A Comparative Analysis of a Navigation Platform Designed in Arduino and FPGA

Aarsha Anil S and Rajesh Kannan Megalingam

Abstract—**The paper carries out a comparative analysis of a fixed path navigation platform designed in both Arduino and FPGA in terms of accuracy, complexity and error percentage. The platform is designed using a rotation encoder as the distance estimation device with motors and drivers controlled by an overall controller which could either be a microcontroller or an FPGA. The rotation encoder used for both designs consists of a MOC7811 encoder IC, which translates number of pulses into distance required. Other common components are wheelchair motors and their corresponding drivers. The test device used is a normal wheelchair. The device was designed to move through three different paths with fixed paths between each room programmed into the controller. These paths were kept constant for both the Arduino microcontroller and the FPGA. The results were thoroughly analyzed for each controller and the observations recorded.**

Index Terms—**Rotation Encoder, Fixed Path Navigation, FPGA, Arduino, Distance Estimation, MOC7811**

I. INTRODUCTION

Autonavigation is an upcoming concept full of potential. Technological advancements in this field have been promising and already automated navigation devices are becoming a reality. Navigation services are needed in two cases Indoor and Outdoor Navigation. While Outdoor Navigation services have considerably advanced, the same cannot be said about Indoor Navigation systems in terms of commercialized products. Here we attempt to conduct a comparative analysis of performance of a navigation platform designed in both the Arduino UNO and FPGA. The design in this paper follows fixed path Navigation protocols. Hence an Automated Navigation Platform which can guide itself to a given location by following pre-programmed paths is proposed. This is the best case of navigation since the error percentage of fixed path navigation is much lower. The design also aims to create a platform which is cost effective, consumes less power and can be durable. Hence we analyze both designs for the same. To achieve this, primary component used is an FPGA (Field Programmable Gate Arrays)/Arduino. There are many advantages in using an FPGA as opposed to a processor or controller, which we found out during the course of the project. As it would effectively reduce any delays in execution and hence increase the overall efficiency of the system.

Aarsha Anil S is with the Electronics and Communication Engineering Department, Amrita School of Engineering, Kollam, Kerala, India (Email:aarsha.anil@gmail.com)
Dr. Rajesh Kannan Megalingam is with the Electronics and Communication Department, Amrita School of Engineering, Kollam, Kerala, India (Email:megakannan@gmail.com)

Since logic cells in the FPGA are programmable, it gives the device a higher degree of versatility in terms of user accessibility. Additionally it can process larger blocks of data in fewer clock cycles and hence would be an ideal solution. Also optimization levels of FPGA are much higher. The final product can be easily enhanced to Application Specific Integrated Circuit (ASIC) designs which are high performance embedded systems. But complexity of FPGA designs is high as compared to Arduino. Hence in terms of ease of design Arduino is much easier because it has a lot of built in modules and also is easily reprogrammable.

In the following sections we discuss the design in detail. Section II discusses the basic motivation behind the design and its applicability to the real world. Section III lists the related developments regarding the mentioned design. In section IV, we explain the implementation of hardware and software sections of our design. Section V records the observations made during our experiments and finally we conclude our paper in section VI.

II. MOTIVATION

In this paper we attempt to conduct a comparative analysis between the same navigational platform designed in both FPGA and Arduino. The motivation behind this being to decide which device would be suitable to create a more efficient design. Even though there are a lot of new age technologies such as Wi-Fi, Bluetooth based, IR rays, RFID based etc. for indoor navigation and GPS based methods for outdoor navigation purposes, all of these suffers from some demerits in one form or other. Like in case of GPS systems, they are highly susceptible to noise. Also other systems like RFIDs have high installation cost per unit area as compared to the design proposed here.

