The Development and Application of a Smart Ultrasonic Sensor IP-Core in a Mobile Autonomous Robot

Liberty Mutauranwa, Thanks Marisa, Peter Manyere and Emmanuel Rashayi
libertym01@gmail.com,thanxm@gmail.com

University of Zimbabwe, Department of Electrical Engineering, Harare, Zimbabwe

Abstract. Autonomous robots are being proposed as the future drives of industry, transport and defense sectors. This paper describes the design and development of a soft IP core for an ultrasonic sensor on an Autonomous Mobile Robot. The IP core is implemented on the Xilinx Spartan 3 FPGA as part of a System-on-Chip control system for the robot. The ultrasonic sensor IP core is utilized in a Drive-by-Wire application in which a line of robots follow a lead robot along a marked highway. As the robots motion is based exclusively on sensor data, IP cores are developed for the robot smart sensors and actuators. In this paper we focus on the development of the ultrasonic ranging sensor IP core, its properties and how it interfaces with the robot hardware. Our target applications are in industry and agriculture.

Keywords: Ultrasonic Sensor; Soft IP Core; FPGA; I2C; System-on-Chip; autonomous mobile robot

1 Introduction

The project outlined in this paper is part of a Drive-by-Wire application for a group of autonomous robots. The robots are meant to navigate their environment in a synchronous motion. This would allow the robots to be coordinated in carrying out a required task. An important part of this arrangement are the sensors that enable the robots to move in a synchronous motion. One of the sensors employed is the ultrasonic sensors at the head of each robot. The design and implementation of a soft Intellectual Property Core (IP Core) for the ultrasonic ranging sensors in a System-on-Chip (SoC) control system is described in this paper. This IP core allows the robots to detect each other and also measure distances between each other. Drive by wire, technology is the use of electrical or electro-mechanical systems for performing functions traditionally achieved by mechanical linkages [1]. These systems are already in use in vehicles mainly to enhance safety, by providing computer controlled intervention of vehicle controls [2]. Our work implements and evaluates ultrasonic ranging sensors in grouped robot coordination for the Gecko3 robots [3]. In this application the Gecko3 robots were developed into autonomous mobile robots with their behaviour being highly dependent on the developed soft IP-cores for their sensors.

ACRID 2017, June 20-21, Victoria Falls, Zimbabwe

DOI 10.4108/eai.20-6-2017.2275890

2 **The Drive-By-Wire Project**

The overall goal of the drive-by-wire project was to develop software and hardware algorithms to allow a group of Gecko3 robots to drive in coordinated motion. For example implementing "train" motion, where a head robot runs along a marked "high-way" on a floor, being followed by a lined group of robots separated by a fixed distance between them. The coordination was to function with no physical connections between adjacent robots but was to be based solely on sensor data obtained by the robots. The sensors to be used in the project were the optical reflective sensors beneath the robots and the ultrasonic range-finder sensors at the front of the robots. The optical reflective sensors for the lead-robot would detect the directions marked on the floor. Ultrasonic range-finder sensors for slave-robots would in turn detect the robot in front of them and ensure that a pre-set distance is maintained as the slave-robots coordinate their motion with the lead-robot.

2.1 The Gecko3 Robot

The Gecko3 is a platform for concurrent software and hardware co-design in realtime information processing and SoC applications [3] . The Gecko3 comprises of different credit-card size hardware modules that can all be stacked together by a common backbone connector to make up a miniature robot. In the project described here the Gecko3Main and the Gecko3Robot modules were applied. The Gecko3 is a platform robot is shown in Fig. 1.

