Introducing the Blackfin Handy Board

Fred G. Martin and **Andrew Chanler**
University of Massachusetts Lowell
Computer Science
1 University Avenue
Lowell, MA 01854 USA

Abstract

The Blackfin Handy Board is a new robot controller inspired by the original MIT Handy Board. The Blackfin Handy Board was developed in collaboration with Analog Devices, Inc., and is based on their Blackfin DSP chip, which combines a 16-bit integer DSP engine with a 32-bit RISC CPU. This paper provides an overview of the architecture of the Blackfin Handy Board, including the hardware design and supported software environments, and results from the Fall 2006 semester of use.

Introduction

The Blackfin Handy Board is a hand-held robot controller board inspired by the original MIT Handy Board. Like the original, the Blackfin Handy Board is an all-in-one solution intended for classroom and mobile robotics projects.

The first Handy Board was developed as part of the MIT LEGO Robot Design Competition (Martin 1994). Its modest feature set, which was powerful at the time, was targeted precisely at the needs of students who were building small classroom robots. It included an 8-bit microprocessor with a 16-bit address space (the Motorola 68HC11), 32K of static RAM, hardware for running DC motors and interfacing to analog sensors, an integral rechargable battery pack, and a custom software environment called Interactive C.

In earlier work presented at this venue, we described an upgrade to the MIT Handy Board, which we called the "Handy Arm," as it was based on a 32-bit ARM processor (Martin & Pantazopoulos 2004). During development of the Handy Arm, we changed course and partnered with Analog Devices.

The result is the Blackfin Handy Board, based on the Analog Devices® Blackfin ADSP-BF537 processor. It is equipped with 64 MB SDRAM and 512 MB of NAND flash, plus a Xilinx Spartan 3e FPGA to support its sensor/motor I/O subsystem.

This paper presents an overview of the hardware design, communications and peripheral interfaces, supported software environments, results from a fall 2006 semester robotics class, and distribution plans.

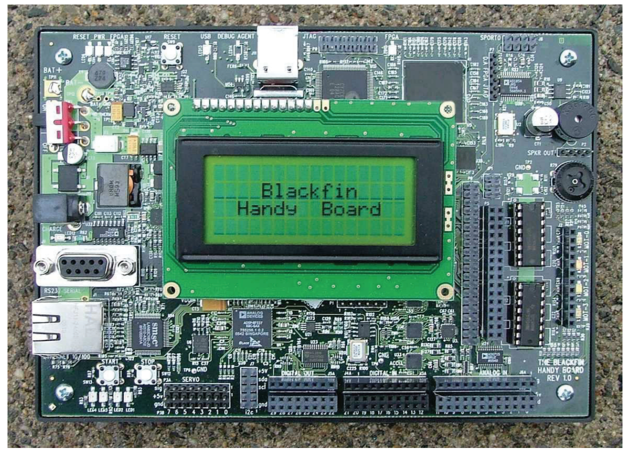

Figure 1: The Blackfin Handy Board.

Hardware

A photograph of the Blackfin Handy Board is shown in Figure 1. The board measures approximately 5 inches $\times$ 7 inches; for those who are familiar with the original Handy Board, it is very close to twice the size. A block diagram of the design is shown in Figure 2.

CPU and Memory

The heart of the design is the Analog Devices Blackfin BF537 chip running at 600 MHz. The Blackfin is remarkable because it combines a 16-bit DSP engine with a modern 32-bit RISC microprocessor core. Thus the single chip is capable of significant signal processing tasks, while also providing good general-purpose control functionality. The Blackfin is also appealing because of its low cost (in OEM quantities) and power management features.

Analog Devices is marketing the chip for widespread, high volume use in consumer applications. For Handy Board users, the Blackfin will allow signal processing applications, such as robot vision, to be supported in software.

In the Handy Board design, the Blackfin is given 64 MB of SDRAM and 512 MB of NAND flash. There is also 1 MB of SPI boot flash.