The paper discusses a fixed path navigation platform designed in both Arduino and FPGA platform, and compared for better operation with the end goal being a commercially marketable design which is easily affordable. In FPGA coding is done in Verilog, while Arduino follows C programming. Coding ease was found to be higher in Arduino whereas Verilog coding offers more flexibility and addition of future components is easier

III. RELATED WORKS

The different technologies of navigation, both indoor and outdoor, include RFID based, Wi-Fi , Bluetooth based systems etc. RFID based navigation have been implemented in both indoor and outdoor navigation applications. RFID tags based identification and navigation for outdoor fields have been discussed in [1]. Here the RFID reader reads the tags located at

978-1-5090-0396-9/16/$31.00 ©2016 IEEE

the grid of nodes where each node has a particular RFID tag which represents a specific location inside the area of interest. The two components are an RFID reader and a number of RFID tags distributed across the intended field. Location and orientation information of each node is stored in these RFID tags. The fact that we require a RFID tag for each node presents it selves to be a disadvantage. Since with increase in plot size the number of tags to be used increases, which leads to an overall increase in installation charges of the system. Also damage in even one of the nodes would affect the overall accuracy of the system.

Fuzzy logic control system has also been implemented for wheelchair motion control designed in an FPGA platform in [2]. In use a fuzzy logic feedback and control mechanism to rectify misalignment in wheelchair controlled by a joystick. Multi robotic navigation systems have also been designed in FPGA platform [3] with a start point and end point designed with IR sensors and controllers. Other types of wheelchairs [4] which utilize retina control or even nervous system control are also in development. In such system visual information is gathered from facial data and processed to give control signals. But this requires large amount of data base and calculations which makes the system tedious to implement.

Voice based motion detection and control systems [5] can be used if degree of disability of individuals has reached a point where hand movement is not possible. The system uses FPGA platform to process the voice signals. The system is integrated with IR signals for path finding and Ultrasound sensor for obstacle detection. This system also needs large amount of databases and calculations.

IV. DESIGN AND IMPLEMENTATION

The design uses a Digilent BASYS 2 Spartan 3E FPGA board for overall control of the system or an Arduino with an ATmega 328 microcontroller, a rotation encoder, motors and corresponding motor drivers. The FPGA/Arduino controller controls the actions of the motor driver with inputs from the rotation encoder. The rotation encoder is fitted to the motor shaft to record the number of rotations completed by the motor. The motor drivers control the direction of rotation of the motors. The overall architecture of the design is shown in the block diagram in Fig. 1. The design can be fitted to any small scale indoor navigation device. Herein we use a normal wheelchair, replacing the back wheels with 320W, 24V motor driven wheels with a speed of 4600RPM. These motors are controlled by Pololu 24V, 23A motor drivers. The basic design is common for both Arduino and FPGA alternatives. While Arduino supports both Hercules and Pololu motor drivers, FPGA normally supports only Pololu motor driver, since output voltage range provided by FPGA is not sufficient for proper operation of Hercules motor driver for a long time. To use Hercules motor driver with FPGA we need some additional circuitry like a voltage level shifter to pull up the FPGA output voltages. We use a DC-DC level converter for this purpose. Still it is advisable to use Pololu motor controllers for FPGA as it avoids the use of any additional circuitry and also reduces the risk of overshoot or damage of the controller.

A. FPGA Board

In this design one of the controllers that we use is a Digilent Basys 2 Spartan 3E shown in Fig. 2. The FPGA has 18-bit multipliers, 72 Kbits of dual port block RAM and operates at frequencies above 500MHz. External supplies in the range of 3.5V to 5.5V is possible. User I/Os can be provided with the four 6-pin headers, 8 LEDs, 4 pushbuttons and 8 slide switches. Inbuilt clocks provided are of 50MHz and 100MHz The FPGA receives signals from the Encoder, and calculates the required distance to be travelled. It also controls the operation of all other components accordingly and also is open to future enhancements. Coding is done in Verilog HDL improving the user accessibility of the design.

B. Arduino UNO Board

The main component of the UNO is the ATmega328P microcontroller. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. The ATmega328 is an 8-bit microcontroller developed by Atmel which follows the RISC architecture and supports serial communication protocols like USART, SPI and TWI (otherwise known as I2C). The board is shown in Fig. 3.

C. Rotation Encoder

One of the main components of the design is a Rotation encoder which is used to estimate the distance travelled from the number of pulses obtained. The IC used for the same is an MOC7811 detector IC. The MOC7811 is divided into two sections, an LED section and a detector section. When properly biased the emitter emits continuous radiations in the direction of the detector. To obtain the number of pulses we fix a patterned metal disc on the motor shaft. The leaves of this disc are placed in such a way that it rotates in between the emitter and detector sections of the encoder IC as shown in Fig. 4. The circuit arrangement of MOC7811 is shown in Fig. 5.

