TTY

A terminal independant

screen interface

Revised: 05-Jun-92

Dunfield Development Systems

----------------------------

High quality tools for

Embedded Development

at low prices.

http://www.dunfield.com

Copyright 1983-2005 Dave Dunfield

All rights Reserved

The TTY interface

TABLE OF CONTENTS

Page

1. INTRODUCTION 1

2. TTY INTERFACE ROUTINES 1

2.1 ENTRY 1

2.2 PUTCHR 1

2.3 GOTOXY 2

2.4 GETKEY 2

3. THE TTYPATCH UTILITY 3

3.1 Entering definitions 4

4. SAMPLE DEFINITIONS 8

4.1 VT100 8

4.2 VC4404 9

The TTY interface Page: 1

1. INTRODUCTION

The TTY interface routines provide a set of functions for

accessing the terminal screen which make use of an internal table to

determine the actual codes sent and received to and from the

terminal.

A separate utility program "TTYPATCH" is provided which updates

that internal table, allowing the terminal control codes and function

keys to be modified in the programs executable image ".EXE" file.

This allows screen oriented programs such as the "EDT" editor to be

customized by the user to operate on virtually any terminal.

The assembler source for the "TTY" interface routines is provided

on the CUBIX distribution diskette in the file "[EXAMPLES]TTY.ASM",

and should be included in any programs makeing use of this interface.

2. TTY INTERFACE ROUTINES

The following subroutines comprise the TTY interface:

2.1 ENTRY

The "ENTRY" routine is not actually a subroutine, but must be

included as the absolute first code to occur in the ".EXE" file.

It serves two purposes: First, it outputs the initialization

string to the terminal to set up operating modes and/or function

keys, Second, the "TTYPATCH" utility keys on this code to

determine if this program uses the "TTY" interface, and the actual

address of the terminal independance tables located within the

image.

2.2 PUTCHR

This routine outputs the character in 'A' to the console

terminal. If the character is in the range of normal ASCII

characters ($00 to $7F), it is output to the console without

translation.

If the character output is greater than $7F (Has the high bit

set), the following operation is performed by outputing the

appropriate codes to the terminal:

$80 - Clear terminal screen

$81 - Clear from cursor to end of line

$82 - Clear from cursor to end of screen

$83 - Turn on special video mode (reverse etc..)

$84 - Turn off special video mode

$85 - Scroll the screen forward one line

The TTY interface Page: 2

2.3 GOTOXY

This routine positions the cursor on the terminal to the

indicated X/Y position passed in 'A' (X = 0 - 79), and 'B' (Y =

0-23).

2.4 GETKEY

This routine gets a single character from the console terminal.

Special code sequences representing function keys are trapped,

decoded, and returned as a single character of a value greater

than $7F (High bit set), with the following meaning:

$80 - Up arrow

$81 - Down arrow

$82 - Right arrow

$83 - Left arrow

$84 - Page up

$85 - Page down

$86 - Page right

$87 - Page left

$88 - Home

$89 - End

$8A - Delete character under cursor

$8B - Delete previous character

$8C - Clear screen key

$8D - Function key 1

$8E - Function key 2

$8F - Function key 3

$90 - Function key 4

$91 - Function key 5

$92 - Function key 6

$93 - Function key 7

$94 - Function key 8

$95 - Function key 9

$96 - Function key 10

$97 - Function key 11

$98 - Function key 12

$99 - Function key 13

$9A - Function key 14

$9B - Function key 15

The TTY interface Page: 3

3. THE TTYPATCH UTILITY

The TTYPATCH provides a method of modifying an application program

which uses the "TTY" interface to customize it for a particular

terminal.

TTYPATCH does not itself use the TTY interface, but operates using

a line by line menu system. This allows configuration for a

particular console terminal when no defintions for that terminal

previously exist.

Once TTYPATCH is invoked, it will display the following menu:

1 - Standard key definitions

2 - Function key definitions

3 - Control code definitions

4 - Load settings from program

5 - Save settings to program

6 - Exit TTYPATCH utility

If you are defining a terminal for the first time, you may

immediatly begin entering key/code definitions using the options (1,

2, and 3) of the menu.

If you wish to modify the terminal interface of an existing

program, you should first select option (4), and enter the name of

the programs executable (".EXE") file at the "Filename?" prompt. This

will load the key and function code settings with the values which

are currently defined in that program.

