LabVIEW ON SMALL TARGET

Silviu FOLEA*, Marius GHERCIOIU, Horia HEDESIU*,
Crisan GRATIAN**, Ciprian CETERAS**, Ioan MONOSES****

** Technical University of Cluj-Napoca,*
*** National Instruments USA,*
Silviu.Folea@aut.utcluj.ro, Horia.Hedesiu@mae.utcluj.ro
{Marius.Ghercioiu, Gratian.Crisan, Ceteras.Ciprian, Jani.Monoses}@ni.com

Abstract: The science / engineering goal of the "LabVIEW on small Target" project is to demonstrate that a complete, LabVIEW programmable sensor that has intelligence and communication capabilities can be integrated into a very small (as of today's technology) 5cm x 3cm x 2cm package. This involves both evolutionary and revolutionary advances in miniaturization, integration of hardware and software, and energy management. The research is not targeting a particular sensor here, the idea being to be able to use the "LabVIEW on small Target" controller board with any of the new Micro Electromechanical System (MEMS) sensors that combine sensing elements with electronics, and mechanical elements on a silicon chip. National Instruments Corp is in the business of virtual instrumentation, so its focus with regard to mobile small devices is embedded measurement and automation. We, the group of people who author this paper, are part of National Instruments' world wide research effort on embedded measurement and computing. This paper describes the research to produce the smallest target device we can according to today's technology, that is battery powered, can read a sensor, it is easy to program and powerful enough to perform relatively complex – for this footprint - data analysis.

Key Words : LabVIEW, smart sensor, small embedded, low-power.

1. INTRODUCTION

Advances in processor miniaturization, small footprint Operating Systems, and availability of methods to execute LabVIEW graphical code on processors other than traditional X86 architectures, are the three very important technologies that powered our research, and made our "LabVIEW on a small Target" project possible. We have designed and prototyped an optimal – in our opinion - embedded measurement device, "LabVIEW on a small Target", that has the following capabilities:

1) IR connected to Host computer
2) Can be programmed from LabVIEW Host environment via Serial Port
3) Runs application in self-standing mode
4) Performs data acquisition, data analysis, and data communication
5) Battery powered, with efficient power consumption

The LabVIEW on small Target device is: [1]

- Relatively small at 5cm x 3cm,
- Low-power, runs on 3.3V battery,

- Can be programmed from LabVIEW Host (PC or PDA) via serial port (RS-232), or InfraRed port (IR),
- Contains the LabVIEW minimal engine,
- Once programmed starts running the VI independently from the Host computer,
- Supports wireless Data communication with the Host Computer, via Serial or IR,
- Boots up as an instrument, i.e. if a certain VI is downloaded on the LabVIEW Stamp, and then the user turns the LabVIEW on small Target off/on again, the boot-up sequence on the LabVIEW on small Target will end by starting the downloaded VI.

2. LabVIEW ON SMALL TARGET, HARDWARE ARHITECTURE

The LabVIEW on a small Target device is a processor board, built around the AT91FR40182 ARM7 processor from Atmel, with components for IR communication, acceleration sensing, and temperature measurement. The following is the precise feature list of the LabVIEW on a small Target Processor Board:

- Processor: AT91 family (ARM7) - Atmel; [2]
- Memory: 2MB Flash, 256kB of RAM;
- Speed: configurable: 33.3MHz, 3.3MHz, 330kHz;
- Connectivity: serial port, infra-red interface, I2C bus; [3]
- Sensors on board: TCN75 temperature sensor [4] and ADXL202 acceleration sensor [5].