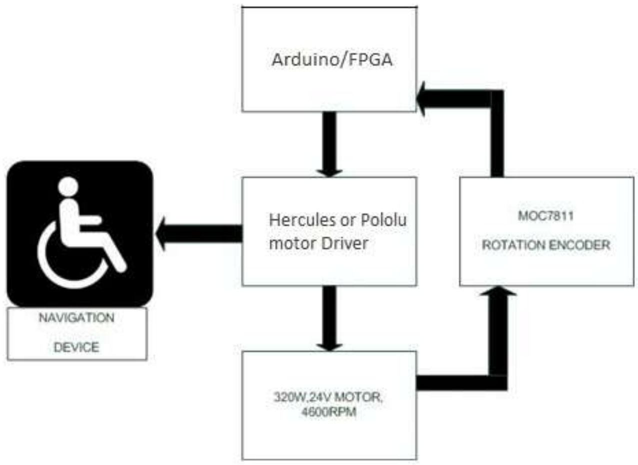

Fig. 1. Block diagram

Fig. 2. Digilent Basys 2 FPGA board

Fig. 3. ARDUINO UNO board

Fig. 4. Rotation Encoder Setup

This output from the emitter is then a pattern of alternate light and dark which is captured at the detector section. This pattern is converted into a pulse waveform with the light and dark sections represented by ON and OFF of the pulse. Hence the output of the rotation encoder is a pulse waveform. Now from the number of pulses we can calibrate distance using the following formula.

$$Distance = \frac{Wheel\ Circumference \times Counts}{Counts\ per\ Revolution} \quad (1)$$

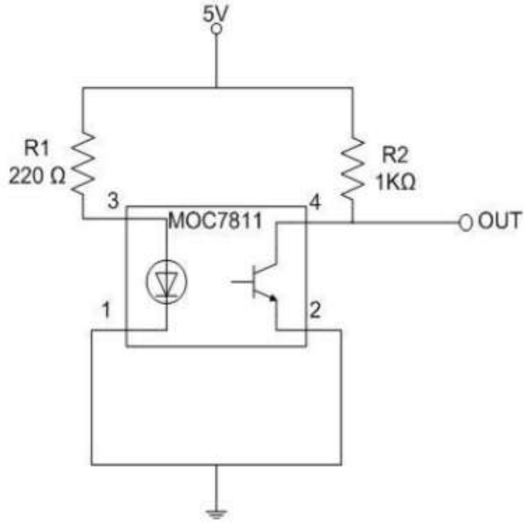

Fig. 5. Circuit Diagram of Rotation Encoder

D. Motor Drivers

To drive the wheelchair motors we use two Pololu 24V, 23A motor drivers. These MOSFET H-bridge motor drivers enable bidirectional control of high-power DC motors. The H-bridge is made up of N-channel MOSFETs which determine the performance of the drivers. The maximum frequency allowable for PWM is 40KHz. The port connections between the FPGA/Arduino and the motor driver and also between the driver and motor are shown in Fig. 5.

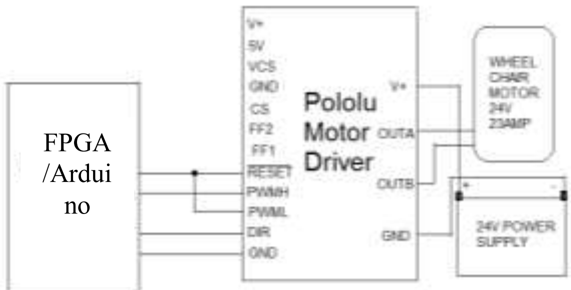

Fig. 6. Port Connections between controller, Motor Driver and Motor

V. EXPERIMENT RESULTS

Test runs were conducted using a Rotation Encoder with four leaves with diameter 6.5cm. The motors used were 24V, 320W wheelchair motors, which were driven by 24V, 23A Pololu motor drivers. The FPGA used is a Digilent Spartan 3E, Basys 2 board and Arduino used is UNO. The experimental setup is shown in Fig. 6.

