MindTel LLC and SenSyr LLC
111 College Place 2-212
Syracuse NY 13244-4100
315-443-5654

Technology Overview

SenSyr, LLC was founded in March of 2003 to commercialize and extend the technology developed by MindTel, LLC, which itself was founded in 1997. Our technology consists of two main components: NeatTools, a visual programming language, and TNG interface devices with their associated sensors and peripherals.

NeatTools Visual Programming Environment (freeware)

NeatTools is an object-oriented, visual programming environment, coded in C++ (with a Java-like thin-layer API). NeatTools, examples, developers' kit, and documentation may be downloaded from www.pulsar.org/2k/neattools at no cost. The language and its toolkit are extensible and expandable. NeatTools modules (visual objects) are selected and dragged into the workspace from toolbox collections. Modules possess properties, parameters, and various data inputs and outputs. Inputs, outputs, and parameters are connected to other modules by links (lines) drawn by the programmer.

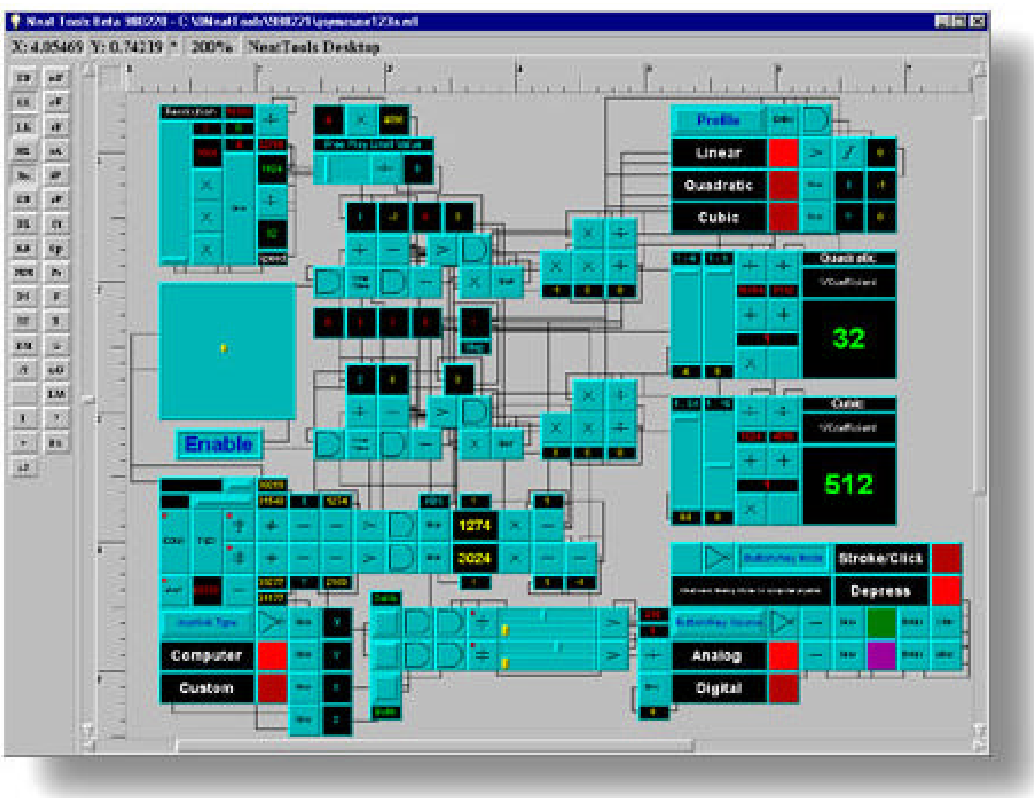

Fig. 1. A NeatTools program.

NeatTools is event-driven. That is, functional elements execute when one or more of their inputs change. This means that NeatTools is also multi-threaded—separated chains of objects without event dependencies will execute nearly simultaneously. Finite state machines are easy to implement using the State module.

Special objects and tools allow NeatTools to access a PC's basic peripheral hardware: keyboard, mouse, joystick, serial port(s), parallel (printer) port(s), sound card, etc. Other modules have been developed to interface with a family of devices we refer to as TNGs.

NeatTools is dynamic. Visual editing and program execution take place concurrently, so there is no need to alternate between editing and execution modes, as in other visual environments. NeatTools is also network-ready, robust, secure, architecture-neutral, and portable.

The built-in, or *internal*, modules (over 200 in number, not counting individual keys), arranged in seven toolbars, include device-support modules, switch and slider modules, calibrator module, internet sockets, state-machine module, timers, graphical displays, arithmetic (integer and real, including transcendental) and logic operations, character generation, multimedia sound, Musical Instrument Device Interface (MIDI) controls, and visual relational database modules with multimedia functions.

There are also *external* modules. External modules are separately compiled visual objects, analogous to those in the main toolbars, loaded at runtime. In the Windows environment, these are implemented as dynamic link libraries (DLLs).