Once modifications are complete, you should select option (5) to

write the new settings to the file. Enter the names programs

executable file at the "Filename?" prompt. If you wish to define the

same terminal interface for several programs, you may use option (5)

to write the settings to each one in turn.

Note that simply reading (Option 4), and then writing (Option 5)

to a different program file provides a method of copying the settings

from one program to another.

The TTY interface Page: 4

3.1 Entering definitions

The first three items of the main menu allow you to display and

change the key and control codes used by the terminal interface.

WARNING: Do to the the way the GETKEY function traverses the

table for the input keys (Options 1 and 2), it will stop

processing if it encounters a NULL key definition, and any keys

defined past that definition will be unavailable to the

application program. In other words, ALL standard and function

keys MUST be defined.

Option (1), "Standard key definitions" defines the following

keys which should be common to all screen oriented applications.

A - Cursor up

B - Cursor down

C - Cursor right

D - Cursor left

E - Page up

F - Page down

G - Page right

H - Page left

I - Home

J - End

K - Delete character

L - Delete previous

M - Clear screen

CURSOR UP, DOWN, LEFT, and RIGHT should be defined to the codes

generated by the "arrow" keys on the terminal.

PAGE UP, DOWN, LEFT and RIGHT are provided for movement between

screens and/or fields on the screen.

HOME and END are provided for rapid access to the first and

last data entry fields within an application.

DELETE CHARACTER is intended to represent a key which will

delete the character under the cursor without moving it.

DELETE PREVIOUS represents a key which will backup the cursor,

and then delete the character under it, in effect removing the

last character entered.

CLEAR SCREEN is used to refresh the terminal screen,

re-displaying all infromation.

The TTY interface Page: 5

Option (2), "Function key definitions" defines 15 function

keys, which are available for application dependant key

requirements.

Option (3), "Control code definitions" defines the output codes

which are sent to the terminal to perform specific functions.

A - Terminal initialization

B - Cursor positioning

C - Clear screen

D - Clear to end of line

E - Clear to end of screen

F - Special effect ON

G - Special effect OFF

H - Scroll screen forward

TERMINAL INITIALIZATION is a string which is sent to the

terminal when the application is first invoked. It is used to set

up function keys and or terminal modes.

CURSOR POSITIONING is the string which is sent to the terminal

to position the cursor.

CLEAR SCREEN is a string which is sent to clear the terminal

screen.

CLEAR TO END OF LINE is a string which is sent to clear from

the cursor position to the end of the line.

CLEAR TO END OF SCREEN is a string which is sent to clear from

the cursor position to the end of the screen.

SPECIAL EFFECT ON is a string which is sent to enable a special

video mode. REVERSE VIDEO is the preferred special mode if it is

available.

SPECIAL EFFECT OFF is a string which turns off the special

video mode and returns to normal output.

SCROLL SCREEN FORWARD is a string which causes the the terminal

to shift the display up one line. For most terminals this string

should position the cursor at the bottom left hand corner of the

display, and issue a LINE FEED character.

The TTY interface Page: 6

To enter a new value for a particular string from the sub-menu,

press its corresponding letter. Entering a CARRIAGE RETURN at the

sub-menu prompt will return you to the main menu.

When entering a string, pressing the CARRIAGE-RETURN key will

cause a prompt for an action to take, which is entered as a single

character. The available options are:

(D)elete - This option causes TTYPATCH to delete the last

character entered, re-display the input line, and continue input.

(C)ontinue - This option simply re-displays the input line and

continues input.

(E)nter-CR - This option causes the carriage return character

(^M) to be entered into the input line, and continues input.

(H)exchar - This option must be followed by two hexidecimal

characters (0-9, A-F), after which a character of that binary

value till be entered into the input line.

(Q)uit - This option terminates input, and returns to the

sub-menu, allowing you to pick another function.

(X) - During entry of CURSOR POSITIONING, use of this option

causes the desired 'X' (horizontal) position to be inserted at

this position in the output string. Following this selection

TTYPATCH prompts for (B)inary or (D)ecimal output. Selecting

Binary causes the position to be output as a single character with

the 'X' value. Selecting Decimal causes the position to be output

as a ASCII encoded decimal string. TTYPATCH then prompts for the

OFFSET value, entered two hexidecimal digits. The OFFSET value is

added to the 'X' value before it is output.