Table I shows the difference between actual distance travelled and Expected distance Travelled when we tested the device to move in a straight line to different distances ranging between 1 to 4m. As shown in the table the error percentage for FPGA is lesser than Arduino.

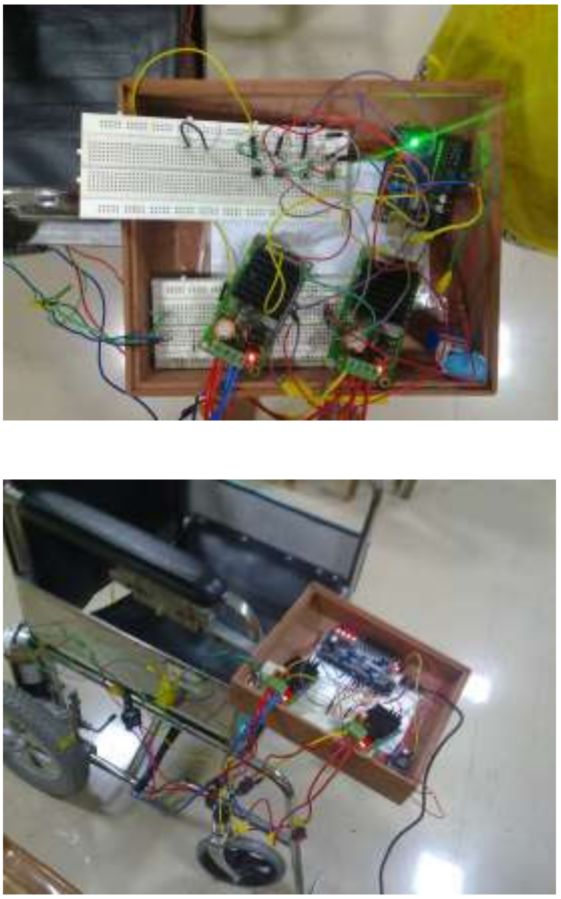

Fig. 7. Experimental Setup of the design in Arduino and FPGA

TABLE I
DISTANCE ESTIMATION WITH ROTATION ENCODER FOR FPGA AND ARDUINO UNO

No of Tries	Expected Distance in cms	Obtained Distance in cms for Arduino	Obtained Distance in cms for FPGA	Error percent age for Arduino	Error percent age for FPGA
1	100	95	97	5	3
2	100	101	101	1	1
3	150	148	148	1.3	1.3
4	150	146	145	2.6	3.3
5	200	201	200	0.5	0
6	200	199	200	0.5	0
7	250	248	250	0.8	0
8	250	249	248	0.4	0.8
9	300	301	300	0.3	0
10	300	303	301	1	0.3
11	350	351	351	0.28	0.28
12	350	348	349	0.57	0.28
13	400	401	404	0.25	1
14	400	400	399	0	0.25

The device was tested in a plot with a grid size 0.6m in Arduino and FPGA as shown in Fig. 8a and Fig. 8b respectively. The whole plot was divided into three different sections and a fixed path was set to travel from one section to another. Fig. 8a and Fig. 8b shows the Expected path and the Actual path taken by the device for the best case among several trials conducted.