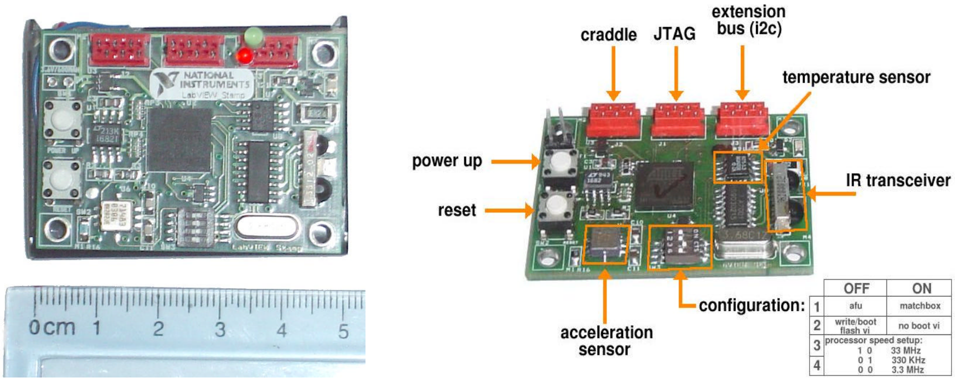

Figure 1 – LabVIEW Stamp front view and component placement

The LabVIEW on a small Target is a processor card that can be programmed from LabVIEW to perform embedded measurement, and data communication. The idea was to make this card as small as possible, and still be able to use LabVIEW to program the board to perform sensor reading and data communication using an IR transceiver. Two very popular MEMS sensors were chosen, the ADXL202 acceleration sensor, and the TCN75 temperature sensor, to place on the processor card.

The ADXL202 (Analog Devices) is a 2-axis acceleration sensor that can measure both static and dynamic acceleration with a resolution of 0.4mg and with an adjustable bandwidth that is set at 100Hz on the current stamp. The ADXL202 has an output of type “duty cycle” with adjustable period. A counter on the ARM is used to read the ADXL202.

The TCN75 can be read over the I2C bus. This sensor has a precision of 0.5ºC in the range [-55ºC…+125ºC]. The reading comes in 2’s complement format.

We believe this architecture for a tiny processor board and the above features list is a good compromise (as of today’s technology) between power consumption and computational power that is needed to perform sensor readings and some elements of analysis on measurement data, on such a small platform. We have previously experimented on an earlier version of this ARM7 chip that had only 1MB of internal flash and 128KB of internal RAM. Most of the work we have done on this particular chip was to port the LabVIEW Real-Time engine, from its very large X86 based size to a very small footprint that fits into the current memory sizes.

The LabVIEW on a small Target board can be looked at as a viable platform for smart sensing, or prototyping for smart sensing platforms where the prototyping work is done in LabVIEW, and you benefit from being able to interface your board with a Host computer or PDA for design validation and benchmarking.

3. LabVIEW ON SMALL TARGET, SOFTWARE ARHITECTURE

The LabVIEW on small Target is running eCos operating system and the LabVIEW minimal engine. The minimal engine is a reduced modular version of LabVIEW Real-Time engine (LVRT) that contains the execution system and a subset of the front panel protocol. Establishing a connection and running a VI on the stamp, is identical to the procedure used by LVRT. [6]

3.1. Downloading a VI to the LabVIEW on small Target

Communication between the Host computer and the LabVIEW on small Target is implemented by using Serial Line IP (SLIP). The SLIP protocol defines a simple mechanism for framing datagrams for transmission across serial lines. SLIP sends the datagram as a series of bytes, and it uses special characters to mark when a series of bytes should be grouped together as a datagram. Maximum transfer rate is 115kbits/s. The physical communication bus is either SLIP over serial cable, or SLIP over IR.

The Host computer is connected to the LabVIEW on small Target via IR or serial cable, as shown below.

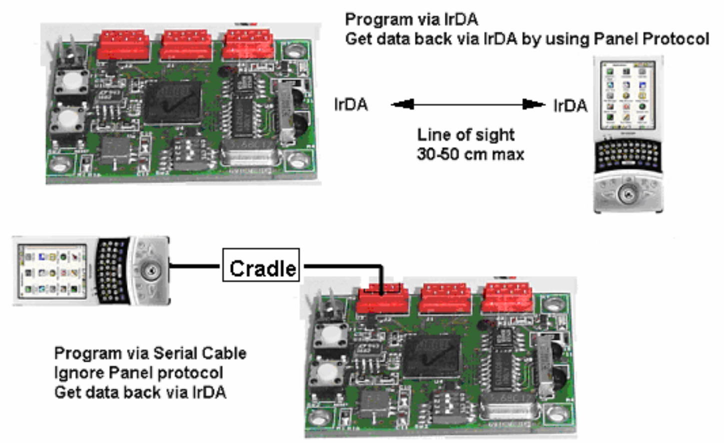

Figure 2 – Communication between LabVIEW Stamp and PDA

You open LabVIEW on the Host, and simply specify execution target. At connection time, LabVIEW Host detects the type of Target being the LabVIEW small engine. In the current implementation you may download your VI, from the Host computer to the LabVIEW on small Target in two ways:

1) SLIP over serial cable

The LabVIEW on small Target device looks like a regular IP target to LabVIEW Host. LabVIEW host connects to the stamp by choosing, from the target selection

Dialog, the stamp's IP address. The TCP/IP connection relies on SLIP over serial cable for data transmission.

2) SLIP over IR

In this configuration the TCP/IP connection relies on SLIP over infrared for data transmission, hence during downloading and talking to the stamp the Host' IR transceiver must be oriented towards the stamp's transceiver. The setup needed for this implies that on the Host there's a SLIP interface attached to the serial UART that drives the IR transceiver.

The drawback of using IR it is limited range and bandwidth. The main advantage is a completely wireless setup. Other scenarios where the medium is fully taken advantage of should use a protocol designed for a wireless medium such as IrDA or Bluetooth. TCP/IP does not work optimally here because it is point-to-point nature and it's assumption of a full-duplex medium among others.

3.2. Run Time communication

Communication between Host and LabVIEW on small Target device is implemented in two modes:

1. Via the LabVIEW front panel protocol which runs over TCP/IP just as in "classical" setups: LVRT over Ethernet, except that in this scenario the TCP/IP connection relies on SLIP over infrared or serial. The host displays data received from the stamp and the user controls the stamp using the controls on the front panel.
2. Via a custom protocol, developed by us, and using the Infra-Red wireless connection between the Host and the stamp. This protocol was designed for LabVIEW on small Target demo purposes. Its main goal is to be as small, simple and effective as possible.

4. THE APPLICATION

In order to demonstrate LabVIEW on small Target device functionality, the following LabVIEW applications has been implemented:

1) Develop under LabVIEW environment an application that will be downloaded on the LabVIEW on small Target device. This application, in our DEMO setting, is named Stamp.vi, again it opens on the Host computer and at run time it gets downloaded on the LabVIEW on small Target device.

2) Develop or open Host PDA monitoring application. In our DEMO, this application is called Zaurus.vi and it has been developed to run on a Zaurus SL-5600 series PDA. The monitoring application will interact with the LabVIEW on small Target downloaded application via IR, in order to get data from the device to the host.

"Stamp.vi" contains a data acquisition section and an infra-red (IR) communication section. The data acquisition section reads data from the acceleration sensor, at a scan rate of 10 samples/sec. The current temperature is read when requested by the Host application. The data points read from the acceleration sensor are stored in a circular buffer, 32Kbytes wide, in stamp's RAM.

The communication section manages data transfers between the LabVIEW on small Target device and the PDA host. The data is transferred via IR to the Host at Host's request.

"Zaurus.vi" is the application that runs on the Host PDA and communicates with one or several LabVIEW on small Target devices by using a custom made protocol for infra-red (IR). This protocol is very similar to the first two layers of the IrDA SIR stack.

At the lowest level there is an IR manager included in LabVIEW. This manager sends/receives IR packets, and verifies the CRC16 control sum for error detection.

The next level of communication is implemented in Zaurus.vi, and communicates with the LabVIEW manager via “Call function library” type calls.

For the purpose of this demonstration you can choose to monitor acceleration (figure 3) or temperature (figure 4).

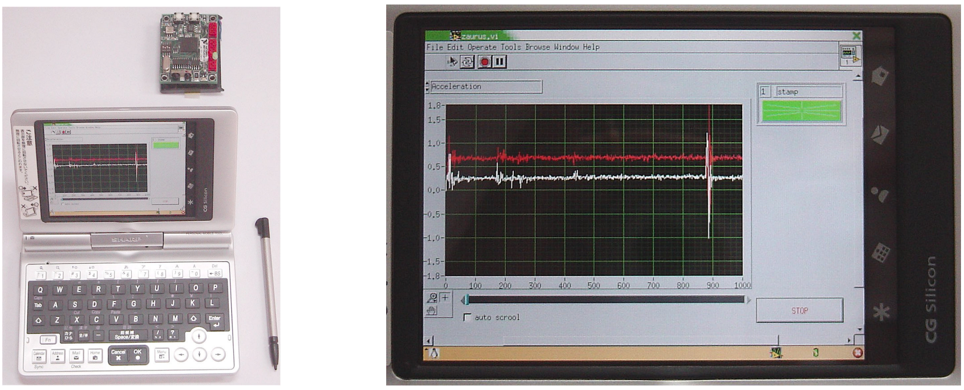

Figure 3 – Monitoring acceleration

Acceleration: Zaurus.vi stores 8192 pairs of points of type (x; y) for each LabVIEW on small Target that it monitors.

X = index of point in array 100 ms apart, and
Y = acceleration in g (gravitational value).

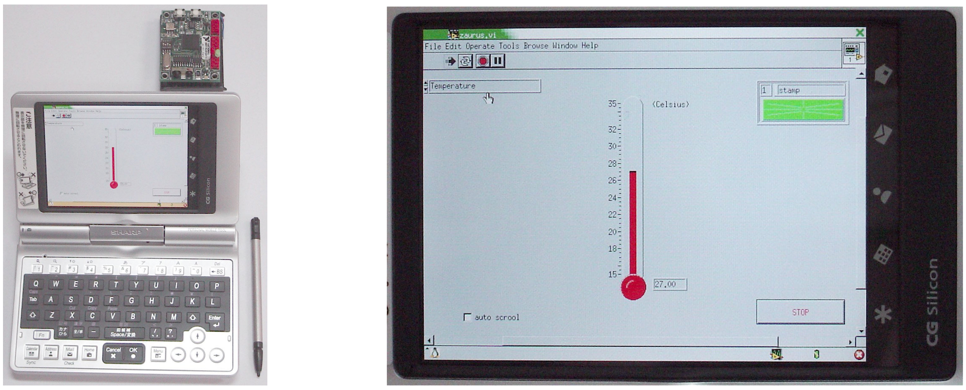

Figure 4 – Monitoring temperature

The data is downloaded from LabVIEW on small Target using IR link and displayed on a graph. The Host will read a maximum of 240 points at a time, from the stamp’s buffer. The Host will continuously send data request messages to the LabVIEW on small Target which will return as many points as it has available in the acceleration data buffer.

Temperature:

In this case the application will read and show one temperature value at a time. The Hosts sends temperature read requests to the LabVIEW on small Target and the later will respond with the current temperature value.

The “Zaurus.vi” also contains an automatic discovery feature that detects the

LabVIEW on small Target and selects the current target for IR communication.

5. POWER CONSUMPTION

The LabVIEW on small Target can run at three frequencies: 33MHz, 3.3MHz and 330kHz. The power consumption on the LabVIEW on small Target was measured as follows:

1) Processor running from Flash at 33MHz uses ~100mA;
2) Processor running from Flash at 3.3MHz uses ~50mA.

With a set of NiMH of the shelf batteries (600mA at 3.6V), the stamp will run for 6 hours, at 33 MHz. We did not implement any power management methods so far but implementing them should increase battery life. The docking station recharges the battery at 100mA/hour rate.

The power consumption and execution time of LabVIEW on small Target loaded with an application that is running continuously, performing 1024 points Finite Impulse Response Filter (FIR) in either Flash or RAM, and no IR transmission/reception is shown in Table 1.

Table 1 – Power consumption and execution time

f_{clk} (MHz)	Flash				RAM		Observation
	$wait_{st}$	test (s)	FIR (ms)	I (mA)	FIR (ms)	I (mA)	
33	2	28.5	150	82.8	30	46.2	FIR:1024 pct. 50 taps
	1	19	85				RAM: flat.obj stored in RAM
10	0	33	180	64.3	90	33.4	

6. CONCLUSION

In the embedded arena today, applications whose total system cost is dominated by software development can benefit greatly from LabVIEW's ease of programming and extensive data analysis libraries.

Here is a sampling of some possible applications of the LabVIEW on small Target processor card, in no particular order:

- Defense-related sensor networks
  - Battlefield surveillance, treaty monitoring, transportation monitoring,
- Inventory Control
  - Know where your products are and what shape they're in any.
- Product quality monitoring
  - Temperature, humidity monitoring of meat products, dairy products,
  - Impact, vibration, temp monitoring of consumer electronics,
  - Failure analysis and diagnostic information.

7. REFERENCE

[1] National Instruments USA, Internal notes.

[2] Atmel Corporation, 2000, AT91ARM® Thumb®Microcontrollers, Rev. 2632A–ATARM–03/02.

[3] Agilent Technologies Inc., 2000, IR 3/16 Encode/Decode IC, 5980-0461E (4/00).

[4] TelCom Semiconductor Inc., 1998, 2-Wire Serial Temperature Sensor and Thermal Monitor, TCN75-2 4/9/98.

[5] Analog Devices Inc., 2000, Low-Cost ±2g Dual-Axis Accelerometer with Duty Cycle Output, C02064–2.5–10/00 (rev. A).

[6] http://ecos.sourceware.org/