(Y) - During entry of CURSOR POSITIONING, use of this option

causes the desired 'Y' (vertical) position to be inserted at this

position in the output string. Following prompts are the same as

for (X).

The TTY interface Page: 7

For example, a VT100 terminal expects the 'X' and 'Y' positions

to be output as ASCII encoded decimal with a value beginning at

'1'. The format of the cursor positioning string is "<ESC>[Y;XH".

You would enter this string as: (Note <ESC> displays as '^[')

^[[ (Carriage return Y)

(B)inary (D)ecimal? D

Offset value (hex)? 01

^[[<YD01>; (Carriage return X)

(B)inary (D)ecimal? D

Offset value (hex)? 01

^[[<YD01>;<XD01>H (Carriage return Q)

A VC4404 Terminal expects 'X' and 'Y' position to be output as

printable characters with a value beginning with a space. The

format of the cursor positioning string is "^PYX". You would enter

this string as:

^P (Carriage return Y)

(B)inary (D)ecimal? B

Offset value (hex)? 20

^P<YB20> (Carriage return X)

(B)inary (D)ecimal? B

Offset value (hex)? 20

^P<YB20><XB20> (Carriage return Q)

The TTY interface Page: 8

4. SAMPLE DEFINITIONS

The [EXAMPLES] directory of your distribution diskette contains

sample defintions for a VT100 and VC4404 terminals which are placed

in small "STUB" files called "VT100.TTY" and "VC4404.TTY".

These samples may be examined or copied to a screen oriented

application (such as the "EDT" editor) using "TTYPATCH".

In both defintions, the cursor keys are mapped to the terminal

arrow keys. Other key definitions are as follows:

4.1 VT100

The VT100 defintion makes use of the alternate keypad mode of

the terminal, and defines the keys a follows:

"TTY Definition" "EDT Editor usage"

+------+------+------+------+ +------+------+------+------+

| Home | End | F9 | F8 | |Top of|End of|Delete|Delete|

| | | | | | File | File |to end| Line |

+------+------+------+------+ +------+------+------+------+

| Page | Page | F6 | F10 | | Page | Page |Cursor| Un- |

| Up | Down | | | | Up | Down |to top|Delete|

+------+------+------+------+ +------+------+------+------+

| Page | Page | F4 | F2 | |Top of|End of|Toggle| Last |

| Left | Right| | | | line | line | EOL | Cmd |

+------+------+------+------+ +------+------+------+------+

| F12 | F11 | F5 | | | Word | Word | Show | |

| | | | | | Left | Right|Cursor| Exec |

+------+------+------+ F1 + +------+------+------+ +

| F7 |Clear | | | Tag |Redraw| Cmd |

| |Screen| | | Line(s) |Screen| |

+------+------+------+------+ +------+------+------+------+

Delete character ---- DELETE ---- Delete character

Delete previous ---- BACKSPACE ---- Delete previous

F3 ---- LINE-FEED ---- Toggle INSERT

F13 ---- <ESC> 1 ---- Not used

F14 ---- <ESC> 2 ---- Not used

F15 ---- <ESC> 3 ---- Not used

The TTY interface Page: 9

4.2 VC4404

The VC4404 defintion makes use of the function keys of the

terminal, and defines the keys as follows:

"TTY Definitions"

PF1 PF2 PF3 PF4 PF5 PF6 PF7 PF8 PF9 PF10

+------+------+------+------+------+------+------+------+------+------+

| F3 | F5 |Del Ch| F8 | F7 | | F12 | F11 | Page | Page |

| \*F4 | \*F6 | \*F10 | \*F9 | | | | | Up | Down |

+------+------+------+------+------+------+------+------+------+------+

Note: Keys marked with '\*' are accessed by pressing <ESC> before

pressing the indicated PF key.

Page Right ---- <ESC> Right arrow

Page Left ---- <ESC> Left arrow

Home ---- <ESC> Up arrow

End ---- <ESC> Down arrow

F1 ---- Home key

F2 ---- <ESC> Home key

F13 ---- <ESC> 1

F14 ---- <ESC> 2

F15 ---- <ESC> 3