Serial Master

Overview

The Serial_Master application is an Autohotkey program that manages 'input from' and 'output to' a serial port. The program combines several novel features that are intended to ease integration of serial I/O with other applications. When started, a timed Splash screen is displayed alerting the user that the program is running in the background and connects a Partner Application to serial I/O. The Partner Application to receive Serial Input, as well as other controlling parameters may be set within the **Program Options** window.

Installation, Setup and Start-up

Serial_Master.zip is a compressed folder containing all files needed to install and operate Serial Master. The zip file contains the following documentation files:

Serial_Master_readme.txt -- is a plain text file that is substantially the same as this section.
Serial Master documentation.PDF – This file in pdf form
Com_Port_Services_V2.ahk -- Autohotkey source code supporting ComPort functions.
GIDEI_2_Decoder_v2.ahk -- Autohotkey source code for the GIDEI decoder.
Serial_Master_V2_Splash.ahk -- Autohotkey source code for the timed Splash and 'About' displays.
Serial_Master_Menu_V2.ahk -- Autohotkey source code for the program configuration menu.
Serial_Master.ahk – Top level Autohotkey source code for start-up and operation of the above sub-systems.

The zip file contains the following files required for operation:

Green_gear.png -- Image file of the tool tray icon.
Seed.xlsx – An otherwise empty EXCEL file used when a new EXCEL file name is opened.
Serial_Master.exe – The complete, compiled, executable Serial Master program.
COM_Port.cfg – A plain text configuration file that Com_Port_Services reads and writes.
Serial_Master.cfg – A plain text configuration file that Serial_Master_Menu reads and writes.

Installation and set-up consist of un-zipping Serial_Master.zip into a folder of your choice. Files required for operation must remain together in the same folder. You may create short-cuts to Serial_Master.exe and move the shortcut to the desktop for convenience or to the START Folder of your computer to auto start when you turn on the computer. You may also choose to use Windows Explorer to locate Serial_Master.exe double click the program name. Note that if the Serial Com Port number saved in Serial_Master.cfg is not available for use, Serial Master will open the COM Port configuration Menu, notifying the user and wait for selection of an available serial Com Port.

Program Options window

Configuration parameters are retrieved from Serial_Master.cfg file and activated when Serial_Master starts. The Program Options window is opened by left clicking the Green Gear icon located in the Tool Tray and selecting the **Program Options** menu item. See Figure 1 for a view of the Program Options Menu. Clicking the Cancel button will exit the menu without saving any changes. Configuration parameters may be changed, saved and activated when the Program Options window is closed by clicking the OK button in the upper right portion of the window. Serial Master associates operating parameters as an independent profile for each Partner Program. Selecting a Partner Program populates the remaining parameters with the last values saved for this Partner Program. Serial_Master.cfg is saved in the same directory as the Serial_Master program.

Within the Program options menu the user may adjust the following controls:

1) **Partner Application**. Selects a program to receive Serial Input and produce Serial output.

a. When a Partner Program is identified, Serial Receive Data will make the Partner Program active/in focus, thereby directing Serial Input only to the Partner Program. When the partner program is active/in focus keyboard data *may* be sent to the serial port.
b. When 'None' is selected, serial input is sent to the active/ in focus program and any keyboard data *may* be sent to the serial port. No Companion File is opened.

2) **Start Options** . Chooses the way the Partner Program is started and interacts with a selected Companion File. Serial Master will always wait for the designated Partner program to start before any other service is provided. Note that any Partner Program opened is NOT closed when Serial_Master exits!

Start options:

a. Launch. If the Partner Application is already running, Serial_Master will activate the most recently active Partner Application instance, along with any Companion File open of that instance. If the Partner Application is not already running, Serial_Master will open the Partner Application with no companion file.
b. Launch/Open File. Serial_Master will launch/start a Partner Program and open a Companion File associated with the Partner Program. (default) If the Companion File does not exist, one will be created in the same directory that holds Serial Master. When Serial Master recognizes that the Companion File_name is open in any other program, the user is warned and prompted to act. Note that any file opened or created by this option is NOT closed when Serial_Master exits!
c. Wait, User Start. If the Partner program is already open, it will be made active/in focus and Serial Master services available. If the Partner Program is not open, a Message Box will open signaling the user that Serial Master is waiting for the user to start the Partner Application (and companion file if desired). The user must click 'OK' when the program is open and ready for operation with Serial Master.
Tip: When a Partner Program is activated with a Companion File, the file cursor position is directed to file end.

3) **Keyboard Input**. Chooses where keyboard input is directed.
a. Off – Keyboard input is not sent to the serial port, for Serial "receive only" operation. Keyboard input is sent to the Application that is 'in focus'.
b. Serial -- Keyboard input is sent to the Serial port only. This mode is useful where the attached serial port echoes received data.
c. Both – (default) All keyboard input is sent to the serial port and to any 'in focus' application. Key data is sent to the present cursor location.
d. Force – The Partner Program is made Active/In focus and keyboard input is sent to the Partner Program.
Tip: Holding the Control key and pressing Enter key sends a Space to the application and CR/LF to the serial port. This allows one to type a command to the device attached to the Serial port and receive the device response on the same display line of the application.
Note: Key codes sent to the serial port are limited to QWERTY keyboard keys, their 'shift', and the control keys Enter, Ctrl-Enter, Backspace, Escape and Tab.

4) **File Name**.
a. A Check Box, when selected, will add the 'COM:n', to the filename of the Companion File. Where *n* is replaced by the open comport number,
b. A drop-down list of common file names is provided. Selection of the 'none' option will force Start Options to default to Launch.

c. A drop-down list of file extensions is provided. Selection of the 'none' option will force Start Options to default to Launch.

5) **GIDEI Decoder**. Choose if Serial Input data is interpreted by a GIDEI escape sequence decoder.
   a. On +Vol (default) - GIDEI escape sequence decoder is active. Volume_Mute, Volume_Down, and Volume_Up key names are included in the list of recognized keys. See the GIDEI Command Decoder description for more information.
   b. On - GIDEI escape sequence decoder is active with a 'standard' list of key names.
   c. Off - all serial data characters are sent to the Application program without escape sequence interpretation by Serial_Master

