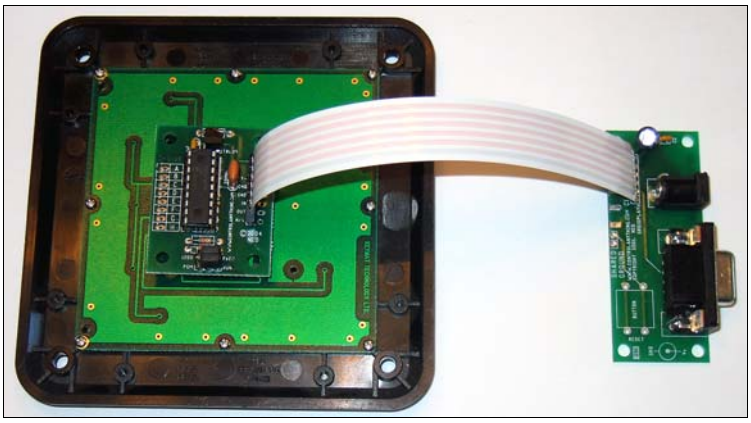
ABTPRO

16-Key RS-232 Keypad Encoder



ABTPRO shown connected to a Storm Keypad (left) and an RSIO (right, included with the QS12-F6 Quick Start Kit).

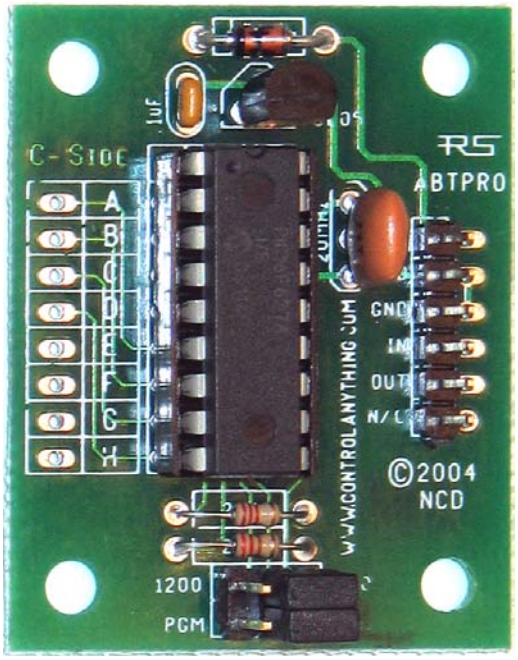
The ABTPRO is our second generation ASCII Keypad Encoder board. Designed to take the place of the original ABT, the ABTPRO offers a very simple, low-cost solution for converting key presses from a 16-button keypad into RS-232 ASCII character codes.

The ABTPRO has two modes of operation, set by changing the position of the PGM/ RUN jumper and power cycling the ABTPRO board.

PGM: Program Mode
Program Mode is used to configure the ABTPRO ASCII Character Output Range. You can also retrieve the stored setting from the ABTPRO while in PGM Mode. The Keypad encoder does not function while in this mode, this mode is used to set device parameters only.

RUN: Runtime Mode
While in runtime mode, the ABTPRO will send an ASCII character code every time a key is pressed. The ASCII character code output range is defined while in Program Mode. A new ABTPRO board will output ASCII characters 0-15 when a corresponding key is pressed on the keypad.

Keypad Characteristics:
The current version of the ABTPRO is only capable of sending one ASCII character code per key press. Individual output codes cannot be assigned, only an output range. The ABTPRO does not support hold and repeat key pressed. You must release a key before the next key press will function. Holding down multiple keys does not trigger a combinational output.



Converts Key Presses into ASCII Output
Encodes up to 16 Button Keypads
User-Programmable Output Range
7-18VDC Operation
100 ma Maximum Power Consumption
Storm Keypad Compatible Interface
1200 or 9600 Baud Operation

Keypad Interface

ABTPRO	*Storm Keypad*	*Description*
A	*Pin 8, Row C*	*Scans Row C*
B	*Pin 7, Row D*	*Scans Row D*
C	*Pin 6, Col. 4*	*Scans Column 4*
D	*Pin 5, Col. 3*	*Scans Column 3*
E	*Pin 4, Col. 2*	*Scans Column 2*
F	*Pin 3, Col. 1*	*Scans Column 1*
G	*Pin 2, Row B*	*Scans Row B*
H	*Pin 1, Row A*	*Scans Row A*

A/1	2	3	4
B			
C			
D			

Keys are Scanned using Cross/Point Intersection method whereby a Row Must be Connected to a Column to trigger an ASCII output. The processor will wait for the key to be released before scanning the keypad again.