The NeatTools programming model has its roots in the formal input/output automaton model[1]. In NeatTools, module abstraction is offered as a set of class methods for inter-module communication. Functional components (implemented as class objects) of a concurrent system are written as encapsulated modules that act upon local data structures or objects inside object class, some of which may be broadcast for external use. Relationships among modules are specified by logical connections among their broadcast data structures. Whenever a module has updated data and wishes to broadcast the change and make it visible to other connected modules, it should implicitly call an output service function that will broadcast the target data structure according to configuration of logical connections. Upon receiving the message event, the connected modules execute its action engine according the remote data structure. Thus, output is essentially a byproduct of computation, and input is handled passively, treated as an instigator of computation. This approach simplifies module programming by cleanly separating computation from communication. Software modules written using module abstraction do not establish or effect communication, but instead are concerned only with the details of the local computation. Communication is declared separately as logical relationships among the state components of different modules.

---

[1] Nancy Lynch and Mark Tuttle. An Introduction to Input/Output automata. CWI-Quarterly, 2(3):219-246, September 1989. Centrum voor Wiskunde en Informatica, Amsterdam, The Netherlands. Also, Technical Memo MIT/LCS/TM-373, Laboratory for Computer Science, Massachusetts Institute of Technology. Available online at http://theory.lcs.mit.edu/tds/papers/Lynch/CWI89.html.

The base code and internal modules of NeatTools constitute about 54,000 lines of C++ code. The author, Yuh-Jye Chang, did this for his Ph.D. dissertation[2] in computer science at Syracuse University, working in close collaboration with David Warner who conceived the conceptual architecture based on previous generations developed by Warner independently. A number of extensions and improvements are currently in progress by programmers on our team.

Our NeatTools software has been compared to LabView™ (National Instruments; www.ni.com), which is widely used in scientific and industrial research laboratories. Some ask us why we didn't just use LabView instead of developing NeatTools. Our answers are that a) we do not own or have access to the LabView source code so that we cannot optimize it for human-computer interfacing applications, b) LabView is expensive (professional version costs $2,000 plus options), c) we cannot provide LabView for free via our Web site as we do with NeatTools, and d) NeatTools has been designed *ab initio* primarily for human-computer interaction, guided in part by needs of persons with disabilities. Yuh-Jye Chang barely knew about LabView when he wrote the C++ code that produced the NeatTools environment; rather, he based his design on high-end visualization software for use on workstations, notably Application Visualization System (AVS; see www.avs.com). Nevertheless, as part of our inclusive and accommodating approach, we have written LabView interface programs for our TNG devices, so that those who already own, and are proficient with, LabView can use it in their laboratories to interface to our sensors and devices.

TNG Serial Interfaces

TNGs are an evolutionary line of data-acquisition devices, based on Microchip's (www.microchip.com) PICmicro™ microcontrollers. TNG is pronounced "thing" in partial reference to *The Cat in the Hat* by Dr. Seuss, and stands for "totally neat gadget".

NeatTools can operate with certain other devices and TNGs can operate with software other than NeatTools. Any software package that can control a standard PC serial port can communicate with TNGs. That software must be able to continuously assert DTR and RTS which are RS-232 handshake lines. However, since NeatTools and TNGs have been designed, from the outset, to work together, there are clear advantages in doing so. As mentioned above, NeatTools is provided free.

TNGs are easy to install and to use. No card has to be installed on the inside of the computer. Sensors and other devices simply plug into TNGs. Multiple TNGs may be attached to a single computer and used simultaneously. With a single serial cable connecting a TNG to the host computer, cabling is often simplified over installed-board solutions.

[2] Yuh-Jye Chang. NeatTools — a Fine-Grained Data-Flow-Network Programming Environment, Ph.D. dissertation (Electrical Engineering and Computer Science). 2000, Syracuse University: Syracuse, NY.

TNG-3B serial interface

TNG-3B is an input-only device that streams analog (8 channels with 8-bit resolution) and digital data (8 lines) to a host computer connected by an RS-232 serial port at 19.2 kbps. Operating power is derived from otherwise unused RS-232 handshaking and data lines. Each input has its own connector jack: a 3.5 mm stereo phone jacks for analog inputs, and 2.5 mm mono phone jacks for digital inputs. TNG-3B is a 5 volt device (www.sensyr.com/manuals/TNG3bFAQ.pdf).

Fig. 2. TNG-3B serial interface.

TNG-4 serial interface

TNG-4 is much more versatile. Communication is bidirectional. The phone jacks have been replaced with RJ-12 (same size as familiar RJ-11) 6-conductor modular jacks. Each jack has four signal lines plus power and ground. TNG-4 has 8 analog input channels to its analog-to-digital converter (ADC) and 16 digital I/O lines. TNG-4 also has a 4-channel, 0–4.09 V, 8-bit resolution digital-to-analog converter (DAC), and an “expansion port” for communications with certain TNG peripherals. There are 8-, 10-, and 12-bit analog input resolution versions. There are versions that communicate at 57.6 kbps, rather than the default 19.2 kbps. TNG-4 accepts an external power source, and, when operated from such a source, is optically isolated from the host computer. Typically, TNG-4 operates at 5 V, but it can be configured to operate at 3.3 V.