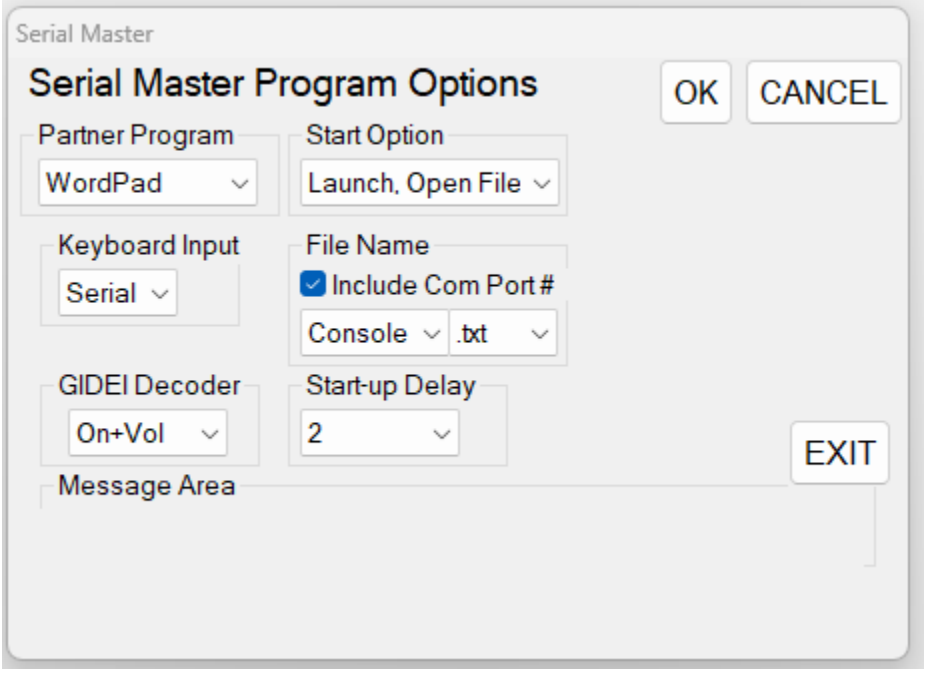

Figure 1

6) **Start-Up Delay.** This parameter is the maximum time that Serial Master will wait for the Partner Program to become active prior to Processing serial input data. If the Partner Program becomes active any time before the timer expires, serial input data processing immediately resumes. For most systems a 2 Second delay is adequate. If Serial Master (or a shortcut to Serial Master) is included in a System Start Folder, Partner Program startup may be substantially longer and some experimentation with this parameter may be necessary.
   The serial data input process demands that the Partner Program is active. If the Start-up Delay timer expires and the Partner Program has not yet become active, Serial master will open the Program Options Menu with an error message in the message box.
7) View or edit User Alias list used by the GIDEI Decoder. (Not implemented.)
   a. Enter the base Alias word (or letter)
   b. Select included Modifiers.
   c. Select System Key Name or letter

COM Port Configuration menu

The COM Port configuration Menu is opened by left clicking the Green Gear icon located in the Tool Tray and selecting the COM Port configuration menu item. See Figure 2 for an example of the COM Port

configuration menu. Configuration parameters are retrieved and made active when Serial_Master starts or when the COMPort configuration window is closed by clicking the OK button. A change of the Port number will close the initial Serial Port and open the new Serial Port when the COM Port configuration window is Closed by clicking the OK button. Click the Cancel button to close COM Port configuration without changes. COM_Port.cfg must be in the same directory as the Serial_Master program.

Within the COM Port configuration window the user may adjust the following controls, all of which are selected from a drop-down list:

1) Ports — Choose from COM1 thru COM8
2) Bit Rate — Choose data rate 110bps to 115200bps.
3) Data bits — Choose 7 or 8 bits/character.
4) Parity bit — Choose ODD, EVEN or None.
5) Stop Bits — Choose 1 or 2 stop bits
6) Tx Flow Control — Choose XON/XOFF, CTS, DSR, ALL or None.
    a. Choosing All activates XON/XOFF, CTS and DSR Tx flow control.
    b. XON/XOFF Stops transmission following reception of the XOFF control character until the XON control character is received. Note that Receive XON/XOFF flow control is also active when Tx Flow Control is active.
    c. None allows character transmission independent of interface control lead or XON/XOFF state.
    d. CTS and DSR selections require that the respective interface control lead must be 'On' for character transmission.
7) Rx Flow Control — Choose XON/XOFF, RTR/RTS, DTR, DSR or None
    a. RTR/RTS – The RTS interface pin assumes the Ready to Receive (RTR) functionality. The RTS/RTR interface signal will be 'On' to signal Receive buffer space is available for character reception. When selected, this setting will 'gray out' the RTS Control. RTR/RTS is not supported by many drivers.
    b. DTR -- The DTR interface signal must be 'On' to allow character reception. When selected, this setting will 'grey out' the DTR Control. This setting is not supported by many drivers.
    c. XON/XOFF – XOFF signal is transmitted when the receive buffer space is less than 512 bytes. XON signal is sent when the receive buffer occupancy has been reduced to less than 128 bytes. The default receive buffer is 4096 bytes. The system is configured to suspend data transmission from when XOFF is sent, until XON is transmitted.

    Note: Closing Com Port Configuration with any configuration setting that is not supported by the system Serial Port Driver, will re-open Com Port Configuration menu with notification in the Message Area.

The Com Port Configuration menu provides a real time display of Serial Port Input control signals. A **Green** indicator signals the control is 'On', **Red** signals the control is 'Off' and **Gray** indicates the port is not yet open.

Serial Master services continue independently while the Com Port Configuration menu is displayed.

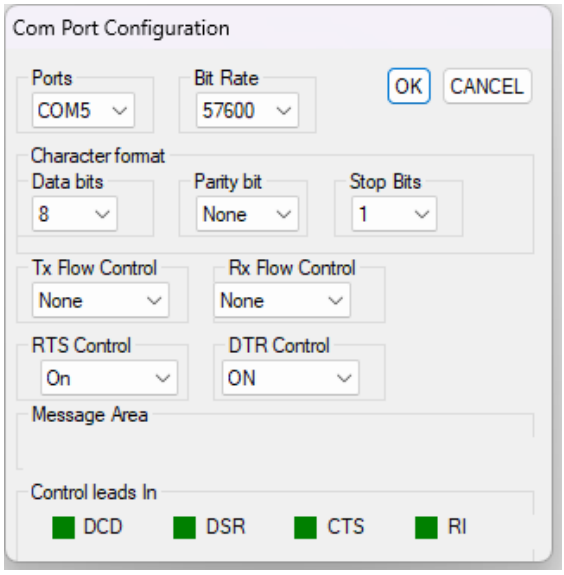

Figure 2

Computer Sleep, Hibernate and Screen Savers

Power Plan setting of your computer may adversely effect Serial Master performance.

- **Hibernation:** Operation of Serial Master will temporarily cease while the computer hibernates. Exiting from hibernation requires activation of the computer power button where upon Serial Master will resume operation after the system wakes up. To check or change hibernate settings open Control Panel and navigate to: Control Panel\All Control Panel Items\Power Options\Edit Plan Settings and select 'Change Advanced power settings' to open the Power Options Menu. Within the menu choose a Power Plan to customize and click on 'Sleep'. Then click on the hibernate sub item, if you wish to change the setting click on the 'setting' line to change the system idle minutes to wait before entering hibernate. Enter 'Never' to prevent hibernation for this power plan. This hibernate setting applies to this power plan only.
- **Sleep:** Serial Master operation is suspended during system sleep. Your serial port may be able to wake the system when serial data is received. To determine if your serial port adapter and driver can wake the system, open the 'Device Manager' program. Device Manager will display a list of system resources, first expand 'Ports(Com & PLT)' and right click the Com Port that Serial Master uses and click 'Properties'. If the properties window has a 'Power Management' tab, open the tab and enable wake from suspend/sleep. If there is no Power Management tab on this serial port and the adapter is a USB serial adapter, expand the 'Universal Serial Bus controllers' resource line. Look for a USB Serial controller or similar resource. Right click on this resource, then click on 'Properties'. If the properties window has a 'Power Management' tab, open the tab and enable wake from suspend/sleep. Failure to locate a power management tab probably indicates the serial port lacks the ability to wake the computer from sleep. Sleep may prevent un-attended operation of Serial Master. If the serial port lacks the ability to wake the system preventing sleep in the power plan, similar to the prevention of hibernation, may be an option.
- **Screen Savers**: Serial Master is fully functional while standard screen savers are active. Serial port activity will first close a standard screen saver, put the Partner Program in focus and deliver serial data. Screen Saver operation will resume after the prescribed time of system idle.