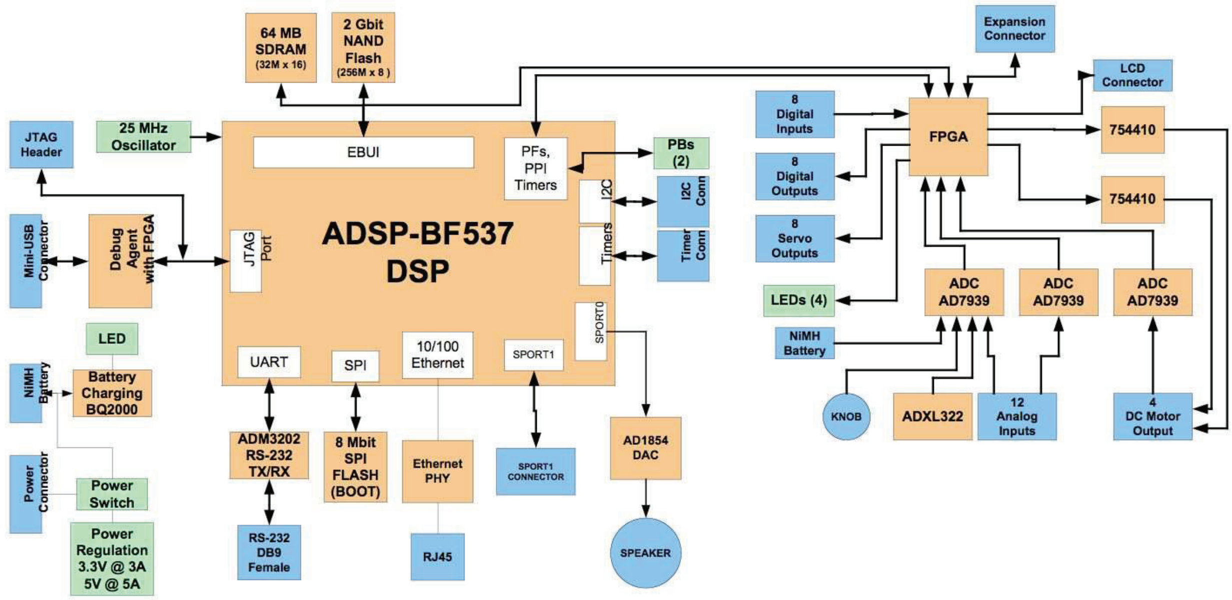

Figure 2: Block Diagram of the Blackfin Handy Board.

FPGA

In addition to the Blackfin processor, the Handy Board includes a Xilinx Spartan 3E-series FPGA (field programmable gate array). This FPGA is responsible for handling all of the sensor-motor I/O. A "board support package" has been developed for the FPGA, implementing a register-mapped interface to the sensor and motor functionality. Thus, there is no load on the Blackfin itself for motor PWM, sonar ping timing, and any of the I/O features.

The FPGA program image is loaded into the FPGA by the Blackfin during board boot. At present, this image is stored in the SPI boot flash; we plan to migrate this to the NAND flash when a filesystem is brought up on the NAND.

Because the FPGA is soft-booted, its program may be improved or changed at will. Also, the signals for the camera interface (discussed below) are routed through the FPGA. At present, the board support package simply provides a signal passthrough.

DC Motors

Two SN754410NE motor driver chips provide drive for four DC motors. Power control is implemented in the reference FPGA program. Both sign-magnitude and locked-antiphase PWM drive are supported on any motor output.

The motor driver design also includes a back-EMF voltage sensing circuit. The FPGA program automatically suspends the PWM drive when a back-EMF reading is performed.

Servo Motors

Outputs for 8 servo motors are provided. The FPGA handles generation of the servo waveforms. The built-in power supply circuit is capable of supplying 5 amperes of current at 5 volts; this current is available to the servo motors.

Analog Sensors

12 Handy Board-style analog inputs are provided, using 10-bit A/D converters. Sensors may be sampled at a rate of approximately 100 kHz. Each input has a 47K pullup resistor and high impedance op-amp drivers.

The Handy Board includes a two-axis accelerometer. The sensor's axes are aligned with the plane of the printed circuit board.

Digital I/O

8 digital inputs and 8 digital outputs are provided. The inputs accept 0 to 5v logic levels.

Sonar Inputs

Each pair of digital inputs and outputs can be configured for use with a 2-wire ultrasonic sonar (e.g., the Devantech SRF04); thus, up to 8 sonars are supported. The FPGA handles sonar triggering and echo timing.

LCD/Buttons/LEDs/Knob

The Handy Board includes a 16×4 character LCD screen. The Blackfin can write characters to the FPGA at full bus speed; the FPGA program handles updating the LCD at its relatively slow bus rate.