TNG-4 firmware comes in two versions: a) streaming mode (www.sensyr.com/manuals/NewTNG4StreamingMode.pdf) and b) command mode (www.sensyr.com/manuals/CommandMode.pdf). In streaming mode, the device continuously exchanges data (signal and configuration) with the host computer. Each packet contains the current data for all the inputs. In command-mode TNG-4, packets of configuration and output data are simultaneously sent *to* TNG-4 on demand (e.g., when something

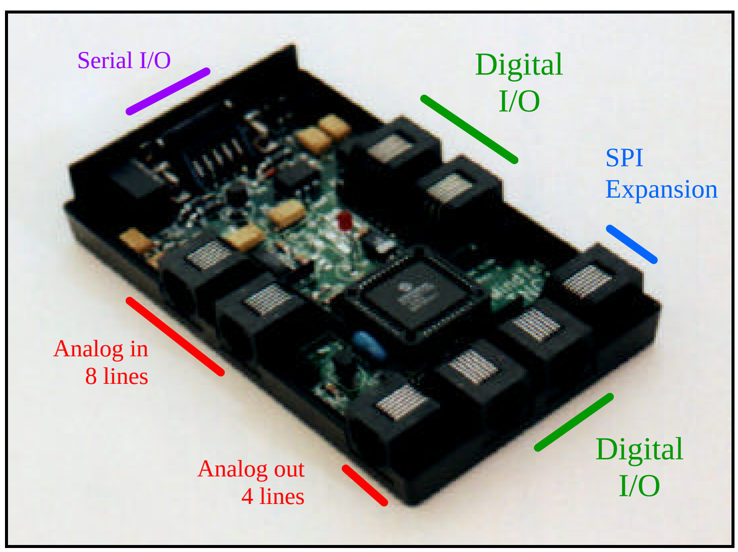

Fig. 3. TNG-4 with cover removed.

changes); similarly, input data are obtained from TNG-4 upon request. In TNG-3B, it made sense to stream, because the host computer was unable to “talk” to the device and alter its behavior.

TNG-4, with its bidirectional communications, can support a query/response method typical of many systems. In applications that cannot tolerate the constant barrage of data, the command mode is an attractive option. It’s particularly useful when TNG-4 is attached to a PDA and when using USB-to-RS-232 adapters. The PDA cannot handle the constant onslaught of data, and streaming mode tends to unnecessarily hog bandwidth in multi-device bus systems like USB. Command mode allows you to get the data you want in the order you want, and only when you request it. For a limited set of data channels, this technique also can implement a faster sampling rate than in streaming mode.

One of the key advantages of TNG-4 is the presence of the Serial Peripheral Interface (SPI) expansion port. This port, alone or in combination with some digital I/O port lines, allows one or more auxiliary devices to be connected via TNG-4.

The SPI protocol was established by Motorola as an efficient way of interfacing “smart” integrated circuits (IC) to their microcomputers without using up the number of lines a parallel bus structure would consume. There are effectively four lines in an SPI connection: clock (SCLK), data out (DOUT), data in (DIN), and select (CS). SPI devices are either bus masters or slaves. The master device provides the clock and drives the select line(s). Select lines are generally active low. That is, the line is normally held at a logic-high level until the device is selected by asserting the CS line to a logic-low level. The device selected by that transition stays selected only so long as the line remains at logic-low. An unselected device is immune to transitions on the clock and data lines. Clock rates are generally in the 1 to 10 MHz range—much faster than a typical RS-232 connection. The DOUT and DIN lines are defined in terms of the master. What’s DOUT for the master is DIN for a slave device, and *vice versa*.

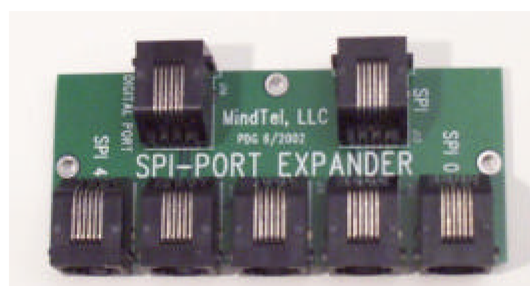

Fig. 4. SPI Port Expander.

TNG-4 acts as the SPI bus master with a default clock rate of 1 MHz. There are a number of auxiliary peripheral boards and devices that can attach to this SPI port including a) type-K thermocouple readout (0–128°C or 0–1024°C), b) thermistor readout (0–50°C), c) LCD display controller, d) 8-motor servo motor controller, e) keypad matrix decoder (up to 8×8), f) 8-channel pulse-width-modulation (PWM) peripheral driver controller, g) SPI-to-RS-232 converter, and h) Sony Playstation-2 dual-shock game controller. An SPI expander board combines a digital I/O port with the SPI expansion port to allow the attachment of up to five separately selected SPI-compatible devices. Current TNG firmware only supports eight separate SPI device select lines. Certain boards, such as the servo-motor controller, allow up to eight boards to be daisy-chained together, each with its own address on a single select line. So, theoretically, a single TNG-4 could communicate with 64 servo-control boards controlling a total of 512 servo motors. Other SPI peripherals are in development: DAC boards, high-resolution ADCs, etc.

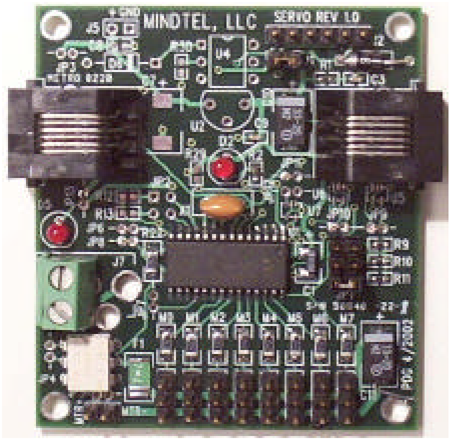

Fig. 5. TNG servo controller.