GIDEI Command Decoder

The General Input Device Emulating Interface (GIDEI) Command decoder is *emulation interface software* that runs on a computer with the Windows operating system. The decoder interprets Escape Sequences and provides services defined in the GIDEI 2.1 Standard. This standard should be consulted for a more complete understanding of this program's operating characteristics. In particular, the "List of Currently defined Key Names" and words in italics have meaning to this decoder. The Serial Master program makes every effort to comply with the requirements of the standard while allowing functional flexibility not prohibited. The goal is to provide 'out of the box' operation with serial communication connected devices that may loosely follow the Standard.

An Escape Sequence is defined as the set of character codes, received from the serial interface, starting with an *escape* character code (0x1B), containing one or more *fields,* and ending with a *period* character code (0x2E). A *Field* is delimited by the escape character code, a comma character code (0x2C), or the ending period character code. This decoder interprets Escape Sequences using the following rules and procedures:

a) Notationally, brackets [ ] denote optional *fields*, literal expressions are enclosed a ' ', and the escape character code is replaced by the notation '<esc>'.
b) All space character codes (0x20) are removed from within the escape sequence. Where an escape sequence must produce a space, include the *Key name* 'space' as a *field* in the sequence.
c) Character codes A-Z (0x41 – 0x5A) within the escape sequence are converted to lower case equivalents.
d) Comma character codes, within the escape sequence, are used to separate *fields.* Where an escape sequence must produce a comma, include the Key name 'comma' as a *field* in the sequence.
e) The escape sequence takes the form: '<esc>[*,field1,field2,fieldn*].'. Brackets indicate optional *fields*. '<esc>.' alone is silently ignored.
f) The GIDEI Standard defines the syntax <esc>*,field1*,… as a command field. Command fields effect how subsequent fields are interpreted.
g) The GIDEI 2.1 Standard contains a "*List of Currently defined Key Names*". This implementation of the standard divides entries in the list into four categories;
    a. 'Native' Key names – *Key Names* that are interpreted as real or virtual keyboard or mouse functions recognized by the Windows Operating System. Includes System keys, Cursor and Edit keys, State keys, Function keys, Multimedia keys and Keypad keys. See Appendix A
    b. Modifier Keys – as discussed below.
    c. Character names, common symbols, diacritical marks and keys that may be omitted from special input devices. This group of *Key Names* is further discussed below. See Appendix B.
    d. Aliases – Are *Key Names* recognized by this decoder as having the same function as a *Key name* in a. ,b. or c. above. See Appendix C
    e. Unknown -- Some *Key Names* in the "*List of Currently defined Key Names*" have no known or equivalent function in the Windows Operating System. These *Key Names* may cause an escape sequence to fail or be silently ignored.
h) A GIDEI escape sequence lacking a Command field is referred to as an "*Implied Press*" and typically has the form <ESC>*field.* and the field is a *Key Name* or alias. This decoder allows modifier fields in an *Implied Press*. An *implied press* may also consist entirely of modifiers.
i) Unless a specific Command interprets subsequent *fields* as *Key Name*(s), *shift*, *ctrl*, *control*, *alt*, and 'win' are considered key modifiers. Note that Key modifiers are only produced by escape sequences and effect how the computer operating system interprets a keystroke. Modifiers are prepended to an immediately following serial character code or <esc> sequence that produces a keystroke.
j) Key modifiers that are not applied within an escape sequence are cumulative except that any preceding duplicate is deleted from the group. '<esc>ctrl. <esc>shift. v' is equivalent to '<esc>ctrl,shift,v.'. Similarly, <esc>shift. <esc>shift.c is equivalent to <esc>shift,c. Modifiers are deleted after they are applied to a character code or escape *field* that produces a keystroke or mouse movement.
k) Serial character codes (not in an escape sequence) that do not have a prepended modifier are sent to the operating system as literal character codes. This includes all 256 extended ASCII character codes without

consideration to how the operating system responds to control character codes or other non-printing Character codes. The effect of ASCII character codes 128 to 255 depends on the System Code Point the computer has in effect. See Appendix D for extended ASCII character sets.

l) Within GIDEI 2.1 Standard "*List of Currently defined Key Names*", several are interpreted as diacritical marks, characters with diacritical marks, or symbols that result in keystrokes. All have no code assignment in 7-bit ASCII. These *key names* are recognized by the decoder and sent to the operating system as Unicode character codes, allowing recognition of these character symbols without regard to the System Code Point the computer has in effect. The shift modifier, or a locked shift key (see the lock command), shall designate the capital letter keystroke and send the appropriate Unicode character code. Any other modifier will cause the *Key Name* to be unrecognized. See Appendix B for a table of key names processed in this manner. The *implied press* form for these *Key Names* is '<esc>[shift,]*Key Name'.* See the <esc>,*hold*, command for additional examples of how these key names may be composed.

m) The decoder allows a Unicode character code to be sent as a keystroke to the operating system with an escape sequence of the form '<esc>u+NNNN.' where NNNN is any 4-digit hexadecimal number. Any prepended modifiers will cause this sequence to fail.

n) The decoder treats many of the *Currently defined Key Names* to be equivalent to another *Keyname*, a character code, character code with modifiers (aka shortcut), or a duplicate name for a control key press. Therefore, the Serial Master hosts an **alias list** to map parameter field content to decoder recognized content. Command fields are never aliased. Prior to individual command processing, each parameter field's content is searched for a match in the alias list. If a match is found, the field content is replaced with a corresponding parameter in the alias list. See appendix X for *Currently defined Key Names* in the alias list and the keycodes that replace them.

Escape Command Sequences

The GIDEI Standard defines <esc>,*field*,… as an Escape Command. Escape command fields effect how subsequent fields are interpreted. The following command descriptions and examples are intended to supplement descriptions contained in the GIDEI 2.1 Standard. These descriptions may include operations omitted from the Standard but not expressly excluded by it. *Fields* included in [ ] are optional parameters. The decoder may reject some commands and open a small, timed message box and requesting the computer to sound a tone. The message box signals a failure condition, the escape sequence in notational form and where possible a hint indicating the reason for rejecting the escape sequence.

Tip: The message box failure notification prevents the potential havoc that can ensue if the failed escape sequence is passed to the operating system in literal form. This is particularly true if control keys are locked at the time of failed sequence arrival.