The ABTPRO is compatible with any matrix style keypad with 16 keys or less. Smaller matrix keypads may be used, simply omit connection to rows and columns that do not exist your matrix keypad. The interface of the ABTPRO is programmed for direct connection to the following Storm series keypads, available from Keymat Technology (www.keymat.com):

GS0401 Grey 4x1 Keypad
GS0402 Black 4x1 Keypad
GS1201 Grey 4x3 Keypad
GS1202 Black 4x3 Keypad
GS1601 Grey 4x4 Keypad
GS1602 Black 4x4 Keypad

The ABTPRO is Compatible with ANY Matrix (Column/Row Intersection Method) Keypad up to 16 Keys.

Jumper Settings:
Jumpers are used to define the Baud Rate and Program or Runtime Operation of the ABTPRO. If a jumper is changed, the ABTPRO MUST be power cycled by removing power from the controller, then reapply power to reboot the ABTPRO CPU. Jumper settings are only read by the CPU when power is first applied to the controller.

The ABTPRO supports 1200 or 9600 baud operation, and 2 program modes: PGM and RUN

PGM: Program Mode

Program Mode is used to define the ASCII Character Output range of the ABTPRO. By default, the output range is ASCII Character Codes 0 to 15. Program Mode allows a computer to send and receive configuration data, the keypad is disabled while in this mode. The current version of the ABTPRO supports the following configuration commands:

Program ASCII Character Range:
The following command configures the ABTPRO ASCII Output Range. By default the Range is 0-15. By defining a new range, different ASCII characters can be transmitted when a key is pressed. For example, Defining a range of 240 will send ASCII character code 240 when key 1 is pressed. ASCII Character 255 will be sent when key 16 is pressed on the keypad.

```
Send ASCII 1              'Program Range Command
Send ASCII 0-240          'Program First Key Output Value
```

Read ASCII Character Range:
This command is used to retrieve the ASCII output value of Key 1 on the keypad. For instance, if the device returns a value of 0, the ASCII output range will be 0 to 15. If the ABTPRO returns a value of 240, the ASCII output range will be 240-255.

```
Send ASCII 2              'Get Range Command
Receive 1 ASCII Character 'Device Sends Key 1 Value
```

RUN: Runtime Mode

In runtime mode, the ABTPRO becomes a spontaneous output device. If a column pin is crossed with a row pin, an ASCII output will be generated. The firmware on the ABTPRO will then wait for an electrical release of the key before scanning for new key presses. The ASCII output range is defined in PGM mode (see above). The electrical output of the ABTPRO is current limited, direct drive TTL RS-232 and is suitable for use with most embedded systems and desktop PCs. The RS-232 output is not compatible with Macintosh Systems without additional hardware.

ABTPRO Characteristics:

Voltage:	7-18VDC
Current:	<100 ma
PCB Size:	1.375” x 1.775”
Mounting Holes:	4 x .150” diameter
Hole Locations:	The center of each hole is inset from each corner .175” x .175”
Output Type:	Current Limited +5V TTL, 1200 or 9600 baud Direct Drive RS-232
Output Format:	Programmable ASCII Range, Single Byte Output

Sending Serial Commands

NCD Controllers are capable of sending and receiving data via RS-232 serial communications. The ABTPRO controller is compatible with just about any computer or microcontroller ever produced, including the Amiga, Basic Stamp, and of course, Windows & DOS based machines. The ABTPRO is not compatible with Macintosh systems without additional hardware.

Regardless of the system you are using, you will need access to a programming language that supports program control of the serial port on your system.

A terminal program is not suitable for controlling most NCD devices. Commands should be sent using ASCII character codes 0-255 rather than ASCII characters (A, B, C etc.). See "ASCII Codes vs. Characters" on this page.

Most systems require you to open the appropriate serial port (COM port) prior to sending or receiving data.