All this makes TNG-4 a highly versatile platform—supporting rapid prototyping *and* dedicated applications. Depending on the application, some functions of a dedicated application can be incorporated in the TNG-4 firmware directly by reprogramming TNG-4's microcontroller.

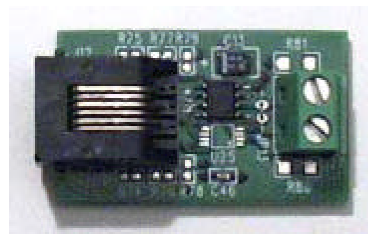

Fig. 6. Thermocouple/thermistor readout.

TNG-5 serial interface

The next generation, TNG-5, is scheduled to debut in mid-November 2003. This version will dramatically increase the capabilities of TNG. A limited model, TNG-5 "lite" will effectively replace TNG-4 with a similar set of I/O; however, the 4-channel DAC will be expunged in favor of a second SPI expansion port. The full TNG-5 will have 16 analog input channels, 16 digital I/O lines, and 2 SPI expansion ports. Both versions of TNG-5 will support 10-bit ADC resolution, USB communications, and faster data rates. Command mode and streaming mode will be jumper-selectable. New firmware will be downloadable through the USB connection. Board population options will allow continued support for RS-232 applications.

TNG-5 connectivity will change yet again. Instead of RJ-12 modular connector, TNG-5 will use 8-conductor, RJ-45 modular connectors; however, TNG-5 Lite will retain use of the RJ12 modular connector.

Either TNG-5 will have about 60 milliamps (mA) of available power when connected as a bus-powered USB device without an external power source. The addition of an external power source allows for up to 750 mA of power for TNG-5 operations.

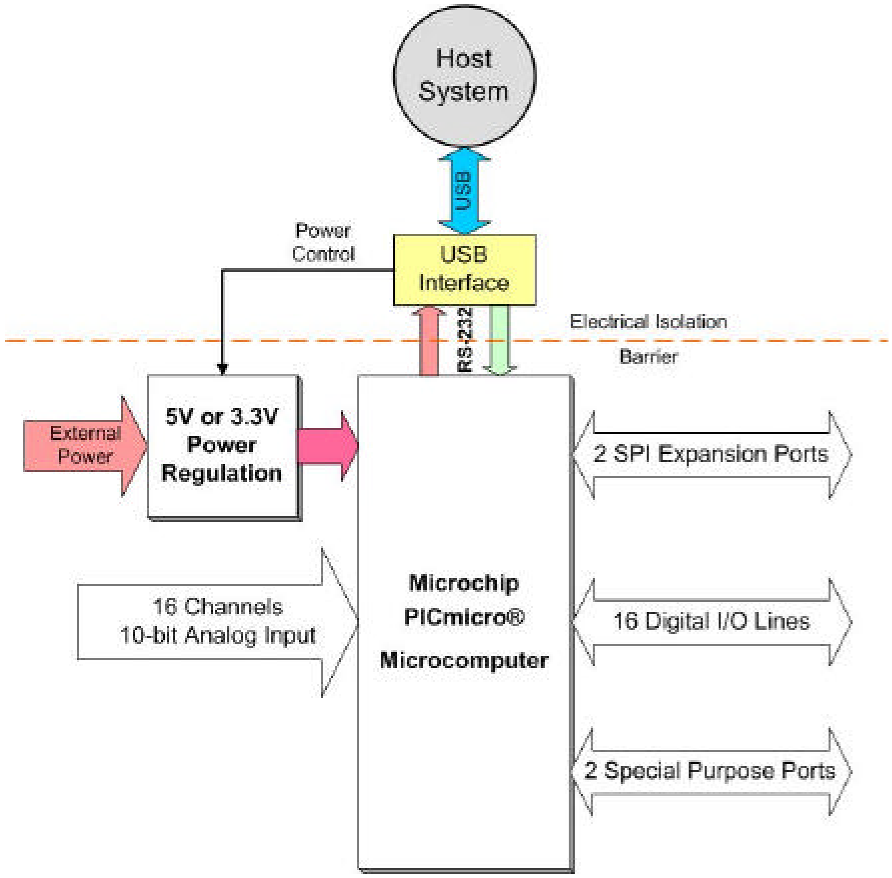

Fig. 7. TNG-5 block diagram.