- Lock command: syntax'<esc>,lock,*Key Name* 1[,*Key Name* 2,*Key Name* 3,*Key Name* n].' .
  All *fields* are treated as a *Key Name.* The specified *Key Name*, or single character code, will be pressed for ALL subsequent keystrokes sent to the operating system until the Key *Name* (s) are released by the <esc>,rel… command. The lock command clears any pending modifiers.
  This decoder allows the lock condition to be set for ANY key name, whether or not it actually exists, without consideration of how the operating system will respond. The program maintains a list of locked keys. Lock commands with duplicate or already locked key name are allowed but only one instance of the name is recorded in the list and any key will be Locked only once.

- Rel command: syntax '<esc>,rel[,*Key Name* 1,*Key Name* 2,*Key Name* 3,*Key Name* n].'
  All *fields* are treated as a *Key Name*, or single character code, to be released for subsequent keystrokes sent to the operating system and removed from the locked key list. A release command with no key fields ('<esc>,rel.') will clear the locked key list and release ALL keys locked in the operating system, including mouse buttons. The rel command clears any pending modifiers. Note that any *field* containing an invalid key name will be silently discarded by the command and will not affect command completion. If the GIDEI decoder is active, Serial Master will execute an <esc>,rel. command prior to exit to prevent any residual locked keys in the operating system.

Tip: The Rel command with no key Name also releases CapsLock key state. NumLock and ScrollLock state will toggle when their respective key name is 'pressed'.

- Combine command: syntax '<esc>,combine[,*field1,field2,field*n].'
  Fields containing *Key Name*(s), *shift*, *ctrl*, *control*, *alt*, and 'win' are key recognized as modifiers. Remaining *field(s)* are concatenated. The Combine command is then evaluated as single *Key_Name* with prepended modifiers. A common use of this command is to send short-cuts to the Partner Application and as such are application dependent. Short-cuts can usually also be sent as *implied commands*.
  Note: The standard states that *fields* contain *Key_Names* that should be pressed in order, then released in reverse order. Strict adherence to this requirement can produce unpredictable results since *Key_Names* include character names and aliases unrecognized by the operating system. By processing as described, single letter Short-cuts, system keys (with modifiers), and character names (including with shift) are all properly recognized. Additionally, 'composed' combinations of numbers (including their alias representation) letters, symbols, character names, and diacritical marks will be accepted.

- Hold command: syntax '<esc>,hold[,*field1,field2,field*n].'
  The hold command will create or change an existing modifier for evaluation by the next <u>following character code</u> or escape sequence. hold command content will not cause a keystroke.
  Fields containing *Key Name*(s), *shift*, *ctrl*, *control*, *alt*, and 'win' are key modifiers. Any remaining *field*(s) are appended literally to an existing modifier or create a new modifier.
  For example: '<esc>,hold,*shift,*a.<esc>,acute.' Will be interpreted as '<esc> *shift,*aacute.' Since aacute is recognized as a defined *key name* that allows the shift modifier, the sequence will produce Á. Note that due to the fact *hold* fields appended to a modifier (or create new) and modifier key names always prepended to (or create) a modifier, the following sequences are equivalent:
  '<esc>,hold,*shift,*a.<esc>,acute.'
  '<esc>,hold,a.<esc> *shift.*<esc>,acute.'
  '<esc>,hold,a.<esc>shift,acute.'
  '<esc>,hold,a.<esc>acute,shift.'
  '<esc> *shift.*<esc>,hold,a.<esc>acute.' All produce Á.

- Anchor command syntax '<esc>,anchor[, *field1*].'
  Anchor command allows the user to designate the current mouse cursor location as a set point to later return the mouse cursor by the goto command. Field1 is a single a-z label that uniquely identifies the anchor point. The GIDEI 2.1 Standard does not identify *Field1* as an optional field but rather requires a single a-z character code to immediately follow the escape sequence to label the anchor. This decoder allows either command format. The GIDEI 2.1 Standard allows a duplicate anchor label to reset an anchor point. This decoder will replace a duplicate anchor label with new coordinates. Modifiers are cleared by this command.

- Goto command syntax '<esc>,goto [,*field1,field2*].'
  The goto command with no additional fields requires a single a-z character code immediately follow the escape sequence to identify the anchor and complete the command.
  This decoder also allows syntax'<esc>,goto,*field1*.' Where field1 must be a single a-z character identifying the anchor that allows the command to complete.
  Syntax '<esc>,goto,*field1,field2*.' Where field1 and field2 are integers >=0 move the cursor to absolute screen X, Y location. Goto stops any mouse movement initiated by a mougo command and completes with no modifiers. Commands that fail to meet the requirements of one of the allowed forms will be rejected.

- Moulock Command syntax '<esc>,moulock[, *field1,field2,field3,field4,field5*].'
  This command will hold the mouse buttons down until released by the mourel or moureset command. The GIDEI 2.1 Standard allows zero to five parameter fields for this command. When no field is supplied, the decoder will 'look-up' "butdefault" in the alias list and lock an allowed button name. If no allowed

button is found, no button will be locked. The GIDEI 2.1 Standard allowed field content as “left”,”right” or “but1” through “but5, of these only “right” appears in the *List of Currently Defined Key Names*. This decoder always recognizes “left” and “right” as decoder recognized LButton and Rbutton respectively. The decoder uses the alias list to make the following (editable) assignments to decoder (and decoder) recognized mouse button names:

Butdefault => LButton	assigns the left mouse button as the default
but1 => LButton	the left mouse button
but2 => RButton	the right mouse button
but3 => MButton	the middle mouse button
but4 => XButton1	the fourth mouse button
but5 => XButton2	the fifth mouse button

All mouse button functions are system configuration dependent. Fields that have anything other than the above assignments will be silently ignored. Duplicate fields will result in only one mouse button press. Modifiers are cleared by this command.

- Mourel command syntax ‘<esc>,mourel[, *field1,field2,field3,field4,field5*].’
  The GIDEI 2.1 Standard allows zero to five parameter fields for this command. When no *field* is supplied, the decoder will release all mouse buttons. The decoder depends on the (editable) alias list to recognize mouse button names allowed by the standard. (See Moulock command above) Any recognized field(s) will be unlocked without regard to the order presented.

- Mougo Command syntax ‘<esc>,mougo, *field1,field2.*’
  Mougo will first stop any movement from an ongoing mougo command. Then starts the cursor moving in the direction specified in *field1* at the Speed specified in *field2*. Movement is relative to the current mouse cursor position. Movement stops when the Moustop command is received, the desktop edge is reached, a mouse (dbl)click received or the moureset command is received. Recognized direction specifiers are up, down, left, right, upleft, upright, downleft and downright. Allowed speed are integers 1 (slowest) to 10 (fastest). Mouse speed 1 moves the mouse cursor at a slow rate where the full vertical space of the desktop will be traversed in approximately 28 Seconds. Mouse speed 10 will traverse the same space in approximately 10 seconds. Increasing the mouse speed by 1 unit will reduce the full vertical travel time by approximately 2 seconds. The relatively slow mouse movement is to allow the mouse cursor to be stopped where desired. The decoder treats mouse cursor movement started by this command as a background process and is fully capable of processing any other escape sequence as the cursor moves. Any deviation from the above requirements will result in command rejection. Modifiers are cleared by this command.