Two pushbuttons, four status LEDs, and a thumbwheel knob are available for interaction with humans.

Camera Interface

The Blackfin includes a "parallel peripheral interface" (PPI) port designed for exchanging high-bandwidth data with image-buffer type devices, such as CMOS cameras. The Handy Board includes a connector for interfacing the port directly with low-cost Omnivision cameras (the type used in the CMUcam) using a simple ribbon cable.

Driver code for configuring the camera and retrieving image data is provided. Using DMA, the PPI hardware copies image data directly into the Blackfin memory. Users can then write code to extract features from the image. The Blackfin's DSP engine is ideal for performing 2-dimensional signal processing, and libraries are available from Analog Devices.

As mentioned, the PPI signals are routed through the FPGA, thereby allowing the DMA operation. In an advanced project, a user could create hardware-assisted vision by reprogramming the FPGA.

Audio Output

The Handy Board includes an Analog Devices stereo DAC. One channel of the DAC is run through a small amplifier and an unimpressive board-mounted speaker. The other channel is available via headers, and jumpers allow access to both signal-level outputs of the DAC.

Battery and Power

A ten-cell, 2000 mAH NiMH battery pack is provided. An intelligent charge circuit handles recharging the battery via an inexpensive 24v, 500 mA "wall wart" adapter.

From the battery, board is supplied power through a capable switching power supply. As mentioned, the supply provides +5v at 5A. It also generates +3.3v and +1.8v levels for use by the Blackfin and other digital chips.

Communications

Several subsystems allow the Blackfin Handy Board to communicate with external devices.

Debug Agent

The Debug Agent is an on-board hardware emulator that allows Windows PCs to communicate via the USB port with the Blackfin and its memory subsystems. Windows DLLs allow Visual DSP++ (the standard development environment supplied by Analog Devices) to interact with the board. This circuit replaces a $1000 external emulator that would normally be required to perform development work.

A 14-pin JTAG connector is also provided for use with an external emulator, if desired.

Ethernet

The Handy Board includes a 10/100 BT Ethernet interface. Wireless 802.11 is not directly supported, but may be added via an external Ethernet-to-Wifi bridge.

I^2C

The Blackfin includes support for the I^2C synchronous serial protocol. These signals are broken out to a pair of headers, and also made available on the Omnivision camera interface (configuration of the Omnivision chip is done via I^2C).

Serial

The Blackfin includes two hardware USARTs. One of these is given RS-232 line drivers and a DB9 female connector; the other is available at pin headers.

Software

Various software environments are available for building applications on the Blackfin Handy Board. We anticipate that this already diverse collection of software environments will grow as the board becomes available to practitioners in the field.

VDSP++

Visual DSP++ is the commercial C/C++ compiler developed by Analog Devices. It also includes tools for bootstrapping the system (e.g., loading compiled boot code into the SPI flash). VDSP++ is a solid, well-maintained IDE for code development and debugging.

LabVIEW Embedded

LabVIEW Embedded is a relatively new version of the LabVIEW graphical programming language created by National Instruments. With the "LabVIEW Embedded Module for ADI Blackfin Processors," users can construct powerful application programs on a Windows PC and compile and download them to a Blackfin target system.

We have developed a significant support package for using the Blackfin Handy Board with LabVIEW Embedded. Key benefits include accessibility to non-programmers, and the ability to develop interactive "front panels" that reveal program state (while the board is connected). Results from the fall 2006 semester are described briefly in the next section.

LabVIEW Embedded makes use of the VDSP compiler, and the Blackfin version of the system was developed in a collaboration between National Instruments and Analog Devices Inc.

uClinux

The gcc toolchain has been ported to the Blackfin processor, and a version of uClinux is available. We are presently adapting the release version of uClinux for Blackfin to the Handy Board.

Filesystem

The Handy Board includes 512 MB of NAND flash. This is the type of flash that is ubiquitous in USB keychain drives and diskless MP3 players. Unlike the more-expensive NOR flash, code cannot be executed directly out of NAND flash. NAND flash is has a register-based interface and must be treated like a disk. Thus, NAND flash is only useful when supported by a file system, which provides services like sector management and wear leveling.

Two filesystems developed for NAND flash are supported in the uClinux code base (jffs2 and yaffs). We are in the processing of bringing up these filesystems on the Handy Board. We plan to also make it possible to take a Blackfin executable created by the VDSP environment and store it into the filesystem for loading and execution.

