Inertial navigation module

Sebastian Sabou, Claudiu Lung , Ioan Orha

North University Center from Baia Mare/Electronic and Computer Engineering, Baia Mare, Romania,

sebastian.sabou@ubm.ro

Abstract—**This paper presents low cost Inertial Navigation Module, using MEMS sensor and an simple onboard microcontroller data processing (Kalman filters, basic interpretation for data). GPS navigation solutions are investigated, but they encounter big issues in indoor environment. The basic problem is the weak signal power received from satellite in indoor location and therefore the increased error for position. Using another technique and sensor add more precision. MEMS based techniques are promising ones. Using an accelerometer and a digital compass allows the determination of additional parameters that can lead to determine the position or improve existing location. The components used are the usual, available at any specialty store and precision obtained using a minimal data processing software is encouraging. Research direction to follow is to process the data obtained for growth and greater accuracy.**

I. INTRODUCTION

Basic GPS principles: Satellites Triangulation, as show in figure 1, and from there result some of the advantages and disadvantages of this method to determine position. GPS devices are limited by having clear access to the satellites that provide the tracking. In locations with tall buildings or sparse coverage, reception can be poor.

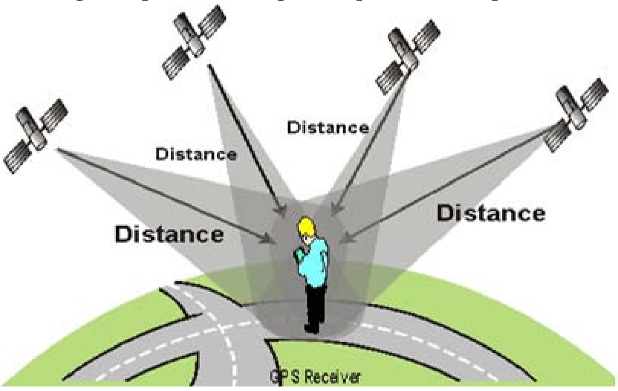

Figure 1. GPS principle

A. *Inertial measurement unit*

Another approach to achieve the goal of determine the position is the use of inertial measurement unit. The classical inertial measurement units are replaced with low-cost and low-dimensions ones and with comparable performance but miniaturizations come with some disadvantage like poor performance in terms of noise density, bias and scale factor stabilities. These are solved with data filtering and extract useful data from sensor.

II. SENSORS

For measure acceleration, ADIS 16209 sensor is used (figure 2). Acceleration is measured just for horizontal axis (X and Y) with 14 bits resolution, data with twos complement. Communications is implemented via SPI bus (figure 2) at a maximum 2.5MHz frequency of clock. From all 14 bits available just 8 bits (MSB) is used for simplify all the calculation made with microcontroller.

The sensor have the option to perform a small filter on circuit, filter with configurable parameters which determines the size of the moving average filter in eight power-of-two step sizes (that is, 2M = 1, 2, 4, 16, 32, 64, 128, and 256). In this case the size for moving average filter is 64. This number was chosen after testing all other values and this has better results. In fact the size depend for each type of movement and speed, in some case is made an auto configurable option to improve data output and minimize errors.

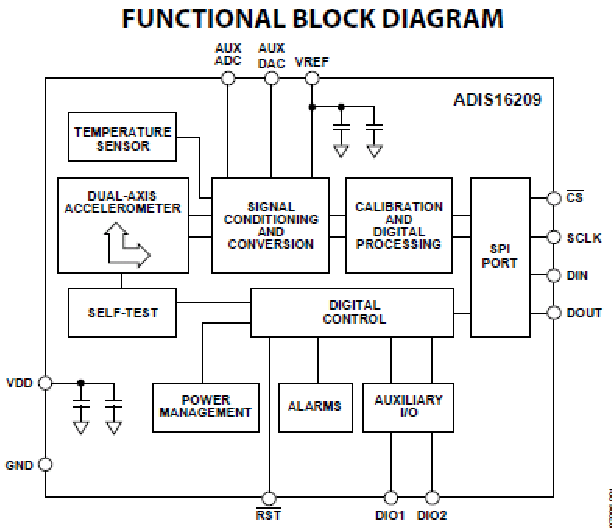

Figure 2: Functional block diagram ADIS 16209

For measure the angular parameter an ADIS 16250 gyroscopic sensor is used. Also data is available in 14 bits format and just 8 bits its used (MSB bits). Also communications with gyroscopic sensor si implemented via SPI bus (figure 3).

ISSN 1844 – 9689

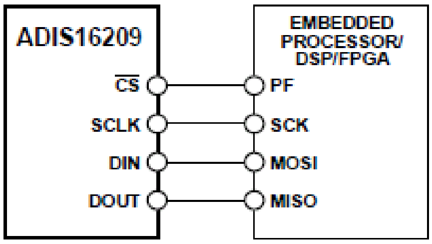

Figure 3: SPI communications with sensors

In figure 4 is present functional block diagram for ADIS 16250 gyroscopic sensor.

.

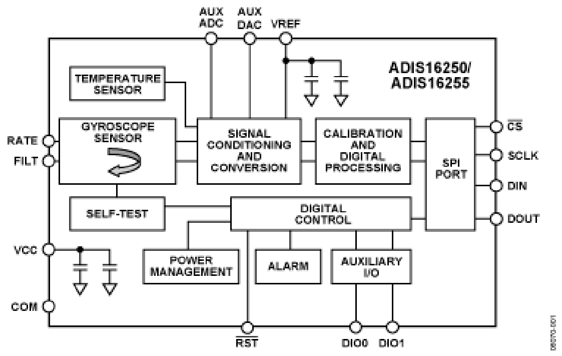

Figure 4. Gyroscop sensor functional block diagram

III. Microcontroller

For data processing a fast microcontroller is used, in 8051 architecture, from Maxim Integrated. The microcontroller is DS89C450 with features like:

- High-Speed 8051 Architecture
- 64kB Flash Memory - In-System Programmable Through Serial Port
- 1kB SRAM for MOVX
- 33 MIPS performance from a maximum 33MHz clock rate
- offer the highest performance available in 8051-compatible microcontrollers.

Functional diagram of the microcontroller is show in Figure 5.

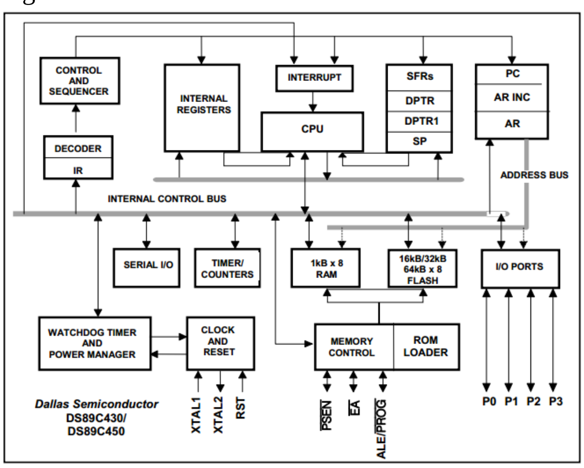

Figure 5. Functional diagram of DS89C450

IV. Neural network

The typical model of a neuron usually used in neural network is related to variations of the McCulloch-Pits (1943) neuron witch can be described by equations (1) and (2). These specify a linear weighted sum of the inputs, followed by a non-linear activation function.

$$u=\sum_{j=1}^{n} w_j * x_j \qquad (1)$$

$$y=\frac{1}{1+e^{-au}} \qquad (2)$$

where u is the activation of the neuron, w are the weights, x are inputs, a controls the shape of the output sigmoid, and y is the output.

This type of equation are relatively hard to implement in microcontroller software and the processing speed obtained are very low. The solution is to find another architecture of neural network who can implement and run at a better speed.

The chosen solution is a weightless neural network, in this case a RAM-based neural network. One of the big advantage over other neural network architecture is in fact to eliminate the necessity of hardware multiply unit, with in microcontroller design is very time consuming. In this case, the hardware multiplier is replaced with an hardware sum device and a matrix type data. The RAM-based neural network investigated here, are based on artificial neurons which often have binary inputs and outputs, and no adjustable weights between nodes (weightless). Neuron logic functions are stored in look-up tables that can be implemented using commonly available Random Access Memories (RAMs).

For training the neural network a MathLab program is used, just to cut down the computational resources and the memory necessary and to speed up the process of learning. In first step data sample was recorded from sensors for each movement, with several movement repeat for each directions and at different speed. In second step, data sample was used for training the neural network until the output was almost errorless. The matrix weights was transferred in microcontroller memory and the module was tested in the same move and directions for verify the learning procedure.

V. Implementation

The microcontroller module is shown in figure 6.

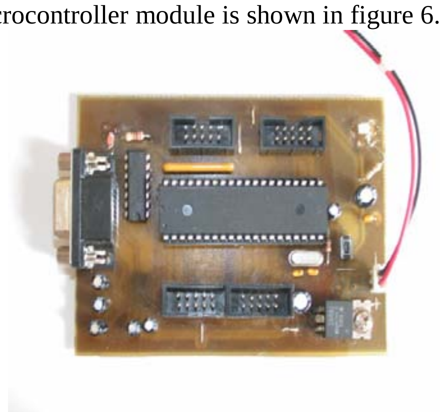

Figure 6. Microcontroller board

Functional diagram of system is presented in figure 7

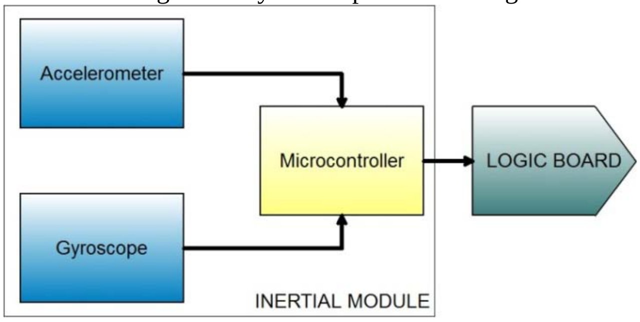

Figure 7. Diagram of the inertial measurement unit

Program flow is show in figure 8, two mod are available for module: one just as smart sensor and one as autonomous navigation, where data from sensors are interpreted and transmit to the decisional block or module.

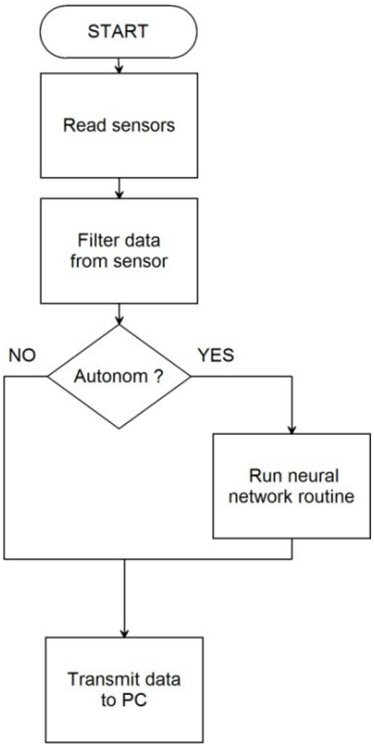

Figure 8. Program data flow

Data filter is implemented partial in sensors with moving average filter with variable size for window an another filter is with Kalman filter algorithm.

Because implementing Kalman filter formulas in C on 8bit microcontroller it's a lot complicated for matrix and vector math. After some research and tests a one dimensional filter was chose and that's simplify a lot the code to implement this.

The final set of formulas are in equations (3) – (7).

$$x = x \quad (3)$$

$$p = p + q \quad (4)$$

$$k = \frac{p}{(p+r)} \quad (5)$$

$$x = x + k(datameasure - x) \quad (6)$$

$$p = (1 - k)p \quad (7)$$

The (3) and (4) equations represent the prediction of the Kalman filter. The next three equation (5), (6), (7) calculate the update for measurement data. The variable *x* is for filtered value, *q* process noise, *r* sensor noise, *p* estimated error and *k* Kalman gain.

The neural network is partial implemented but because of the "main" of this type of network the generalization problem the results are, at this moment, very poor. This type of neural network do not generalize, just find the correct solution on the sample from output and just if these sample was learned before.

A one way to solve this problem is to introduce an element than can generalize, like a sum cells device and divide with number of cells.

In figure 9 is present the elementar unit os this weightless neural network.

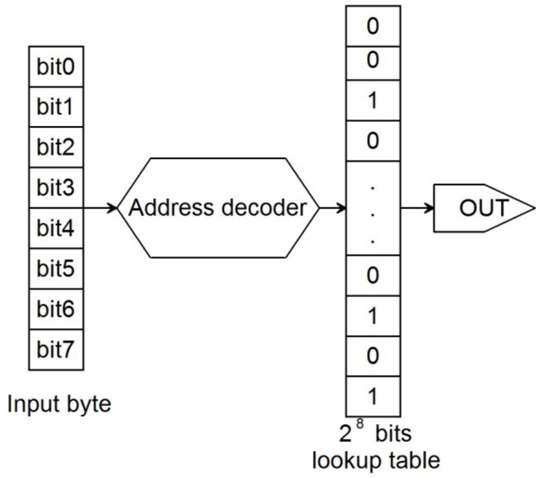

Figure 9. Neural unit

The next component from neural network is the discriminator, which is a bit addressable RAM discriminator, a layer of K n-input RAMs each one storing 2^n one bit word. In order to solve the general problem of RAM-nodes, which is having no generalization, several of these RAM-nodes can be combined to a so called discriminator as in figure 10. The most important feature of a discriminator is that it splits the input pattern into K n-sized tuples, each are used as the input for a RAM-node. In this way the input pattern of the discriminator is distributed over the memory of several RAM-nodes, instead of being stored in only one location.

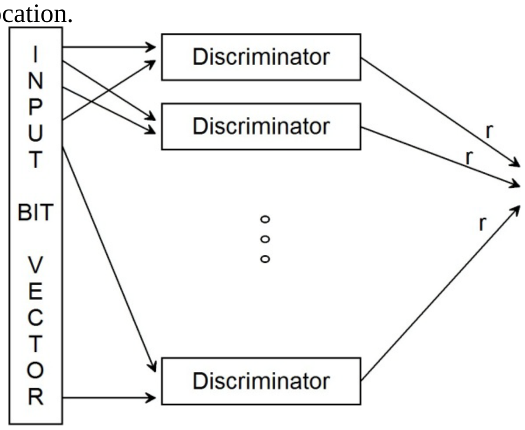

Figure 10. Discriminator

I. RESULTS

The entire software for microcontroller is written in C language, using Keil C compiler. RAM memory is fully used for neural network lookup table, Kalman filter and other variable. Flash program memory is used for about 40% from 64KB totally available.

Neural network is usable but the “classic” problem of no generalization is the cause of some errors that in other neural architecture network isn’t present, but is very fast in compare with other neural network because the “binary weight” and without the necessity of hardware multiplier. It is implemented in a classic 8051 microcontroller architecture with just a 1kB of RAM (and 256 bytes internal).

The speed calculate from sensor data are represent in figure 11, where four test at different speeds are represent on one graph.

Figure 11. Estimated speed

In this stage the Inertial Measurement Unit work as “dead reckoning” and from one starting point calculate the actual localizations – transmit to PC de current position based on inertial measurement

REFERENCES

[1] Aiden Morrison, Valérie Renaudin, Jared B. Bancroft and Gérard Lachapelle, Design and Testing of a Multi-Sensor Pedestrian Location and Navigation Platform, *Sensors* 2012, *12*, 3720-3738; doi:10.3390/s120303720

[2] L. Marchese, “Digital neural network based smart devices”, Project proposal, FP7 European Research Program, http://www.synaptics.org, December 2006, unpublished.

[3] S. Oniga, A. Tisan, C. Lung, A. Buchman, I. Orha, Adaptive hardware-software co-design platform for fast prototyping of embedded systems, Optimization of Electrical and Electronic Equipment (OPTIM), 2010 12th International Conference on, 2010 , Brasov, Romania, pp: 1004 – 1009.

[4] S. Oniga, A. Tisan, D. Mic, C. Lung, I. Orha, A. Buchman, A. Vida-Ratiu, “FPGA Implementation of Feed-Forward Neural Networks for Smart Devices Development”, Proceedings of the ISSCS 2009 International Symposium on Signals, Circuits and Systems, July 9-10, 2009, Iasi, Romania, pp. 401-404

[5] A. Tisan, S. Oniga, A. Buchman, C. Gavrincea, “Architecture and Algorithms for Synthesizable Neural Networks with On-Chip Learning”, International Symposium on Signals, Circuits and Systems, ISSCS 2007, July 12-13, 2007, Iasi, Romania, vol.1, pp 265 – 268.

[6] HaptiMap project , coordinated by Lund University, Sweden, Safe Corridor Pedestrian Navigation (SCPN) Developer Guide

[7] Jared B. Bancroft, Department of Geomatics Engineering, University of Calgary, Multiple Inertial Measurement Unit Integration for Pedestrian Navigation , UCGE Reports, Number 20320, Dec 2010

[8] http://urna.projects.unoc.net

[9] J. Austin, “RAM-Base Neural Networks”, World Scientific, pp 8-9, 1998

[10] D. Gorse J G Taylor. Review of the theory of prams. In N M Allison, editor, Weightless neural network conferece, pages 13-17, University of York, 1993