- Moustop command syntax ‘<esc>,moustop.’
  Moustop ends any mouse movement initiated by a mougo command. Any additional fields included with moustop are silently ignored. Modifiers are cleared by this command.

- Moureset command syntax ‘<esc>,moureset.’
  Moureset stops any mouse movement started by mougo, sets the mouse cursor to the upper left corner of the partner program window, clears all mouse anchor points, and releases any locked mouse buttons. Any additional fields included with moureset are silently ignored. Modifiers are cleared by this command.

- Move command syntax ‘<esc>,move, *field1,field2.*’
  The move command stops any mouse movement started by mougo and then moves the mouse cursor *field1* pixels -left/+right and *field2* moves -up/+down relative to the initial mouse cursor position. The command clears modifiers and will fail if *field1* and *field2* are not integers.

- Click command syntax ‘<esc>,click[, *field1,field2,field3,fieldn*].’
  The click command stops any mouse movement started by mougo and sends a mouse button click to the system. The click command allows modifier fields. After processing modifier fields, any remaining field, if

present, and a recognized mouse button, will be clicked. A field containing 'left' or 'right' is also accepted as their respective mouse buttons. If a recognized mouse button is absent, the alias list is queried for a butdefault alias and if a mouse button is returned, clicked. The command clears modifiers on completion, and any additional fields are silently ignored.

- Dblclick command syntax '<esc>,dblclick[, *field1,field2,field3,fieldn*].'
  The Dblclick command stops any mouse movement started by mougo and sends a mouse button double click to the system. The dblclick command allows modifier fields. After processing modifier fields, any remaining field, if present, and a recognized mouse button, will be clicked. A field containing 'left' or 'right' is also accepted as their respective mouse buttons. If a recognized mouse button is absent, the alias list is queried for a butdefault alias and if a mouse button is returned, clicked. The command clears modifiers on completion, and any additional fields are silently ignored.

- Baudrate command syntax '<esc>,baudrate, *field1.*'
  The baudrate command is not implemented in a GIDEI 2.1 Standard way. Rather, the command opens the Com Port Configuration GUI with a message requesting the Bit Rate of *field1* allowing the user to make the data rate change using the configuration GUI, if needed. The content of *field1* is not verified in any way. The standard allows 300, 1200, 2400, 4800, 9600 and 19200 bits per second rates. Serial Master also allows 110, 600, 38400, 57600 and 115200 bits per second.

Note: Serial Master does not implement the Baud Rate Matching protocol suggested in the Miscellaneous Commands paragraph. The proposed protocol, where the external device sends three or more null character codes (0x00) at 300 baud to the decoder serial port, will cause (up to three consecutive) Framing Errors if the port receive bit rate is greater than 300 Baud. Three consecutive Framing Errors then cause the decoder to: 1) switch to 300 Baud, 2) wait for CTS (or possibly Xon character code), 3) send Xon to the external device in response to a received null character code.

Serial Master allows Flow Control settings that conflict with the GIDEI 2.1 Standard *Initial Connection or Baud Rate Matching* protocols. Serial Master will however, respond to a received null character code and send an Xon character code.

Caution, the serial port option that allows/prohibits the port to pass a received null character code may not be configurable for all port driver software.

Appendix A Key_List

alt	f1	f13	home	pause	shift	Numpad3	NumpadDown
appskey	f2	f14.	ins	pgdn	Sleep	Numpad4	NumpadEnd
backspace	f3	f15	Insert	pgup	tab	Numpad5	NumpadEnter
BS	f4	f16	LAlt	printscreen	up	Numpad6	NumpadHome
capslock	f5	f17	LButton	ralt	WheelDown	Numpad7	NumpadIns
ctrl.	f6	f18	Lcontrol	RButton	WheelLeft	Numpad8	NumpadLeft
del	f7	f19	LCtrl	rcontrol	WheelRight	Numpad9	NumpadMult
delete	f8	f20	left	rctrl	WheelUp	NumpadAdd	NumpadPgDn
down	f9	f21	LShift	right	NumLock	NumpadClear	NumpadRight
end	f10	f22	LWin	rshift	Numpad0	NumpadDel	NumpadSub
enter	f11	f23	MButton	RWin	Numpad1	NumpadDiv	NumpadUp
esc	f12	f24	numlock	scrolllock	Numpad2	NumpadDot	
escape							

Appendix B --- Named characters and symbols recognized by Serial Master

GIDIE Standard 2.1 Characters, Diacritical marks and Symbols

Characters	Lower case	Upper case		Characters	Lower case	Upper case		Diacritical marks	
aacute	á	Á		oacute	ó	Ó		acute	´
acircumflex	â	Â		ocircumflex	ô	Ô		cedilla	¸
adieresis	ä	Ä		odieresis	ö	Ö		circumflex	ˆ
ae	æ	Æ		oe	œ	Œ		dieresis	¨
agrave	à	À		ogonek	ǫ	Ǫ		grave	`
aogonek	ą	Ą		ograve	ò	Ò		ring	°
aring	å	Å		ohungarumlaut	ő	Ő		tilde	~
cacute	ć	Ć		ooblique	ø	Ø			
ccaron	č	Č		otilde	õ	Õ		**Other Symbols**	
ccedilla	ç	Ç		rcaron	ř	Ř		bbar	¦
eacute	é	É		sacute	ś	Ś		divide	÷
ecaron	ě	Ě		scaron	š	Š		exclaimdown	¡
ecircumflex	ê	Ê		tcaron	ť	Ť		micro	µ
edieresis	ë	Ë		uacute	ú	Ú		mordinal	º
egrave	è	È		ucircumflex	û	Û		multiply	×
eth	ð	Ð		udieresis	ü	Ü		onehalf	½
iacute	í	Í		ugrave	ù	Ù		onequarter	¼
icaron	ǐ	Ǐ		uhungarumlaut	ű	Ű		period	.
icircumflex	î	Î		uring	ů	Ů		section	§
idieresis	ï	Ï		yacute	ý	Ý		superone	¹
igrave	ì	Ì		ydieresis	ÿ	Ÿ		superthree	³
ncaron	ň	Ň		zcaron	ž	Ž		supertwo	²
ntilde	ñ	Ñ		zdotaccent	ż	Ż		yen	¥
				sharps	ß				
				comma	,	<			

Tip: Named symbols, diacritical marks or letter with diacritical mark can be selected by implied press.
Examples: <esc>tcaron. Or <esc>shift,tcaron.