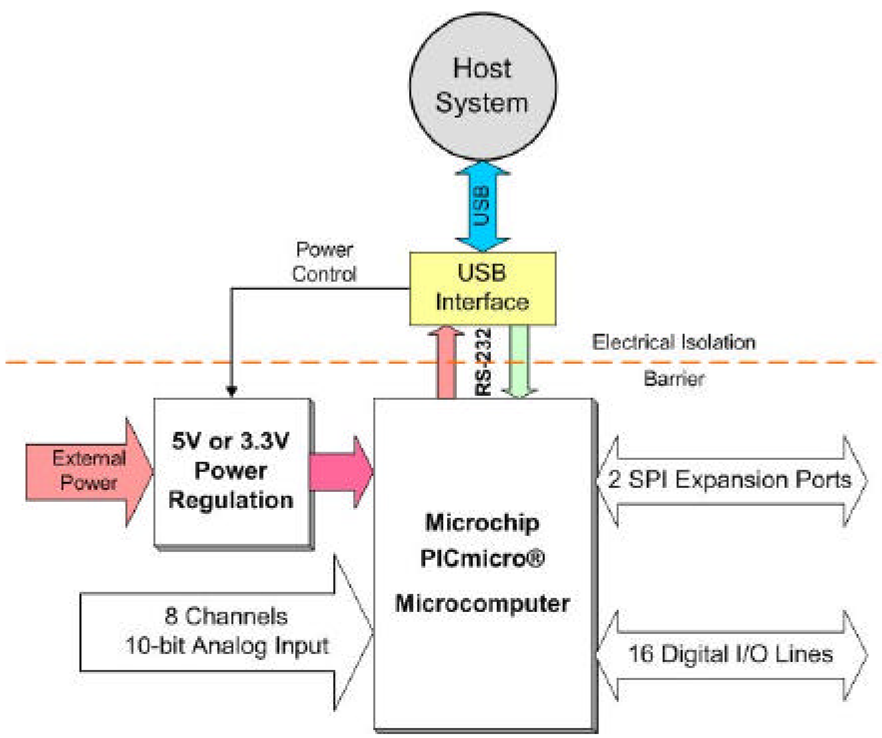

Fig. 8. TNG-5 Lite block diagram.

Sensors and Transducers for TNGs

Devices that can be attached to TNGs fall into three categories: sensors, effectors (actuators), and adapters. Sensors convert real-world phenomena to analog or digital signals that can be connected to TNGs. Effectors convert digital information from the host system into actions in the real world. Adapters allow the resolution of connector or signal incompatibilities.

There are, of course, limitations to what can be interfaced to TNGs. The major limitation has to do with speed (frequency). TNGs can't measure high frequency signals. The upper limit for frequency in a standard TNG-3B or TNG-4 is about 100 Hz (~200 samples per second). TNG-5 should raise that limit, but only by a factor of 5 or so. So, there are many things like audio and video that cannot be used with TNGs—at least not directly. This is not a serious issue, because there are so very many things that can be interfaced well with TNGs, particularly those that take place in human time frames. A car traveling at 65 miles per hour moves less than 6 inches in the time between two successive TNG-3B measurements. The time constant of even the fastest temperature measurement probes is a factor of 10 slower than TNG-3B. A cup anemometer 1 foot across caught in a 70 mph wind makes about 33 revolutions per second, and TNG-3B could make one measurement about every 62 degrees of rotation. A TNG-3B can make more than 80 measurements in the time it takes for a baseball thrown at 95 mph to leave the pitcher's hand and impact the catcher's glove.

Often it is unnecessary to transmit all the points required to measure a high frequency signal. For example, a sample rate of 1000 Hz or more might be necessary to get good time resolution in an ECG signal, but if all you need is the beat-to-beat interval, a dedicated device can do the fast signal processing and then transmit the measured interval to a TNG. Many of our boards adopt that approach. That is, they do a particular task reliably and well, and then transmit their findings to TNG.

Another example is the servo-motor-controller board (www.sensyr.com/manuals/TNGservoManual.pdf). The board simultaneously sends a PWM signal to up to 8 "hobby" servo motors (geared, 4–6 V DC motors with a built-in servo-control module). The pulse for each enabled motor repeats about 60 times a second, and is accurate to better than 4 microseconds. All this work (and the speed required) is offloaded to the servo controller board. All TNG-4 has to do is specify a pulse width for each enabled motor. A stock command mode TNG-4 can send upwards of 320 motor position commands a second.

This actually represents a core belief in our design philosophy: assemblages of small devices that perform dedicated functions in a reliable, but flexible, manner can function synergistically, yielding greater functionality than the sum of the individual components.

TNG Resolution

Another limitation is analog-signal resolution. Eight-bit resolution is standard for TNG-3B and TNG-4. Enhanced versions of TNG-4 offer 10-bit and 12-bit resolution. TNG-5 versions will be capable of 10-bit resolution by default. Admittedly, 8-bit resolution is sometimes inadequate for certain tasks, but 8-bit resolution in other terms translates into 0.4% resolution. 10-bit resolution translates into 0.1% resolution. Again, for human-machine interaction, 0.4% resolution is more than adequate, but this assumes that the dynamic range (maximum-minus-minimum reading) of the analog signal being digitized consumes the full input span of the analog-to-digital converter.

TNGs accept an analog input range of 0–5 volts. If the span of an analog input signal is only 1 volt peak-to-peak, then you can only measure that signal with a resolution of about one part in 50 using an 8-bit ADC, whereas using the full input span yields a resolution in excess of one part in 250. With a 10-bit ADC, having the same 0–5 volt input span, that same 1 volt peak-to-peak signal would have a measurement resolution in excess of 1 part in 200. This is often the reasoning behind digitizing at greater resolutions, namely being able to obtain adequate resolution from signals whose dynamic range is not well-matched to the input voltage span of the ADC.

Sensors

We offer a standard kit of six TNG-3B compatible sensors. This kit includes a switch (digital input), a rotary potentiometer, a linear potentiometer, a bend sensor, a pressure sensor, and a light sensor. The light sensor is a CdS photocell—basically a circuit element that changes its resistance in response to light intensity (a photoresistor). The bend (flex) sensor changes resistance as it is bent through an arc from 0º to about 90º (not a hard bend). The sensor changes from about 10k ohms at 0° to about 40k ohms at 90°. This is the same sort of sensor that was used in the Nintendo-Mattel PowerGlove (invented by Chris Gentile while a student at Syracuse University). Our pressure sensor changes resistance in a fairly linear way with applied force. The sensor element is comprised of 3M Velostat™ plastic film and can be made in many sizes and shapes; normally this antistatic conductive film is sandwiched between sheets of copper foil to which electrode wires are soldered.

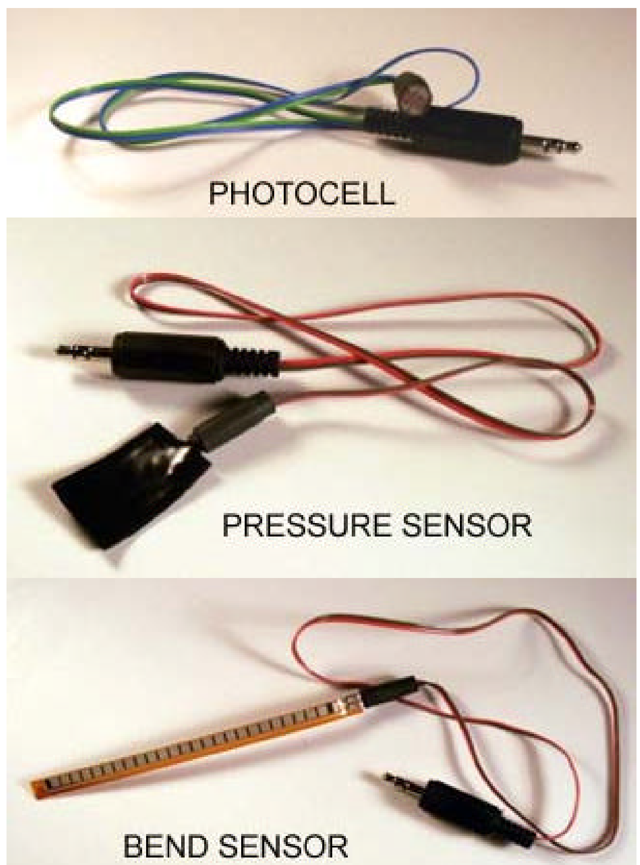

Fig. 9. Some TNG-3B sensors with stereo plugs.

The 2-axis, ±1*g* accelerometer (www.sensyr.com/manuals/TwoAxisTiltSensor.pdf) measures tilt or acceleration simultaneously along two orthogonal axes. The sensor incorporates an ADXL202E MEMS (MicroElectroMechanical Systems) accelerometer IC manufactured by Analog Devices (www.analog.com/Analog_Root/productPage/productHome/0%2C%2CADXL202%2C00.html). When oriented approximately horizontally, the device can measure tilt to a resolution of about ±0.5° for each axis. When using an 8-bit ADC, the limit of tilt resolution is ±1°. This sensor is in its second revision. Future revisions will likely make the device even smaller. SenSyr has plans to produce a 3-axis sensor, and an enhanced sensor with an additional yaw-rate output (a MEMS gyroscope). In due course, SenSyr plans to produce a wireless version.

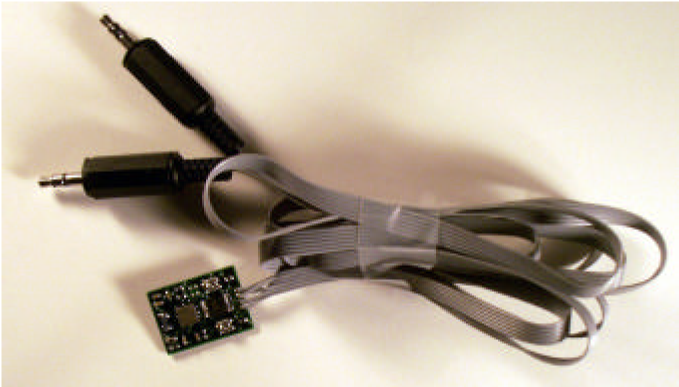

Fig. 10. 2-axis, ±1*g* accelerometer with stereo plugs.

We've already mentioned the thermistor and type-K thermocouple readout boards that attach to the SPI expansion port. TNG-4 could host up to eight such boards. The devices would share a common ground. SenSyr expects to produce several additional temperature measurement devices in the near future.

SenSyr is currently developing an electromyography (EMG; muscle-electric) sensor. This sensor's output signal is proportional to the root-mean-square (rms) voltage in the EMG signal. SenSyr has EEG and ECG sensors in the works too.

SenSyr is currently developing a stand-alone skin-conductance-level (SCL) sensor with built-in calibration circuitry and a baseline suppression circuit. SCL is often referred to as GSR, or galvanic skin response, although technically GSR is a change in SCL in response to a stimulus. The SCL device uses USB to communicate with the computer. The subject electrodes are electrically isolated from the computer for safety. The subject side of the circuit is powered with a 9V alkaline battery.

SenSyr has a prototype 2-channel amplifier board for the Pneumotrace II respiration transducer (www.ufiservingscience.com/DS11321.html).

We have plans for an expanded line of sensors: pressure transducers, force transducers, Hall Effect sensors, giant magnetoresistive (GMR) sensors, humidity sensors, temperature sensors, gas sensors, photointerrupters, motion sensors, LVDT amplifiers, anemometers, load-cell amplifiers, capacitive sensors, and E-field sensors. We've already constructed prototypes of many of these devices.