Because there are so many different ways to send and receive data from various languages on various platforms, we will provide generic instructions that can be easily converted to your favorite language.

For example, if this manual says "Send ASCII 254", the user will need to translate this instruction into a command that is capable of sending ASCII character code 254.

To Send ASCII 254 from Visual Basic, you will use the following line:

MSComm1.Output = Chr$(254)

In Qbasic, you can send ASCII 254 using the following line of code:

Print #1, Chr$(254);

Note that sending ASCII character code 254 is NOT the same as sending ASCII characters 2, 5, and 4 from a terminal program. Typing 2, 5, and 4 on the keyboard will transmit three ASCII character codes.

In your program, you may want to ask a device for information stored within the NCD controller. If so, your programming language will support commands for reading data from the serial port.

For your convenience, we have provided several programming examples in Visual Basic 6 for controlling many of our products. These examples should greatly speed development time. You may want to visit **www.controleverything.com** for the latest software and programming examples.

Programming examples for our devices are much more extensive for Visual Basic 6 users than for any other programming language. If you are not a VB programmer, you may consider looking at the VB6 source code, as it is easily translated into other popular languages.

Regardless of your programming background, the provided Visual Basic 6 source code is very easy to understand and will likely resolve any communication questions you may have. VB6 programming examples may be viewed in any text editor.

ASCII Codes vs. Characters

The differences between ASCII codes and ASCII characters tend to generate a lot of confusion among first-time RS-232 programmers. It is important to understand that a computer only works with numbers. With regard to RS-232 data, the computer is only capable of sending and receiving numbers from 0 to 255.

What confuses people is the simple idea that the numbers 0 to 255 are assigned letters. For instance, the number 65 represents the letter A. The number 66 represents the letter B. Every character (including numbers and punctuation) is assigned a numeric value. This standard of assignments is called ASCII, and is a universal standard adopted by all computers with an RS-232 serial port.

ASCII characters codes can be clearly defined as numbers from 0 to 255.

ASCII characters however are best defined as letters, A, B, C, D, as well as punctuation, !@#$%, and even the numbers 0-9.

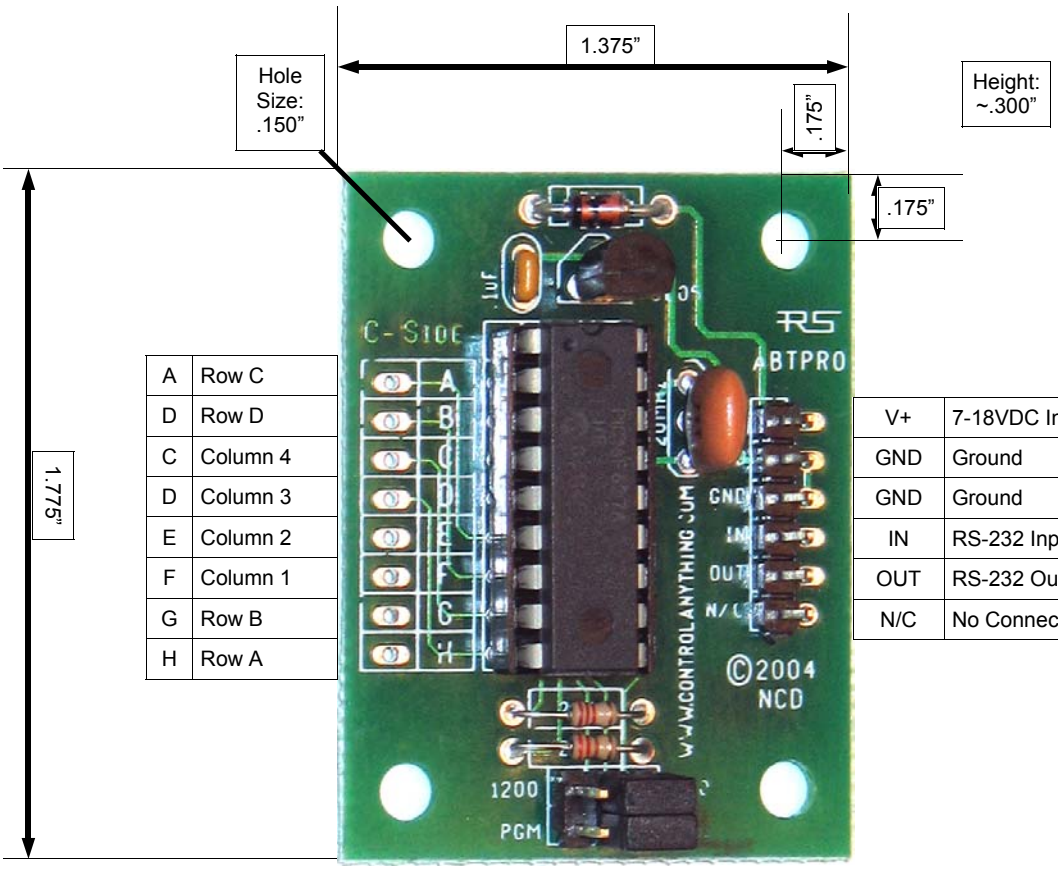
Virtually all programming languages permit you to send ASCII in the form of letters or numbers. If you wanted to send the word "Hello" out the serial port, it is much easier to send the letters H, e, l, l, and o than it is to send the ASCII character codes that represent each letter.

For the purposes of controlling NCD devices however, it is much easier to build a numeric command set. Especially when communicating to devices where you want to speak to lots of outputs (which are numbered), inputs (which are also numbered), or control specific devices using their device number (from 0 to 255).

Put simply, it is easier to control NCD devices using ASCII character codes 0 to 255 than it is to use ASCII characters A, B, C, D, etc.

Because terminal programs are ASCII character based, it may be difficult to generate the proper series of keystrokes that would be necessary to activate a particular function. Therefore, they are not suitable for controlling NCD devices. In a real world control application, a terminal program would not likely be used to control NCD devices anyway. Therefore, a programming language that supports the transmission and reception of ASCII character codes 0 to 255 is highly recommended.

Outline Dimensions ABTPRO



A	Row C
D	Row D
C	Column 4
D	Column 3
E	Column 2
F	Column 1
G	Row B
H	Row A

V+	7-18VDC Input
GND	Ground
GND	Ground
IN	RS-232 Input from Computer’s RS-232 Output
OUT	RS-232 Output to Computer’s RS-232 Input
N/C	No Connection

Jumper Settings

Jumpers settings are read by the ABTPRO only when power is applied. For this reason, it is important to power cycle the board to read the new jumper settings. The ABTPRO a 1200 or 9600 baud selection jumper as well as a PGM/RUN jumper. Under normal operation, this jumper should be set to RUN, set to PGM only if you need to make configuration changes. Configuration information is stored in non-volatile EEPROM and will not be lost when power is removed.

5-Year Repair or Replace Warranty

Warranty
NCD Warrants its products against defects in materials and workmanship for a period of 5 years. If you discover a defect, NCD will, at its option, repair, replace, or refund the purchase price. Simply return the product with a description of the problem and a copy of your invoice (if you do not have your invoice, please include your name and telephone number). We will return your product, or its replacement, via UPS Ground Service in the US and Canada Only. Additional shipping charges will apply to international customers.

This warranty does not apply if the product has been modified or damaged by accident, abuse, or misuse.

30-Day Money-Back Guarantee
If, within 30 days of having received your product, you find that it does not suit your needs, you may return it for a refund. NCD will refund the purchase price of the product, excluding shipping/handling costs. This guarantee does not apply if the product has been altered or damaged.

Copyrights and Trademarks

Disclaimer of Liability
NCD is not responsible for special, incidental, or consequential damages resulting from any breach of warranty, or under any legal theory, including lost profits, downtime, goodwill, damage to or replacement of equipment or property, and any costs or recovering, reprogramming, or reproducing any data stored in or used with NCD products.

Technical Assistance
Technical questions should be e-mailed to Ryan Sheldon at ryan@controlanything.com. Technical questions submitted via e-mail are answered several times daily. Technical support is also available by calling (417) 646-5644 from 9:00 A.M. to 4:00 P.M. Central Standard Time.

NCD Contact Information

Mailing Address:
National Control Devices
P.O. Box 455
Osceola, MO 64776

Telephone:
(417) 646-5644

FAX:
(417) 646-8302

Internet:
ryan@controlanything.com
www.controlanything.com
www.controleverything.com