Appendix C – Built-in alias substitutions

Alias	Replaced By		comments		Alias	Replaced By		comments
altgr	Ctrl,alt		Alt-graphic key		next	pgup		
amp	&				nine	9		
ampersand	&				number	NumLock		
apostrophe	'				numlk	NumLock		
ast	*				one	1		
asterisk	*				pagedown	pgdn		
at	@				pageup	pgup		
backslash	\				paste	ctrl,v		Short-cut
break	Pause				plus	u+002B		Special case +
bslash	\				pound	u+0023		Special case #
but1	LButton		Not included in GIDEI standard		prev	pgup		
but2	Mbutton		Not included in GIDEI standard		print	ctrl,p		Short-cut
but3	Rbutton		Not included in GIDEI standard		prtscr	PrintScreen		
but4	XButton1		Not included in GIDEI standard		rbrace	}		
but5	XButton2		Not included in GIDEI standard		rbracket	]		
butdefault	LButton		Not included in GIDEI standard		rcompose			For a PC the 'RWin' key press
bspace	Backspace		Backspace key press		remove	del		
cancel	esc				ret	enter		
capslk	Capslock		Capslock key press		return	enter		
clear	delete				rightwinkey	RWin		
colon	:				rolldown	WheelDown		scroll down
copy	ctrl,C		Short-cut		rollup	Wheelup		scroll up
cut	ctrl,x		Short-cut		rparen	)		
dblquote	u+0022		Special case "		semicolon	;		
dn	down		Down arrow key alias		seven	7		
dollar	$		Dollar sign		shiftleft	Shift		
eight	8				shiftright	Shift		
equal	u+003D		Special case =		six	6		
exclaim	u+0021		Special case !		slash	/		
five	5				space			
four	4				three	3		
help	F1				two	2		
hyphen	-				underscore	_		
kpslash	NumpadDiv				undo	ctrl,Z		Short-cut
menu	#				zero	0		
minus	-				front	-		ambigous, Alt-Tab to cycle through open windows

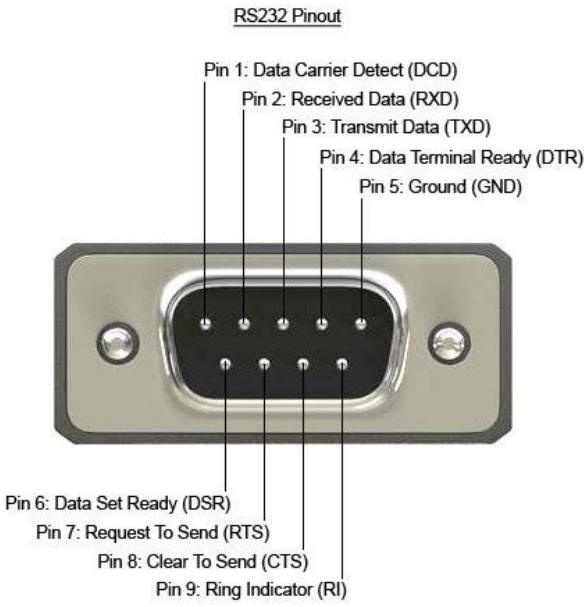

Figure 3 DE-9M Connector

Appendix D

ASCII control characters (character code 0-31)

The first 32 characters in the ASCII-table are unprintable control codes and are used to control peripherals such as printers. The Unicode assignments are non-printing codes and correspond to ISO-8859-1

Symbol	Unicode	Binary	Hex	CP437 or CP1252	Description
NUL	U+0000	00000000	0	0	Null character
SOH	U+0001	00000001	1	1	Start of Heading
STX	U+0002	00000010	2	2	Start of Text
ETX	U+0003	00000011	3	3	End of Text
EOT	U+0004	00000100	4	4	End of Transmission
ENQ	U+0005	00000101	5	5	Enquiry
ACK	U+0006	00000110	6	6	Acknowledge
BEL	U+0007	00000111	7	7	Bell, Alert
BS	U+0008	00001000	8	8	Backspace
HT	U+0009	00001001	9	9	Horizontal Tab
LF	U+000A	00001010	0A	10	Line Feed
VT	U+000B	00001011	0B	11	Vertical Tabulation
FF	U+000C	00001100	0C	12	Form Feed
CR	U+000D	00001101	0D	13	Carriage Return
SO	U+000E	00001110	0E	14	Shift Out
SI	U+000F	00001111	0F	15	Shift In
DLE	U+0010	00010000	10	16	Data Link Escape
DC1	U+0011	00010001	11	17	Device Control One (XON)
DC2	U+0012	00010010	12	18	Device Control Two
DC3	U+0013	00010011	13	19	Device Control Three (XOFF)
DC4	U+0014	00010100	14	20	Device Control Four
NAK	U+0015	00010101	15	21	Negative Acknowledge
SYN	U+0016	00010110	16	22	Synchronous Idle
ETB	U+0017	00010111	17	23	End of Transmission Block
CAN	U+0018	00011000	18	24	Cancel
EM	U+0019	00011001	19	25	End of medium
SUB	U+001A	00011010	1A	26	Substitute
ESC	U+001B	00011011	1B	27	Escape
FS	U+001C	00011100	1C	28	File Separator
GS	U+001D	00011101	1D	29	Group Separator
RS	U+001E	00011110	1E	30	Record Separator
US	U+001F	00011111	1F	31	Unit Separator

Here are some useful tips for entering displayable characters that may not be readily typed on the keyboard.

1) Most windows applications allow keyboard entry of displayable code symbols from the **CP437** character set. Press-and-hold the alt key and enter the decimal value of the of the symbol on the numeric keypad (with Numlock on). Then release the alt key.
2) Most windows applications allow keyboard entry of displayable code symbols from the **CP1252** character set. Press-and-hold the alt key and enter 0 (zero) followed by the decimal value of the **CP1252** code symbol on the numeric keypad (with Numlock on). Then release the alt key.
3) Most windows applications allow keyboard entry of any displayable Unicode symbol. Type the 4 digit hexadecimal value of the symbol immediately followed by Alt-x. It may be necessary to precede the hex number with U+.
4) The Unicode symbol value for any character symbol in this document can be displayed by selecting the symbol and then press Alt-x

Appendix D (cont.) ASCII printable characters (character code 32-127)

Codes 32-127 are common for all the different variations of the ASCII table and correspond to ISO-8859-1 They are called printable characters, represent letters, digits, punctuation marks, and a few miscellaneous symbols. You will find almost every character on your keyboard. Character 127 represents the command DEL.