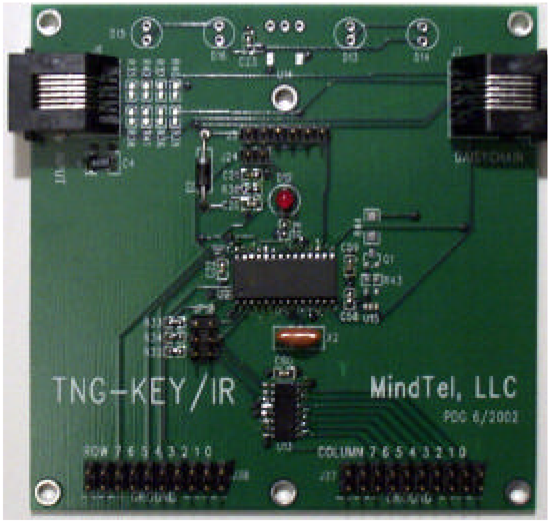

Fig. 11. Keypad matrix decoder.

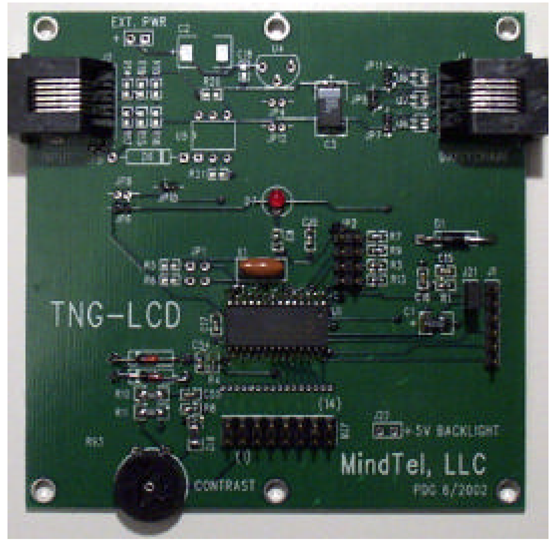

Fig. 12. LCD controller board.

SenSyr also has a line of effectors (or actuators). Items currently available include a 4-channel relay board, an 8 motor servo motor controller, a keypad matrix decoder (up to 8×8), an 8 channel pulse-width-modulation (PWM) peripheral driver controller, a SPI-to-RS-232 converter, a LCD controller, and two types of piezoelectronic alarms.

The 4-channel relay board (www.sensyr.com/manuals/TNGRelayManual.pdf) can accommodate AC or DC optoelectronic relays. The board connects to any one of the digital I/O ports on TNG-4 or TNG-5.

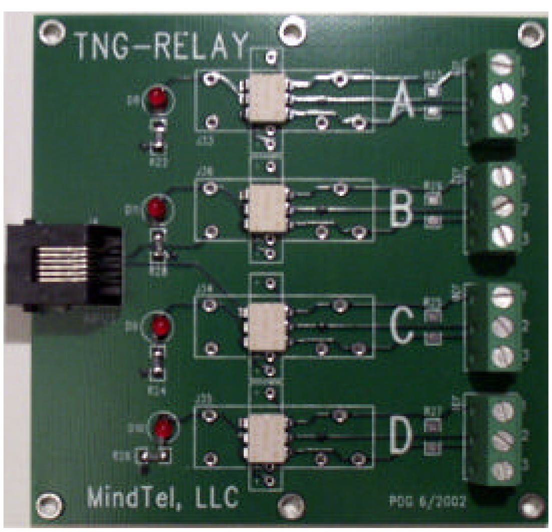

Fig. 13. Relay board with DC relays

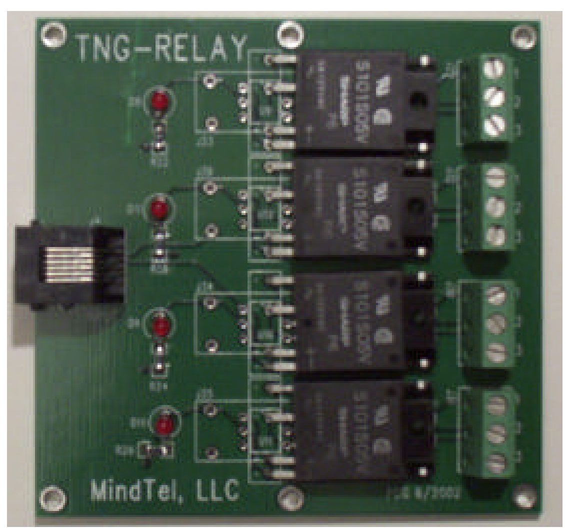

Fig. 14. Relay board with AC relays.

The vibrotactile array driver board (www.sensyr.com/manuals/TNG_VBAManual.pdf) uses PWM to control the speed of DC motors. The drive is unidirectional, but works well in conjunction with vibrator motors. We've used this board to interface consumer back-massage units, as made by Homedics, with TNG-4. These units often have 10 motors, so we have to use two vibrotactile boards daisy-chained together. The unit shown in Fig. 16 actually has one vibrotactile board and one relay board. The relay board activates either of two "thumper" motors.

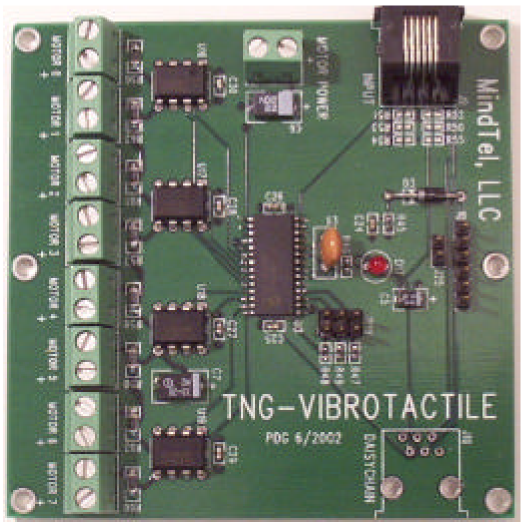

Fig. 15. Vibrotactile-array driver.

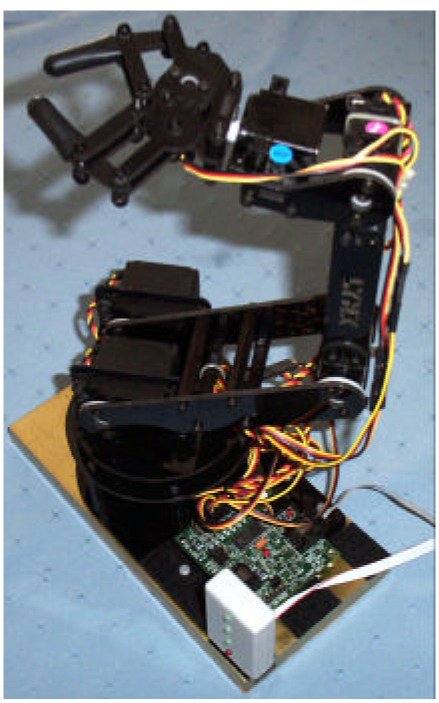

Fig. 5. Robot arm controlled with TNG-servo board.

Fig. 16. Modified Homedics 10-motor massage unit.

We have used our servo motor control board to interface a Lynxmotion (www.lynxmotion.com) robot arm (Fig. 17).

SenSyr is expanding the list of effectors to include LED arrays, stepper and DC motor controllers, buffer drivers, DACs, and non-volatile electronic potentiometers.

To facilitate connecting TNG-3B compatible sensors to the new TNG-4 modular connectors, we provide a series of adapter boxes. SenSyr anticipates producing a new line of adapter boxes to make connecting to the new TNG-5 units easier.

We also plan on making screw terminal, alligator clip, BNC, RCA, and other adapter boards. Another category of adapter will allow construction of small active and passive analog-input circuits.

Fig. 18. TNG-4 adaptor boxes

Universal Preamp

SenSyr has another adapter board of sorts: the Universal Preamp Board (circuit diagram: www.sensyr.com /manuals/UPdoc.pdf). Nowadays it's difficult to prototype analog circuits because of the prevalence of surface mount parts. The universal preamp (UP) board allows a multitude of analog circuits to be prototyped.

There are two separate input analog amplifier chains, each starting with an instrumentation amplifier IC. There are myriad jumpers and population options that allow simple filters and amplifier circuits to be constructed. A voltage reference and amplifier section allows even more flexibility. There is a power section that allows for generation of –5V or ±10V from an input power voltage of +5V. Footprints for memory ICs, DACs, and a PIC16F873/PIC16F73 microcontroller (with provisions for on-board serial programming) round out the board's capabilities.

This board is made even more versatile by the fact that you do not have to "color within the lines", so to speak. By cutting and jumpering and attaching other boards, the uses of this board are extended.

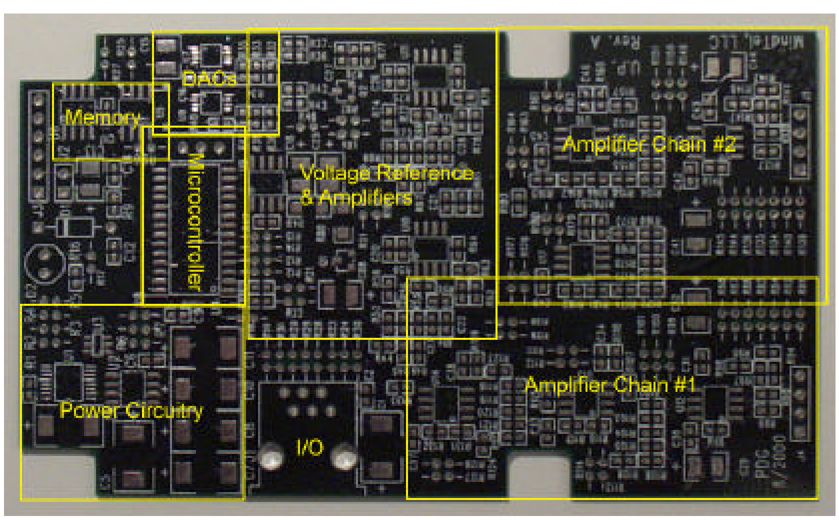

Fig. 19. Universal Preamp Board.

SenSyr plans to offer several versions of this board. One version would be based on the Maxim MAX1463 low-power, two-channel sensor signal-processor IC (pdfserv.maxim-ic.com/en/ds/MAX1463.pdf). Another version would have better support for SPI slave communication, and a fourth version would support a small Xilinx field-programmable gate-array IC (FPGA).