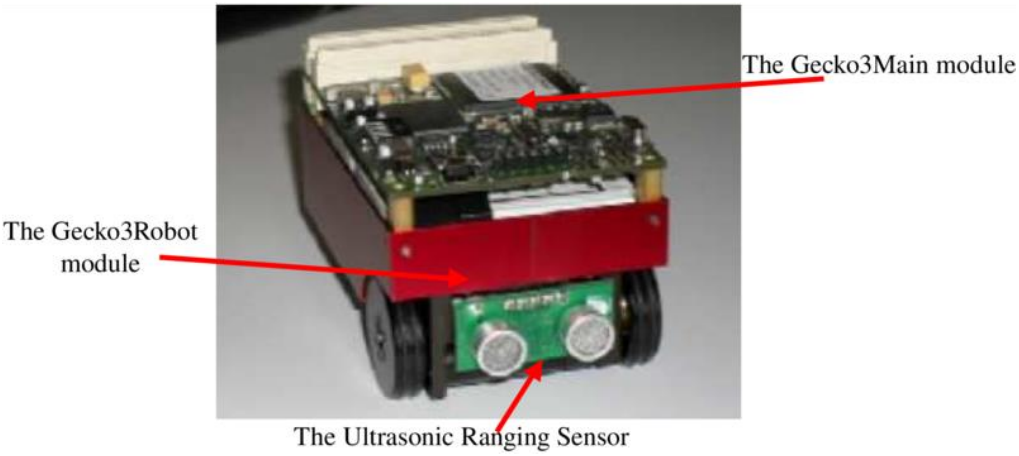


Fig 1. The Gecko3 Robot and the on-board Ultrasonic Sensor

The Gecko3Main is the main controller board comprising a Xilinx Spartan3 Field Programmable Gate Array (FPGA), Input/ Output interfaces and memory chips. The Gecko3Robot is the mechanical robot comprising of a power module, an onboard ultrasonic ranging sensor, reflective optical sensors and the mechanical vehicle. This paper focuses on the work done for the ultrasonic range-finder Sensors for the Drive-by-Wire project.

2.2 The Robot SoC

The robot system was implemented as a SoC on the on-board FPGA. Soft IP Cores were designed, implemented and integrated in FPGA hardware. The Microblaze RISC microprocessor was utilized as the processor core. The Soft IP Cores for the sensor and actuator hardware were developed in VHDL. The IP Cores interface with the Microblaze processor through the OPB bus as illustrated in Figure 2.

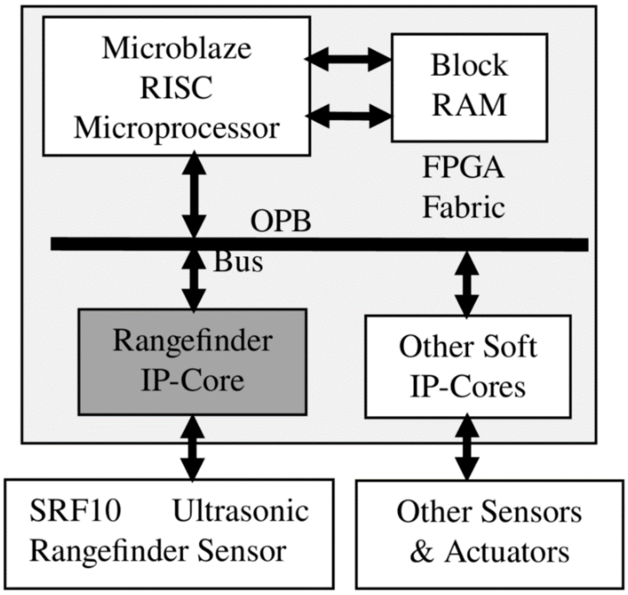


Fig 2. The Gecko3 SoC solution for the Drive-by-Wire Project

The design and development of an IP Core for the ultrasonic ranging sensor is the focus of this paper. This IP Core is termed the ``Range-finder IP Core".

3 The Ultrasonic Ranging Sensor Hardware

The SRF10 Ultrasonic range-finder is the sensor hardware that was utilized for ultrasonic ranging between the robots. The SRF10 Ultrasonic range-finder consists of an ultrasonic transmitter and receiver. The transmitter transmits ultrasonic pulses at a frequency of 40 kHz while the receiver “picks up” reflections of the transmitted pulses off obstacles in its proximity. Communication with the sensor from an external controller utilises the I2C protocol [4], [5].

4 The Range-finder IP Core

The range-finder IP Core was developed for controlling the SRF10 Ultrasonic sensor and reading measurements from it. The output of the IP core is an unsigned decimal value specifying the estimated distance between the sensor and its nearest object.