Symbol	Unicode	Binary	Hex	CP437 or CP1252	Description
SP	U+ 0020	00100000	20	32	Space
!	U+ 0021	00100001	21	33	Exclamation mark
“	U+ 0022	00100010	22	34	Double quotes (or speech marks)
#	U+ 0023	00100011	23	35	Number sign
$	U+ 0024	00100100	24	36	Dollar
%	U+ 0025	00100101	25	37	Per cent sign
&	U+ 0026	00100110	26	38	Ampersand
‘	U+ 0027	00100111	27	39	Single quote, apostrophe
(	U+ 0028	00101000	28	40	Open parenthesis (or open bracket)
)	U+ 0029	00101001	29	41	Close parenthesis (or close bracket)
*	U+ 002A	00101010	2A	42	Asterisk
+	U+ 002B	00101011	2B	43	Plus
,	U+ 002C	00101100	2C	44	Comma
-	U+ 002D	00101101	2D	45	Hyphen-minus
.	U+ 002E	00101110	2E	46	Period, dot or full stop
/	U+ 002F	00101111	2F	47	Slash or divide
0	U+ 0030	00110000	30	48	Zero
1	U+ 0031	00110001	31	49	One
2	U+ 0032	00110010	32	50	Two
3	U+ 0033	00110011	33	51	Three
4	U+ 0034	00110100	34	52	Four
5	U+ 0035	00110101	35	53	Five
6	U+ 0036	00110110	36	54	Six
7	U+ 0037	00110111	37	55	Seven
8	U+ 0038	00111000	38	56	Eight
9	U+ 0039	00111001	39	57	Nine
:	U+ 003A	00111010	3A	58	Colon
;	U+ 003B	00111011	3B	59	Semicolon
<	U+ 003C	00111100	3C	60	Less than (or open angled bracket)
=	U+ 003D	00111101	3D	61	Equals
>	U+ 003E	00111110	3E	62	Greater than (or close angled bracket)
?	U+ 003F	00111111	3F	63	Question mark
@	U+ 0040	01000000	40	64	At sign
A	U+ 0041	01000001	41	65	Uppercase A
B	U+ 0042	01000010	42	66	Uppercase B
C	U+ 0043	01000011	43	67	Uppercase C
D	U+ 0044	01000100	44	68	Uppercase D
E	U+ 0045	01000101	45	69	Uppercase E
F	U+ 0046	01000110	46	70	Uppercase F
G	U+ 0047	01000111	47	71	Uppercase G
H	U+ 0048	01001000	48	72	Uppercase H
I	U+ 0049	01001001	49	73	Uppercase I
J	U+ 004A	01001010	4A	74	Uppercase J
K	U+ 004B	01001011	4B	75	Uppercase K
L	U+ 004C	01001100	4C	76	Uppercase L
M	U+ 004D	01001101	4D	77	Uppercase M
N	U+ 004E	01001110	4E	78	Uppercase N
O	U+ 004F	01001111	4F	79	Uppercase O
P	U+ 0050	01010000	50	80	Uppercase P
Q	U+ 0051	01010001	51	81	Uppercase Q
R	U+ 0052	01010010	52	82	Uppercase R
S	U+ 0053	01010011	53	83	Uppercase S
T	U+ 0054	01010100	54	84	Uppercase T
U	U+ 0055	01010101	55	85	Uppercase U
V	U+ 0056	01010110	56	86	Uppercase V
W	U+ 0057	01010111	57	87	Uppercase W
X	U+ 0058	01011000	58	88	Uppercase X
Y	U+ 0059	01011001	59	89	Uppercase Y
Z	U+ 005A	01011010	5A	90	Uppercase Z

Symbol	Unicode	Binary	Hex	CP437 or CP1252	Description
[	U+ 005B	01011011	5B	91	Opening bracket
****	U+ 005C	01011100	5C	92	Backslash
]	U+ 005D	01011101	5D	93	Closing bracket
^	U+ 005E	01011110	5E	94	Caret - circumflex
_	U+ 005F	01011111	5F	95	Underscore
‘	U+ 0060	01100000	60	96	Grave accent
a	U+ 0061	01100001	61	97	Lowercase a
b	U+ 0062	01100010	62	98	Lowercase b
c	U+ 0063	01100011	63	99	Lowercase c
d	U+ 0064	01100100	64	100	Lowercase d
e	U+ 0065	01100101	65	101	Lowercase e
f	U+ 0066	01100110	66	102	Lowercase f
g	U+ 0067	01100111	67	103	Lowercase g
h	U+ 0068	01101000	68	104	Lowercase h
i	U+ 0069	01101001	69	105	Lowercase i
j	U+ 006A	01101010	6A	106	Lowercase j
k	U+ 006B	01101011	6B	107	Lowercase k
l	U+ 006C	01101100	6C	108	Lowercase l
m	U+ 006D	01101101	6D	109	Lowercase m
n	U+ 006E	01101110	6E	110	Lowercase n
o	U+ 006F	01101111	6F	111	Lowercase o
p	U+ 0070	01110000	70	112	Lowercase p
q	U+ 0071	01110001	71	113	Lowercase q
r	U+ 0072	01110010	72	114	Lowercase r
s	U+ 0073	01110011	73	115	Lowercase s
t	U+ 0074	01110100	74	116	Lowercase t
u	U+ 0075	01110101	75	117	Lowercase u
v	U+ 0076	01110110	76	118	Lowercase v
w	U+ 0077	01110111	77	119	Lowercase w
x	U+ 0078	01111000	78	120	Lowercase x
y	U+ 0079	01111001	79	121	Lowercase y
z	U+ 007A	01111010	7A	122	Lowercase z
{	U+ 007B	01111011	7B	123	Opening brace
|	U+ 007C	01111100	7C	124	Vertical bar
}	U+ 007D	01111101	7D	125	Closing brace
~	U+ 007E	01111110	7E	126	Equivalency sign - tilde
DEL	U+ 007F	01111111	7F	127	Delete •

The extended ASCII codes (character code 128-255)

There are several different variations of the 8-bit ASCII table. The table below is according to Windows-1252 (CP-1252) which is a superset of ISO 8859-1, also called ISO Latin-1, in terms of printable characters, but differs from the IANA's ISO-8859-1 by using displayable characters rather than extended control characters in the code point 128 to 159 range. Characters that differ from ISO-8859-1 are marked by light blue color.

