yong Lee a,n, Hai-Won Yang, Navigation of automated guided vehicles using magnet spot guidance method, Robotics and Computer-Integrated Manufacturing, Elsevier.
5. Quoc, V. Do, Peter Lozo and Lakhmi C. Jain, "Vision-Based Autonomous Robot Navigation", Industrial Technology, Chapter Innovations in Robot Mobility and Control Volume 8 of the series Studies in Computational Intelligence pp: 65-103, Springer Berlin Heidelberg.
6. NI sbRIO datasheet-http://www.ni.com/ pdf/ manuals/374175a.pdf as on 14-03-2016.
7. Jabeen Electr, S. and Comput Eng, North Dakota State Univ, Fargo, ND, USA S.K. Srinivasan, S. Shuja and M.A.L. Dubasi, 2015. "A Formal Verification Methodology for FPGA-Based Stepper Motor Control", IEEE Embedded Systems Letters, (Volume:7, Issue: 3), Issue Date: Sept. 2015.
8. NI ISM 7402 datasheet - http://www.ni.com/pdf/ manuals/373378d.pdf as on 14-03-2016
9. Dineshkumar, U., M. Nisha and N. Mathivanan, 2014. "Design of a Differential Drive Mobile robot and Remote Tracking of Robot using Odometry and WSN Nodes", International Journal of Mechatronics, Electrical and Computer Technology, 4(11): 384-399. ISSN: 2305-0543 Available online at: http://www.aeuso.org.
10. Sanket Mehta, Nirmal Parmar, Jayanee Soni and Arpita Patel, 2016. "Data Acquisition with FPGA Using Xilinx and Lab VIEW", © Springer Science Business Media Singapore 2016, S.C. Satapathy *et al.* (eds.), Proceedings of International Conference on ICT for Sustainable Development, Advances in Intelligent Systems and Computing, pp: 408.