4.1 IP Core Design Concept

To achieve the required operation, the range-finder module is subdivided into three modules, the I2C Master, the range-finder controller and the filter. The I2C Master is an interfacing module to facilitate communication between the FPGA and the SRF10 ultrasonic range-finder.

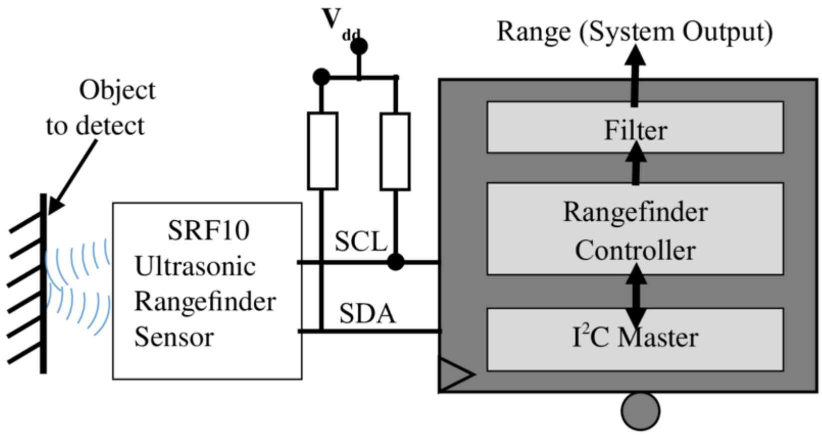


Fig 3. The range-finder IP core module design

The range-finder controller is a controller module for initialization and configuration of the SRF10 slave in addition to reading the range values from it while the filter module filters the range output of any noise. The IP core module design is illustrated in the gray section of Fig. 3.

4.2 The Range-finder Controller Design

For the SRF10 Ultrasonic sensor to function, initialization and configuration commands have to be sent to it from the I2C Master. Range values also have to be read from the sensor. The range-finder controller was designed to achieve this. The module specifies and controls the data written to the sensor via the I2C Master module. The module also controls reading of data from the SRF10. The design goals for the module were:

- To configure the sensor's Range and Gain settings.
- To initiate and control the ranging sessions.
- To interpret the value read back from the sensor for use in the autonomous robot control algorithms.

The Range-finder Controller Algorithm: A ranging session is initiated by writing a Ranging Mode command to the command register of the SRF10 Ultrasonic sensor. This command specifies whether the result of ranging is in inches, centimetres or microseconds. The range and gain settings for the sensor are then set. The Range setting specifies the maximum range that can be measured. The gain setting sets the maximum gain of the sensor analogue stages. During a ranging session the analogue gain starts off at a minimum value of 40 and increases every 96μs up to the maximum gain set by the gain setting [4].

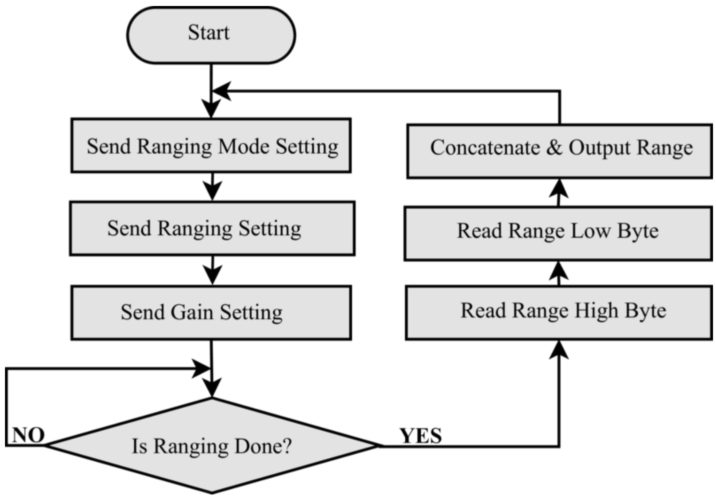


Fig. 4. The rangefinder controller algorithm flow chart

The range-finder controller algorithm illustrated in Fig. 4 was developed to achieve these specifications. The algorithm was implemented in hardware following the FSM-D design approach [6].

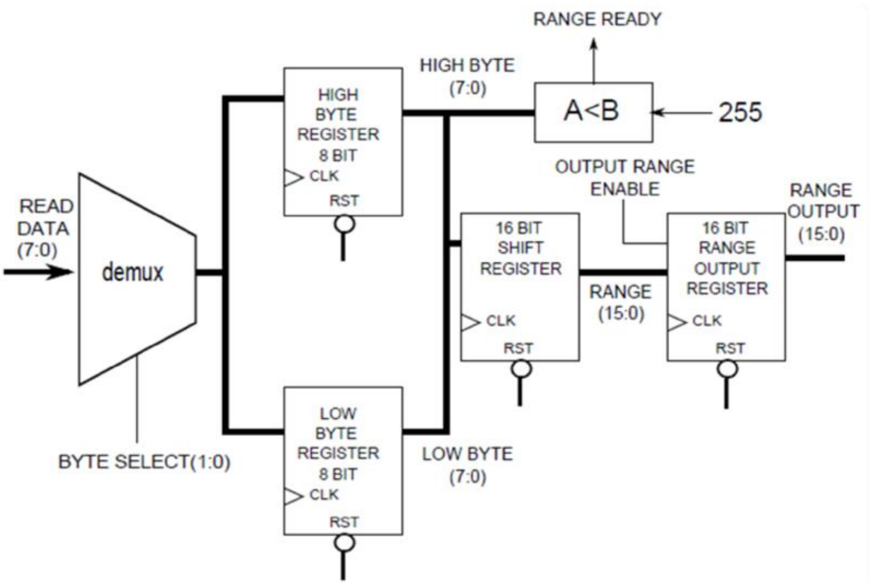


Fig. 5. The range-finder controller datapath

The Rangefinder Controller Datapath: When a range value is read from the sensor, the range-finder controller breaks the value into its lower byte and the upper byte. The module then compares the value of the upper byte with 255 so as to determine whether the ranging is finished. If the ranging is complete an enable input from the FSM results in the presentation of the range value on the Range Output (15:0). The range-finder controller algorithm's data path is illustrated in Fig. 4

The Range-finder Controller FSM: The FSM controls the operation of the datapath. The FSM design is shown in Fig. 6.

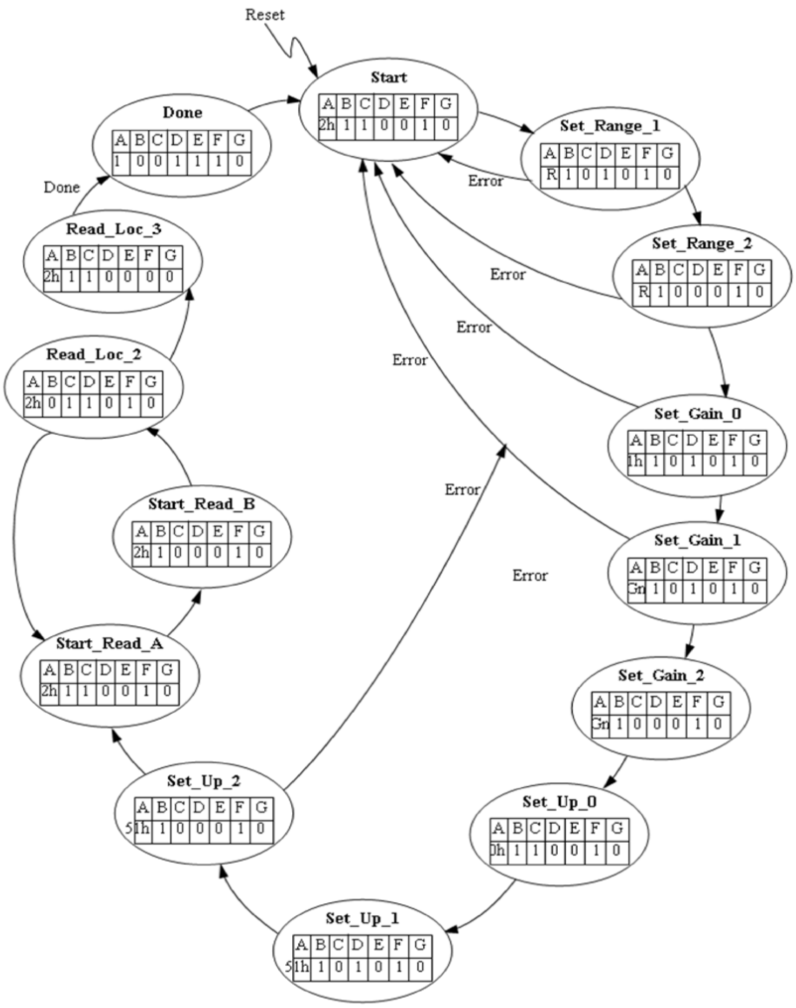


Fig. 6. The range-finder controller integrated FSMD

Table 1 gives the outputs of the FSM state.

Table 1. The range-finder controller FSM output codes

FSM Output	Code
Data_Value	A
Write	B
Start	C
Next_Byte	D
Output_Range	E
Select_Byte	F
Data_Valid	G

The Range-finder Controller Datapath & FSM Integration: The range-finder controller data path and FSM were integrated. The interface definition for the integrated range-finder controller module is illustrated in Fig. 7.

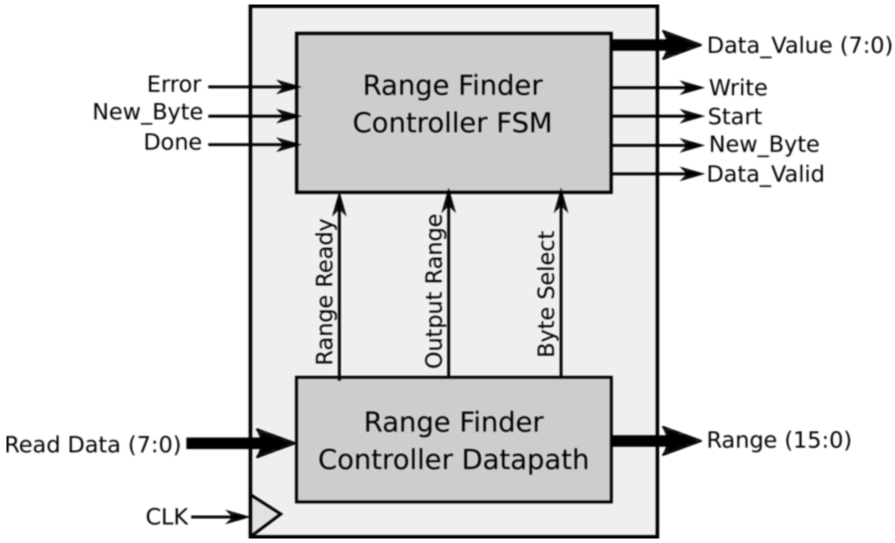


Fig. 7. The complete range-finder controller interface definition

4.3 The Range Filtering Module

The range outputs of the range-finder-controller module were analysed over a wide range of measurements. Although the outputs were accurate there was still some noise influence. Simulations in Matlab indicated median filtering to be the most appropriate scheme for ridding the range output of noise. The Range Filtering Module was therefore developed in the form of a median filter of order 7 [7]. The implemented median filter was based on the Bubble-Sort Architecture [8], [9].

5 Results

Experiments were conducted on the range-finder IP core. Tests were conducted to check the accuracy and precision of the ultrasonic smart sensor IP Core and the effect of the gain and range settings on it.

5.1 Measurements Accuracy Experiments

A range setting of 11 metres and a gain setting of 140 were set on the range-finder IP core. Ranging measurements were performed for an obstacle progressively moved between a range

of 10cm and 80cm from the ultrasonic sensor. The range measurements accuracy is shown in Fig. 8, our algorithm achieved an average accuracy of
98.3%.

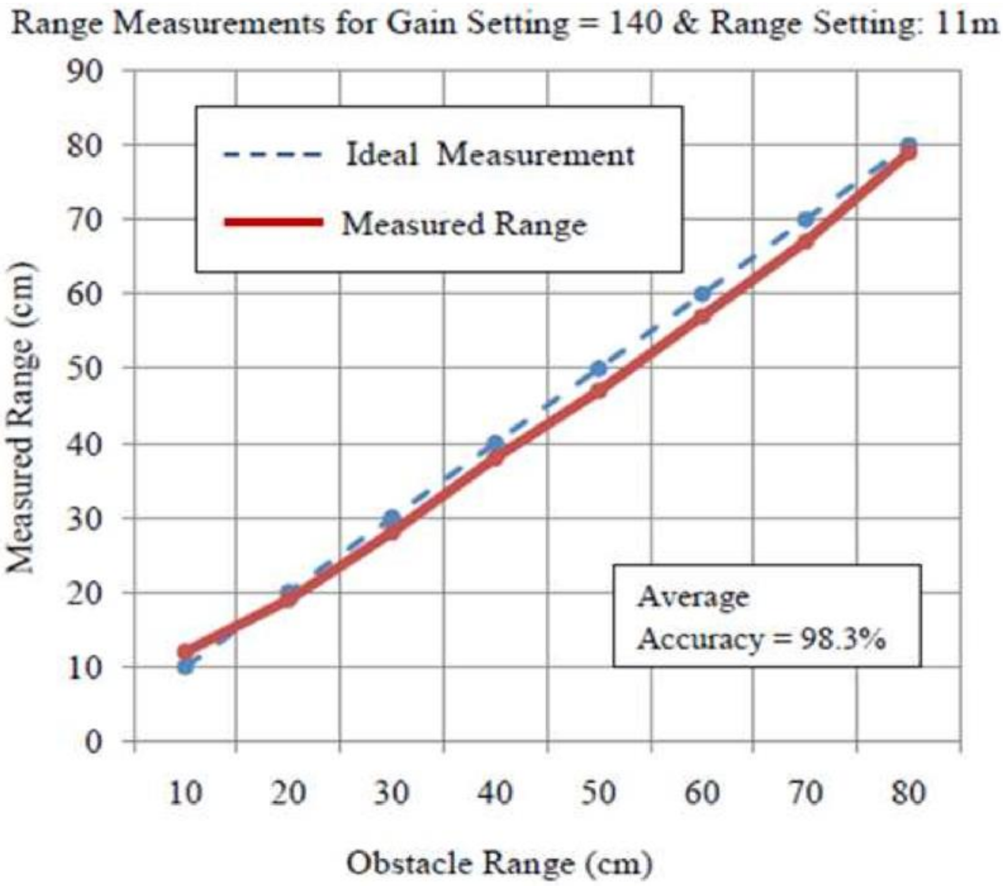


Fig. 8. Ultrasonic range-finder accuracy comparison results.

5.2 Gain/Range Settings Experiments

The range setting was kept at a constant of 146cm whilst the gain setting was increased from 40 to 700. The obstacle was progressively moved from a range of 0cm to 160cm. The experimental results for different gain and range setting are shown in Fig. 9.

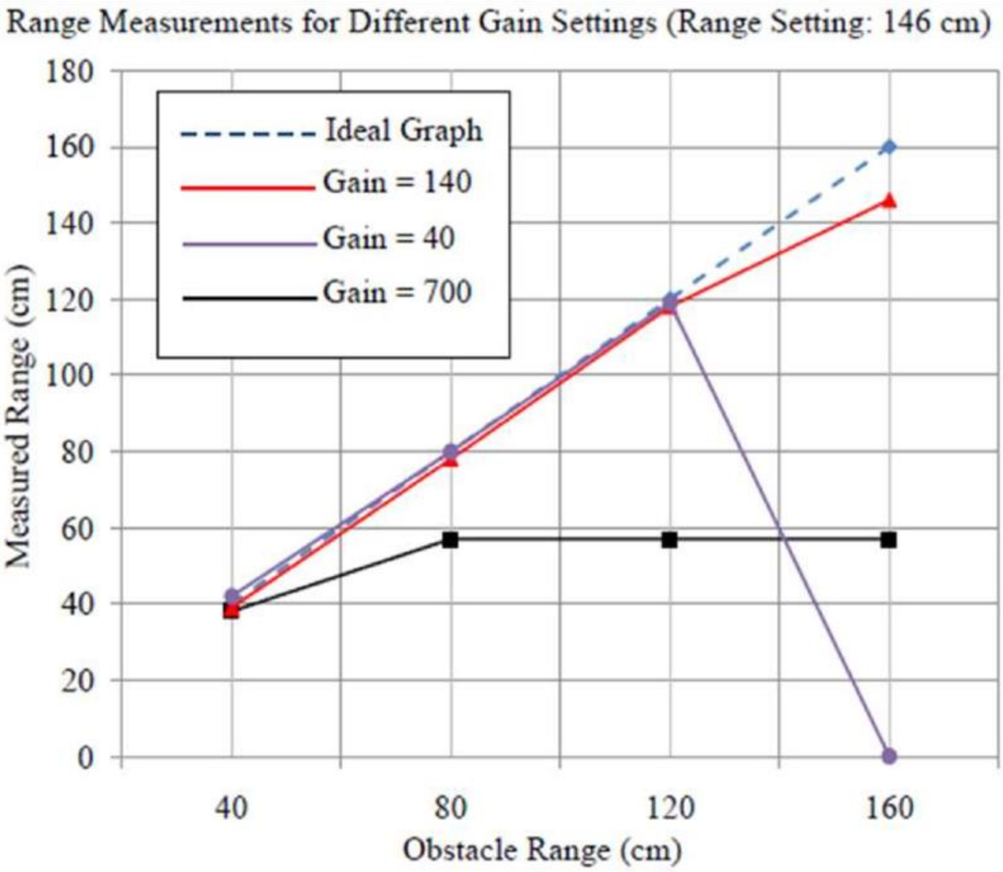


Fig. 9. The effects of range/ gain settings on range measurements.

We observed that the gain of 140 is optimal. A gain that is too low results in the range-finder only being able to detect only close-by obstacles, while performing poorly for longer detection distances. A gain that is too high e.g. 700 in the Fig. 9 results in high amplification of weak echoes from surrounding objects, thus allowing more noise (unwanted signals) into the measurements.

6 Conclusions

The range-finder IP Core for the SRF10 Ultrasonic range-finder was developed for its applications as a smart sensor on the Gecko3 robot for the drive-by-wire application [10]. The range-finder IP Core can be used together with the ultrasonic sensor to detect objects in its proximity. The IP Core designed facilitates a sensor accuracy of above 98%. However it is clear that the accuracy is dependent upon the gain and range settings. The smart sensor accurately measures ranges up to a maximum of 146cm. In Fig. 9, with a gain of 40 the range-finder can only detect the obstacle up to a range of 120cm and beyond 120cm the echoes were too weak to give an accurate value. The range-finder IP core developed enables the application of the SRF10 Ultrasonic range-finder sensor on any other FPGA based system.

References

[1] M.E. Elbuluk, M.D. Kankam: "Motor drive technologies for the power-by-wire (PBW) program: options, trends and radeoffs", IEEE National Aerospace and Electronics Conference, Dayton, v.1 p.511 (1995)

[2] N. Stanton, P. Marsden: "From fly-by-wire to drive-by-wire: safety implications of automation in vehicles", Safety Science, v.24 n.1 p.35, (1996)

[3] M. Jacomet. (2010, April 24). Gecko3 Modules. [Online]. Available: http://labs.ti.bfh.ch/gecko/wiki/modules/start

[4] SRF10 Ultrasonic Rangefinder, Devantech Ltd, Norfolk, England, (2003).

[5] UM10204 I2C-bus specification and user manual, NXP Semiconductors, Eindhoven, The Netherlands, (2014).