FPGA Code Development

The reference FPGA program was developed using a combination of commercial FPGA development tools (Synplicity

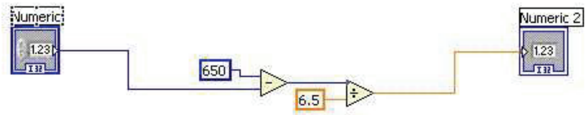

Figure 3: LabVIEW Sensor-to-Motor Normalization VI.

and Active HDL). These tools are complex and expensive (though substantial academic discounts are available).

We have also experimented with allowing students to create FPGA code using the Xilinx WebPACK toolchain. This tool is also complex, but it is available at no charge. Over time, we hope that the reference program may migrate to WebPACK, so that it is more easily modified by users.

Fall 2006 Coursework

The Blackfin Handy Board was used in the lead author's Fall 2006 undergraduate robotics course (91.450 – Robotics I). Robot programming was done exclusively using the LabVIEW Embedded environment. Based on this, students were recruited from the Engineering college as well as the Computer Science department. The resulting class was composed of six computer science students and six engineering students, including one chemistry major and several EE and ECE majors.

As a whole, the course was quite successful. All of the students, including the non-CS students who had little programming experience, built significant control programs for their robots.

Some concepts were ideal for representation in the LabVIEW model. For example, the first programming activities involved building robots inspired by the famous *Vehicles* monograph (Braitenberg 1984). The signal-flow model of the early Vehicles fits perfectly into LabVIEW's signal-flow metaphor.

For instance, in order to implement Vehicle 2 (the light-seeker or light-avoider), a light sensor reading, with an inverted signal over a range from 0 to 650, must be wired to a motor power level, with a range from 0 to 100. Figure 3 shows the LabVIEW program (a "Virtual Instrument") for this. In the full program, this VI is used as a subprocedure, wired between the light sensor icon to the motor power icon.

There were other instances where LabVIEW's programming model made life easier. Multi-threaded code is trivial to create; one simply draws two (or more) while loop structures aside one another. Synchronizing what happens inside of them still requires thoughtful design.

Near the end of the semester, we successfully experimented with rudimentary vision algorithms implemented wholly within LabVIEW. We were able to recognize whether a bright light was on the left or right side of a 320×240 pixel image captured by the Omnivision camera.

There is a learning curve associated with LabVIEW, but once it is climbed, it provides an expressive and effective programming model that is accessible to people without coding backgrounds.

Distribution

The entire Blackfin Handy Board design (with one exception—see below) will be released as open source. This includes the electrical schematics, the printed circuit board artwork, the board support package (including the FPGA program and associated C code), and various other software libraries and drivers.

The only piece of the design that will not be released is the code for the Debug Agent. This code is the property of Analog Devices.

We are arranging for a robotics company located in the United States to handle manufacturing and distribution of the Blackfin Handy Board.

Remarks

The Blackfin Handy Board is a significant departure from the original. It is no longer a simple design that could be hand-soldered by an individual. Instead, it is a state-of-the-art computing platform.

By releasing the bulk of the design as open source, we hope that a similar community will spring up around this new design. Certainly: It was the community of users who made the original design as valuable as it became.

With the new design, we hope to also broaden this community. We are particularly excited about the potential of LabVIEW Embedded to bring hands-on robotics to engineering students who would have no interest in writing C code.

At the same time, the board offers much to the computer science community. With the substantial computational resources that it provides, advanced, graduate-level research projects can be conducted. Also, computer science educators could develop simple programming environments; the board is certainly capable of being programmed in anyone's favorite language.

In sum, we are excited to bring the Blackfin Handy Board to the computer science, engineering, and robotics communities.

Acknowledgments

Analog Devices, Inc. has provided substantial resources in developing the Blackfin Handy Board. Along the way, we also have greatly enjoyed working with the engineers and managers at ADI. Fred would also like to thank his students, who have put up with all sorts of bugs and problems as they have assisted in the development and testing of the board.

References

Braitenberg, V. 1984. *Vehicles: Experiments in Synthetic Psychology*. Cambridge, MA: MIT Press.

Martin, F., and Pantazopoulos, G. 2004. Designing the next-generation Handy Board. In *Proceedings of the AAAI Spring Symposium*.

Martin, F. 1994. *Circuits to Control: Learning Engineering by Designing LEGO Robots*. Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA.