

## Implementation of GPS System using Blackfin Processor BF527

Hrishikesh Jha<sup>1</sup>, Bijesh Kumar<sup>2</sup>, Praveen Kumar<sup>3</sup>, Ganashree R Deo<sup>4</sup> and Rajeshwari Hegde<sup>5</sup>

<sup>1,2,3,5</sup>Dept. Of Telecommunication Engineering, BMS College of Engineering, Bangalore, India

<sup>4</sup>D& E, MCE, Bharat Electronics Ltd, Bangalore, India

<sup>5</sup>rajeshwari.hegde@gmail.com

### ABSTRACT

In today's world, human beings want to stay connected with each other and their surroundings. Global Positioning System (GPS) is one of the technologies which is used to determine the exact position of various objects. It is more widely used for military purposes. Various equipments like GPS are increasingly driven by embedded processors, also entering the picture are embedded design needs that include securing intellectual property, lowering power dissipation, minimizing cost, and cutting development time. Blackfin is one such type of processors. In this paper we implement GPS system using BF527. The Software implementation is done using VDK++.

### KEYWORDS

GPS, Blackfin, VDK++

### 1. INTRODUCTION

Till the past decade GPS system was being implemented using microcontrollers. Earlier USART and RS-232 were used for data transfer between processor and GPS receiver. The GPS system had limited functionality with very little reconfiguration flexibility. The accuracy of navigation systems was average before the GPS was used.

The reconfiguration should be possible through software implementation using advanced processors. The implementation using advanced processor like Blackfin will result in faster and accurate results.

In the implementation carried out here, data is continuously received from the satellite by the GPS antenna and is fed into the Blackfin processor BF527 through the u-BLOX GPS receiver. GPS data is then recorded by the Blackfin processor and displayed on the LCD.

Blackfin processor combines a 32-bit RISC- like instruction set and dual 16-bit multiply accumulate (MAC) signal processing functionality with the ease-of-use attributes found in general purpose microcontrollers. This combination enables Blackfin processor to perform equally well in both signal processing and control processing application-in many cases deleting the requirement for separate heterogeneous processors. This capability greatly simplifies both the hardware and software design implementation tasks.

GPS system is implemented using the Blackfin BF527 EZkit evaluation board. Baud rate of 9600 bps is used for receiving the data. The software is written in combination of assembly and C language. VDK++ is used to implement the software code. It is an integrated software development and debugging

environment (IDDE) that enables efficient management of projects from start to finish from within a single interface. Because project development and debugging is integrated, you can move quickly and easily between editing, building, and debugging activities Data is fed through polling in UART and program is invoked through interrupts. The display file uses VaritronixT350MCQB01 as a source file in which all other files are linked and used for LCD display. The paper is organized as follows. Section 2 explains the GPS system. Section 3 deals with data format and hardware interfaces. Section 4 gives an overview of Blackfin processor. Section 5 deals with the UART protocols. Section 6 explains the DSP compiler Visual DSP++. Section 7 gives an overview of the system design. The paper is concluded in section 8.

### 2. GPS SYSTEM

GPS is increasingly becoming popular worldwide for effective use in applications requiring accurate positioning in real time. Equally, concerns still remain about the continued availability of this system for use as the sole means of navigation in many applications due to the single military Ownership of the system. Several organizations internationally have, therefore, been addressing on this **issue** Some efforts have also been made in evolving appropriate solutions towards this. [1]. GPS receivers are used for positioning, locating, navigating, surveying and determining the time and are employed both by private individuals (e.g. for leisure activities, such as trekking, balloon flights and cross-country skiing etc.) and companies (surveying, determining the time, navigation, vehicle monitoring etc.). Various GPS functions include measuring the signal transit time and the distance is measured by comparing the arrival time of the satellite signal with the on board clock time the moment the signal was emitted. The distance  $S$  to the satellite can be determined by using the known transit time  $\tau$ :

Distance= travel time x the speed of light

$$S = \tau \times c$$

In three-dimensional space four satellites are needed to determine a position because we will have four unknown variable in 3-D space:

- Longitude(X)
- Latitude(Y)
- Height (Z)
- Time error ( $\Delta t$ ) [2].

#### 2.1 CALCULATING A POSITION

In order for a GPS receiver to determine its position, it has to receive time signals from four different satellites (Sat1 ...Sat4),

to enable it to calculate signal transit time  $\Delta t_1 \dots \Delta t_4$ . Calculations are effected in a Cartesian, three-dimensional co-ordinate system with a geocentric origin. The range of the user from the four satellites  $R_1, R_2, R_3$  and  $R_4$  can be determined with the help of signal transit times  $\Delta t_1, \Delta t_2, \Delta t_3$  and  $\Delta t_4$  between the four satellites and the user. As the locations  $X_{Sat}, Y_{Sat}$  and  $Z_{Sat}$  of the four satellites are known, the user co-ordinates can be calculated.

Due to the atomic clocks onboard the satellites, the time at which the satellite signal is transmitted is known very precisely. All satellite clocks are adjusted or synchronized with each other and universal time co-ordinated. In contrast, the receiver clock is not synchronized to UTC and is therefore slow or fast by  $\Delta t$ . The sign  $\Delta t$  is positive when the user clock is fast. The resultant time error  $\Delta t$  causes inaccuracies in the measurement of signal transit time and the distance  $R$ . As a result, an incorrect distance is measured that is known as pseudo distance or pseudo-range PSR. Figure 1 shows the 3-D coordinate system of the GPS.

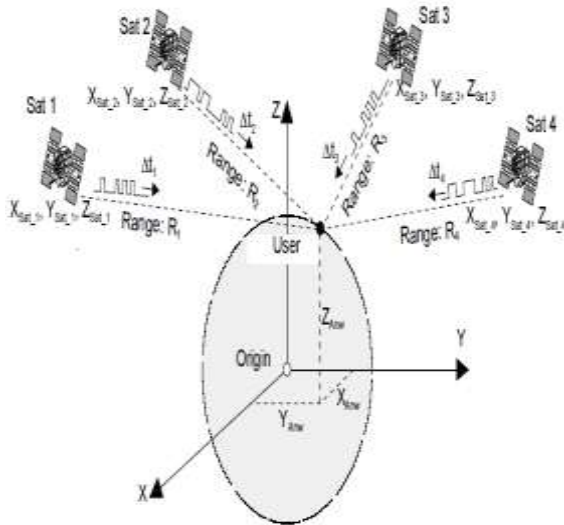


Fig1: 3-D coordinate system

$$\Delta t_{\text{measured}} = \Delta t + \Delta t_0$$

$$\text{PSR} = \Delta t_{\text{measured}} \times c = (\Delta t + \Delta t_0) \times c$$

$$\text{PSR} = R + \Delta t_0 \times c$$

$R$  : true range of the satellite from the user

$c$  : speed of light

$\Delta t$  : signal transit time from the satellite to the user

$\Delta t_0$  : difference between satellite clock and the user clock

PSR: pseudo range

The distance  $R$  from the satellite to the user can be calculated in the Cartesian system as follows

$$R = \sqrt{(X_{Sat} - X_{User})^2 + (Y_{Sat} - Y_{User})^2 + (Z_{Sat} - Z_{User})^2}$$

Thus

$$\text{PSR} = \sqrt{(X_{Sat} - X_{User})^2 + (Y_{Sat} - Y_{User})^2 + (Z_{Sat} - Z_{User})^2} + c \cdot \Delta t$$

In order to determine the four unknown variables ( $\Delta t_0, X_{Anw}, Y_{Anw}$  and  $Z_{Anw}$ ), four independent equations are necessary.

The following is valid for the four satellites ( $i=1 \dots 4$ ):

$$\text{PSR}_i = \sqrt{(X_{Sat_i} - X_{User})^2 + (Y_{Sat_i} - Y_{User})^2 + (Z_{Sat_i} - Z_{User})^2} + c \cdot \Delta t_i$$

### 3. DATA FORMATS AND HARDWARE INTERFACES

GPS receivers require different signals in order to function. These variables are broadcast after position and time have been successfully calculated and determined. To ensure that the different types of appliances are portable there are either international standards for data exchange (NMEA and RTCM), or the manufacturer provides defined (proprietary) formats and protocols.

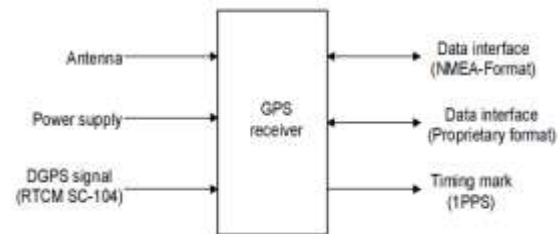


Fig 2 : Block diagram of GPS receiver with interfaces

#### 3.1 DATA INTERFACE

In order to relay computed GPS variables such as position, velocity, course etc. to a peripheral (e.g. computer, screen, transceiver), GPS modules have a serial interface (TTL or RS-232 level). The most important elements of receiver information are broadcast via this interface in a special data format. This format is standardized by the National Marine Electronics Association (NMEA) to ensure that data exchange takes place without any problems. Nowadays, data is relayed according to the NMEA-0183 specification. NMEA has specified data sets for various applications e.g. GNSS (Global Navigation Satellite System), GPS, Loran, Omega, Transit and also for various manufacturers. The following seven data sets are widely used with GPS modules to relay GPS information

1. GGA (GPS Fix Data, fixed data for the Global Positioning System)
2. GLL (Geographic Position–Latitude/Longitude)
3. GSA (GNSSDOP and Active Satellites ,degradation of accuracy and the number of active satellites in the Global Satellite Navigation System)
4. GSV (GNSS Satellites in View, satellites in view in the Global Satellite Navigation System)
5. RMC (Recommended Minimum Specific GNSS Data)
6. VTG (Course over Ground and Ground Speed , horizontal course and horizontal velocity)
7. ZDA (Time& Date)

#### 4. WHY BLACKFIN PROCESSOR?

Processor selection plays a large role in how successful developers are meeting the design requirements of digital surveillance products. Options include application-specific integrated circuits (ASIC), digital signal processors (DSP), and field-programmable gate arrays (FPGA).

Each has pros and cons. The ASIC has limited flexibility to evolve along with ongoing changes in audio/video formats and standards, which limits its usefulness. The conventional DSP possesses very flexible processing capability and is powerful for image processing. However, its traditional signal processing architecture is not optimized for surveillance. With its powerful parallel processing, theoretically the FPGA would excel at high-end surveillance. Power consumption and cost hold the FPGA back from being a successful contender for these applications [7].

Analog Devices' high performance Blackfin® processor family brings something different to this scene: convergent processing. A convergent processor combines the capabilities of a microcontroller (MCU) and a DSP. Such a convergent processor is optimized both for computation on real-time multimedia data flows as well as control-oriented tasks. Based on the Micro Signal Architecture (MSA) jointly developed with Intel, fixed-point Blackfin processors offer programmable 16-/32-bit MCU and DSP functionality in a single-core architecture that enables flexible partitioning among control, network, multimedia, and signal processing. The same development environment extends across the entire Blackfin line, making it faster and less expensive to bring products to market.

#### 5. UART PROTOCOLS

##### 5.1 UART TRANSFER PROTOCOL

UART communication follows an asynchronous serial protocol, consisting of individual data words. A word has 5 to 8 data bits. All data words require a start bit and at least one stop bit. With the optional parity bit, this creates a 7- to 12-bit range for each word. The format of received and transmitted

character frames is controlled by the line control register (UARTx\_LCR). Data is always transmitted and received least significant bit (LSB) first.

##### 5.2 UART TRANSMIT PROTOCOL

Receive and transmit paths operate completely independently except that the bit rate and the frame format are identical for both transfer directions. Transmission is initiated by writes to the UARTx\_THR register. If no former operation is pending, the data is immediately passed from the UARTx\_THR register to the internal TSR register where it is shifted out at a bit rate equal to  $SCLK/(16 \times \text{Divisor})$  with start, stop, and parity bits appended as defined the UARTx\_LCR register. The least significant bit (LSB) is always transmitted first. This is bit 0 of the value written to UARTx\_THR. Writes to the UARTx\_THR register clear the THRE flag. Transfers of data from UARTx\_THR to the transmit shift registers (TSR) set this status flag in UARTx\_LSR again.

##### 5.3 UART RECEIVE PROTOCOL

The receive operation uses the same data format as the transmit configuration, except that one valid stop bit is always sufficient, that is, the STB bit has no impact to the receiver. After detection of the start bit, the received word is shifted into the internal shift register (RSR) at a bit rate of  $SCLK/(16 \times \text{Divisor})$ . Once the appropriate number of bits (including one stop bit) is received, the content of the RSR register is transferred to the UARTx\_RBR registers.

#### 6. VISUAL DSP++

VisualDSP++ is an easy-to-install and easy-to-use integrated software development and debugging environment (IDDE) that enables efficient management of projects from start to finish from within a single interface. Because project development and debugging is integrated, you can move quickly and easily between editing, building, and debugging activities. Key features include the native C/C++ compiler, advanced graphical plotting tools, statistical profiling, and the VisualDSP++ kernel (VDK), which allows a user's code to be implemented in a more structured and easier-to-scale manner. Other features include assembler, linker, libraries, splitter, cycle-accurate and functional-accurate compiled simulators, emulator support, and more. VisualDSP++ offers programmers a powerful yet easy to use programming tool with flexibility that significantly reduces the time to market.

An embedded engineer is often developing on a new platform while maintaining existing products that were likely developed with an earlier version of the tools. VisualDSP++ can be installed discretely an arbitrary number of times at a variety of release levels, allowing engineering to easily switch between current and legacy versions of VisualDSP++. To better integrate to source code control (SCC) systems, VisualDSP++ is able to connect to any SCC provider that supports the Microsoft®

common source code control (MCSCC) interface. VisualDSP++ goes one step further by supporting the control of VisualDSP++ itself within a source code control system [6].

### 7. OVERVIEW OF THE SYSTEM DESIGN

Data is continuously received from the satellite by the GPS antenna and is fed into the Blackfin processor BF527 through the u-BLOX GPS receiver. GPS data is then recorded by the Blackfin processor and displayed on the LCD. This chapter explains the details of implementation.

#### 7.1 HARDWARE

GPS system is implemented using the Blackfin BF527 EZkit evaluation board. Fig3. shows the block level of the implementation with the interfaces used for receiving data. Baud rate of 9600 bps is used for receiving the data.

GPS receiver has to evaluate weak antenna signals from at least four satellites, in order to determine a correct three-dimensional position. A time signal is also

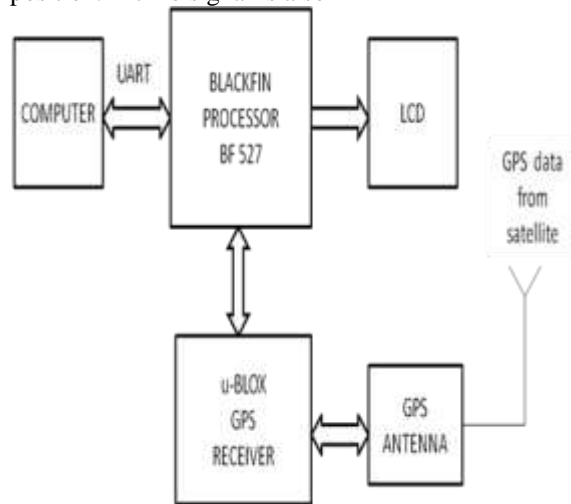


Fig3. Block diagram of GPS system implementation often emitted in addition to longitude, latitude and height. This time signal is synchronised with UTC (Universal Time Coordinated). From the position determined and the exact time, additional physical variables, such as speed and acceleration can also be calculated.

The UART ports provide a simplified UART interface to other peripherals or hosts, providing half-duplex, DMA-supported, asynchronous transfers of serial data. The UART is connected to the external computer and hyper terminal for configuration selection. The software is downloaded through the JTAG interface by the computer. On running the software the hyper terminal displays the menu of options to be selected.

#### 7.2 SOFTWARE

The software will be the brain of the communication system. The implementation of all communication processes and data processing will all be done by software.[8]. The software is

written in combination of assembly and C language. The UART port initialization is written in assembly language, rest in c language. The display file uses **Varitronix\_T350MCQB\_01** as a source file in which all other files are linked and used for LCD display.

Software implementation is divided into three parts

- GPS coding (in C)
- UART interface
- LCD display program

Integration of GPS coding and UART enables to receive and transmit data simultaneously. We use interrupt service routine in the main program. In this program we define all the variables used in GPS program as “extern”. After this we assign the I/O serial ports to be used. We use single color fill to display the data. R\_val , G\_val , B\_val values are set in buffer to get the background color which we have selected green. After this we Set the values “LCD\_LINES\_PER\_FRAME” and “LCD\_PIXELS\_PER\_LINE”. We take this value from the header file “RGB\_data.h”. We include “lookup table” for each characters and symbols used in display. This lookup table is made using 8x8 matrix in which we write the hexadecimal codes for the characters.

Flowchart of main program

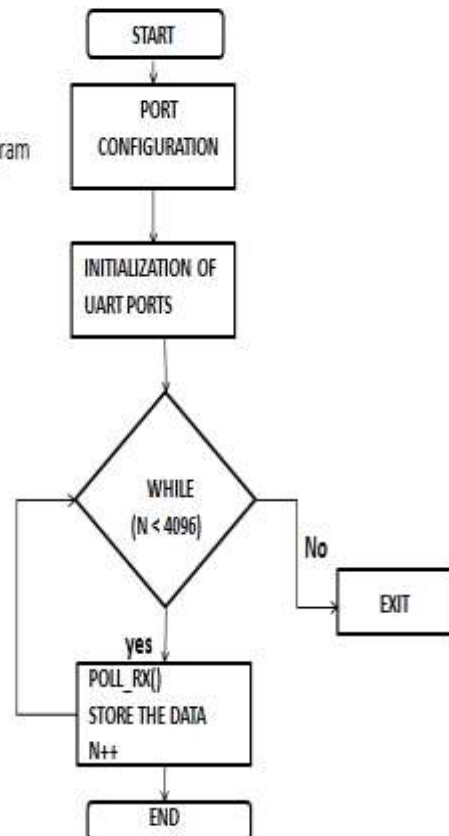


Fig: General Flow diagram of the software CONCLUSION

In this paper, implementation of GPS system using Blackfin processor is presented. This implementation is done on a low power hybrid DSP processor which can also do controller operation. The Blackfin DPS processor increases the accuracy and the transmission rate. VDK has been used to run several codes and is interfaced with the Blackfin processor. With the improvement in processor design and better software tools, implementation can be faster and more accurate.

#### REFERENCES

- [1] R.K. Arora, "A Low Cost Underlay to GPS for a Metropolitan Area Geographic Positioning System", Proceedings of International Radar Conference, IEEE 1996.
- [1] Jean-Marie Zogg "GPS Basics"
- [2] ADSP-BF52x Blackfin Processor Hardware Reference (Volume 1 of 2) Preliminary Analog Devices Revision 0.3 (Preliminary), September 2007
- [3] ADSP-BF52x Blackfin Processor Hardware Reference (Volume 2 of 2) Preliminary Analog Devices Revision 0.3 (Preliminary), September 2007
- [4] ADSP-BF527 EZ-KIT Lite® Evaluation System Manual Analog Devices Revision 1.0, November 2007
- [5] Getting Started With Blackfin® Processors Analog Devices Revision 4.0, November 2008
- [6] Visual DSP++ 5.0 kernel user guide Analog Devices Revision 3.2, March 2009
- [7] [www.analog.com/blackfin](http://www.analog.com/blackfin)
- [8] Leonardo A. Clarke , Joseph Skobla, "A Microcontroller Cellular Based Communication Network for a GPS Error Correction System", 2004 IEEE Aerospace Conference Proceedings.