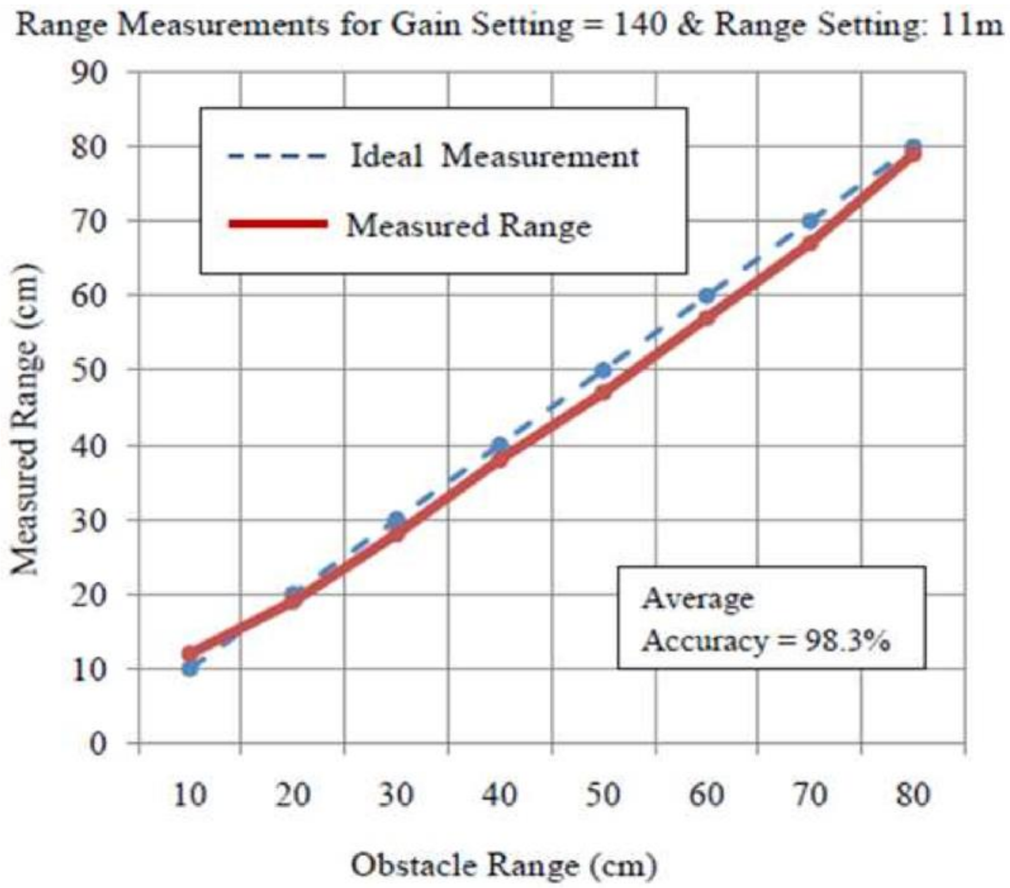
[6] M. Jacomet: VLSI System Design: FSM-D Architecture Model. Biel, Switzerland: Berner Fachhochschule, January (2003).

[7] D. S. Richards: VLSI Median Filters, IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 38, pp.145 - 153 , (1990).

[8] V.G.Moshnyaga and K.Hashimoto: An Efficient Implementation of 1-D Median Filter, Proceedings of the 52-th IEEE Midwest Symposium on Circuits and Systems (MWS2009), Cancun, Mexico, August 2-5, pp.451-454, (2009).

[9] L. Mutauranwa and M. Nleya: An efficient median filter in a robot sensor soft ipcore, AFRICON, Pointe-Aux-Piments, 2013, pp. 1-5, (2013)

[10] L. Mutauranwa : Smart Sensor Hardware for Autonomous Mobile Robot Perception M.Phil thesis, Dept. Electron. Eng., National University of Science and Technology, Bulawayo, Zimbabwe, (


Range Measurements for Gain Setting = 140 & Range Setting: 11m
Ideal Measurement
Measured Range
Average
Accuracy = 98.3%
Measured Range (cm)
Obstacle Range (cm)
0
10
20
30
40
50
60
70
80
90
10
20
30
40
50
60
70
80

2014).