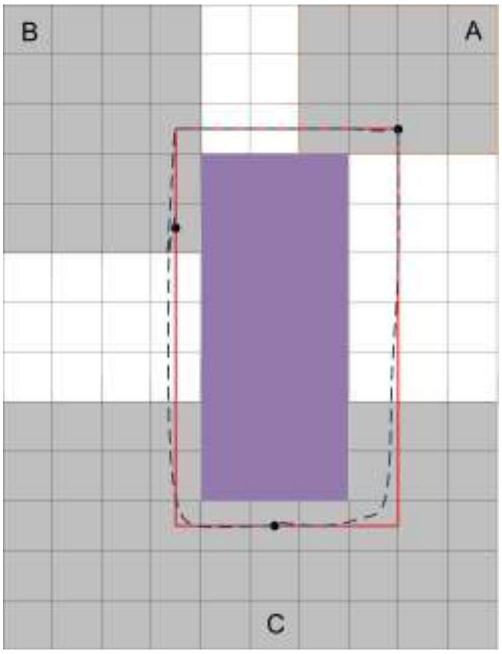

Fig. 8a. Plots showing expected path and obtained path for ARDUINO

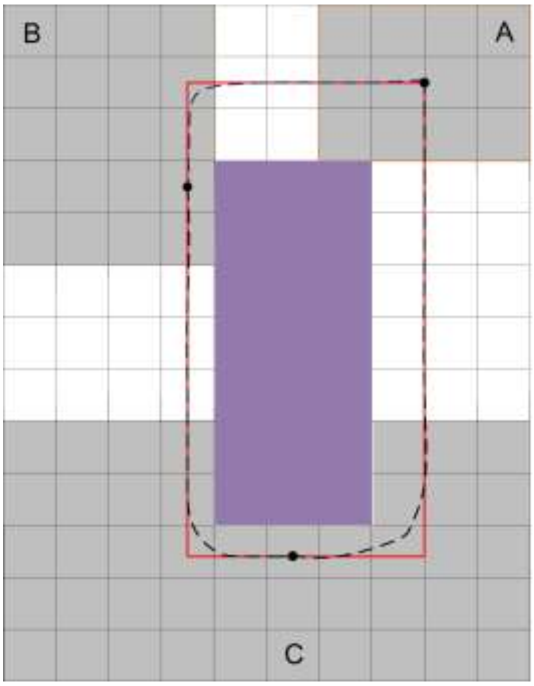

Fig. 8b. Plots showing expected path and obtained path for FPGA

VI. OBSERVATION

Both systems as mentioned above were tested on the same plots for different distances and shapes and the result was recorded. In case of the accuracy of rotation encoder, for Arduino the recorded average error percentage was 1.03. While for FPGA it was 0.82. Even though both this error percentages are low we found that the FPGA performed slightly better. In case of complexity however, programming the Arduino was much easier as compared to the FPGA. Moreover Arduino is much more flexible and reprogrammable but then is not as durable as the FPGA and it is not advisable to use Arduino for long time purposes. Also the actual hardware worked a bit more accurately in case of FPGA.

VII. CONCLUSION

An FPGA and or Arduino based Navigation platform for fixed path navigation has been designed. The design was tested on a normal wheelchair with a rotation encoder fixed on one of the motors. We tested the device for the same paths in case of both Arduino and FPGA. From analysis FPGA provided more accuracy but complexity is more as compared to Arduino.

ACKNOWLEDGMENT

Grace to almighty for granting as the opportunity to take part in this endeavor. The authors wish to thank Amrita Vishwa Vidyapeetham University, Kollam Campus, Kerala for providing us support and funding to carry out this project. We also wish to thank the Humanitarian Technology Lab of ECE Dept., for aiding us in this endeavor.

REFERENCES

[1] P. Dinesh, P. Kishore, Vivekanandan B. and S. Moorthi 'RFID based Navigation System for Unmanned Material Handling Vehicles using FPGA' 2013 International Conference on Intelligent Systems and Signal Processing (ISSP)

[2] Marek Popáawski, and Michal Bialko 'Implementation of Parallel Fuzzy Logic Controller in FPGA Circuit for Guiding Electric Wheelchair' .Krakow, Poland, May 25-27, 2008

[3] Chinnaiah M C, T.Sathya Savitri & P Rajesh Kumar, 'A Novel Approach in Navigation of FPGA Robots in Robust Indoor environment' , ARIS 2015, TAIWAN

[4] Yoshihisa Adachi, Yoshinori Kuno, Nobutaka Shimada, and Yoshiaki Shirai 'Intelligent Wheelchair Using Visual Information on Human Faces' proceedings of the 1998 *lEEE*lRSJ Intl. Conference on Intelligent Robots and Systems Victoria, B.C., Canada October 1998

[5] J. Kathirvelan, R. Anilkumar, Zachariah C.Alex, and A.Fazul 'Development of Low Cost Automatic Wheelchair Controlled by Oral Commands using Standalone Controlling System', 2012 *IEEE International Conference on Computational Intelligence and Computing Research*