Symbol	Unicode	Binary	Hex	CP1252	Unicode name
€	U+20AC	10000000	80	128	EURO SIGN
		10000001	81	129	Unused
‚	U+201A	10000010	82	130	SINGLE LOW-9 QUOTATION MARK
ƒ	U+0192	10000011	83	131	LATIN SMALL LETTER F WITH HOOK
„	U+201E	10000100	84	132	DOUBLE LOW-9 QUOTATION MARK
…	U+2026	10000101	85	133	HORIZONTAL ELLIPSIS
†	U+2020	10000110	86	134	DAGGER
‡	U+2021	10000111	87	135	DOUBLE DAGGER
ˆ	U+02C6	10001000	88	136	MODIFIER LETTER CIRCUMFLEX ACCENT
‰	U+2030	10001001	89	137	PER MILLE SIGN
Š	U+0160	10001010	8A	138	LATIN CAPITAL LETTER S WITH CARON
‹	U+2039	10001011	8B	139	SINGLE LEFT-POINTING ANGLE QUOTATION MARK
Œ	U+0152	10001100	8C	140	LATIN CAPITAL LIGATURE OE
		10001101	8D	141	Unused
Ž	U+017D	10001110	8E	142	LATIN CAPITAL LETTER Z WITH CARON
		10001111	8F	143	Unused
		10010000	90	144	Unused
‘	U+2018	10010001	91	145	LEFT SINGLE QUOTATION MARK
’	U+2019	10010010	92	146	RIGHT SINGLE QUOTATION MARK
“	U+201C	10010011	93	147	LEFT DOUBLE QUOTATION MARK
”	U+201D	10010100	94	148	RIGHT DOUBLE QUOTATION MARK
•	U+2022	10010101	95	149	BULLET
–	U+2013	10010110	96	150	EN DASH
—	U+2014	10010111	97	151	EM DASH
˜	U+02DC	10011000	98	152	SMALL TILDE
™	U+2122	10011001	99	153	TRADEMARK SIGN
š	U+0161	10011010	9A	154	LATIN SMALL LETTER S WITH CARON
›	U+203A	10011011	9B	155	SINGLE RIGHT-POINTING ANGLE QUOTATION MARK
		10011101	9D	157	Unused
œ	U+0153	10011100	9C	156	LATIN SMALL LIGATURE OE
ž	U+017E	10011110	9E	158	LATIN SMALL LETTER Z WITH CARON
Ÿ	U+0178	10011111	9F	159	LATIN CAPITAL LETTER Y WITH DIAERESIS
	U+00A0	10100000	A0	160	NO-BREAK SPACE
¡	U+00A1	10100001	A1	161	INVERTED EXCLAMATION MARK
¢	U+00A2	10100010	A2	162	CENT SIGN
£	U+00A3	10100011	A3	163	POUND SIGN
¤	U+00A4	10100100	A4	164	CURRENCY SIGN
¥	U+00A5	10100101	A5	165	YEN SIGN
¦	U+00A6	10100110	A6	166	BROKEN BAR
§	U+00A7	10100111	A7	167	SECTION SIGN
¨	U+00A8	10101000	A8	168	DIAERESIS
©	U+00A9	10101001	A9	169	COPYRIGHT SIGN
ª	U+00AA	10101010	AA	170	FEMININE ORDINAL INDICATOR
«	U+00AB	10101011	AB	171	LEFT-POINTING DOUBLE ANGLE QUOTATION MARK
¬	U+00AC	10101100	AC	172	NOT SIGN
	U+00AD	10101101	AD	173	SOFT HYPHEN Only visible at a line break that falls between syllables of a word, where it appears as a hyphen-minus, -.
®	U+00AE	10101110	AE	174	REGISTERED SIGN
¯	U+00AF	10101111	AF	175	MACRON
°	U+00B0	10110000	B0	176	DEGREE SIGN
±	U+00B1	10110001	B1	177	PLUS-MINUS SIGN
²	U+00B2	10110010	B2	178	SUPERSCRIPT TWO
³	U+00B3	10110011	B3	179	SUPERSCRIPT THREE
´	U+00B4	10110100	B4	180	ACUTE ACCENT
µ	U+00B5	10110101	B5	181	MICRO SIGN
¶	U+00B6	10110110	B6	182	PILCROW SIGN
·	U+00B7	10110111	B7	183	MIDDLE DOT
¸	U+00B8	10111000	B8	184	CEDILLA
¹	U+00B9	10111001	B9	185	SUPERSCRIPT ONE
00BA	U+00BA	10111010	BA	186	MASCULINE ORDINAL INDICATOR

Symbol	Unicode	Binary	Hex	CP1252	Unicode name
»	U+00BB	10111011	BB	187	RIGHT-POINTING DOUBLE ANGLE QUOTATION MARK
¼	U+00BC	10111100	BC	188	VULGAR FRACTION ONE QUARTER
½	U+00BD	10111101	BD	189	VULGAR FRACTION ONE HALF
¾	U+00BE	10111110	BE	190	VULGAR FRACTION THREE QUARTERS
¿	U+00BF	10111111	BF	191	INVERTED QUESTION MARK
À	U+00C0	11000000	C0	192	LATIN CAPITAL LETTER A WITH GRAVE
Á	U+00C1	11000001	C1	193	LATIN CAPITAL LETTER A WITH ACUTE
Â	U+00C2	11000010	C2	194	LATIN CAPITAL LETTER A WITH CIRCUMFLEX
Ã	U+00C3	11000011	C3	195	LATIN CAPITAL LETTER A WITH TILDE
Ä	U+00C4	11000100	C4	196	LATIN CAPITAL LETTER A WITH DIAERESIS
Å	U+00C5	11000101	C5	197	LATIN CAPITAL LETTER A WITH RING ABOVE
Æ	U+00C6	11000110	C6	198	LATIN CAPITAL LETTER AE
Ç	U+00C7	11000111	C7	199	LATIN CAPITAL LETTER C WITH CEDILLA
È	U+00C8	11001000	C8	200	LATIN CAPITAL LETTER E WITH GRAVE
É	U+00C9	11001001	C9	201	LATIN CAPITAL LETTER E WITH ACUTE
Ê	U+00CA	11001010	CA	202	LATIN CAPITAL LETTER E WITH CIRCUMFLEX
Ë	U+00CB	11001011	CB	203	LATIN CAPITAL LETTER E WITH DIAERESIS
Ì	U+00CC	11001100	CC	204	LATIN CAPITAL LETTER I WITH GRAVE
Í	U+00CD	11001101	CD	205	LATIN CAPITAL LETTER I WITH ACUTE
Î	U+00CE	11001110	CE	206	LATIN CAPITAL LETTER I WITH CIRCUMFLEX
Ï	U+00CF	11001111	CF	207	LATIN CAPITAL LETTER I WITH DIAERESIS
Ð	U+00D0	11010000	D0	208	LATIN CAPITAL LETTER ETH
Ñ	U+00D1	11010001	D1	209	LATIN CAPITAL LETTER N WITH TILDE
Ò	U+00D2	11010010	D2	210	LATIN CAPITAL LETTER O WITH GRAVE
Ó	U+00D3	11010011	D3	211	LATIN CAPITAL LETTER O WITH ACUTE
Ô	U+00D4	11010100	D4	212	LATIN CAPITAL LETTER O WITH CIRCUMFLEX
Õ	U+00D5	11010101	D5	213	LATIN CAPITAL LETTER O WITH TILDE
Ö	U+00D6	11010110	D6	214	LATIN CAPITAL LETTER O WITH DIAERESIS
×	U+00D7	11010111	D7	215	MULTIPLICATION SIGN
Ø	U+00D8	11011000	D8	216	LATIN CAPITAL LETTER O WITH STROKE
Ù	U+00D9	11011001	D9	217	LATIN CAPITAL LETTER U WITH GRAVE
Ú	U+00DA	11011010	DA	218	LATIN CAPITAL LETTER U WITH ACUTE
Û	U+00DB	11011011	DB	219	LATIN CAPITAL LETTER U WITH CIRCUMFLEX
Ü	U+00DC	11011100	DC	220	LATIN CAPITAL LETTER U WITH DIAERESIS
Ý	U+00DD	11011101	DD	221	LATIN CAPITAL LETTER Y WITH ACUTE
Þ	U+00DE	11011110	DE	222	LATIN CAPITAL LETTER THORN
ß	U+00DF	11011111	DF	223	LATIN SMALL LETTER SHARP S
à	U+00E0	11100000	E0	224	LATIN SMALL LETTER A WITH GRAVE
á	U+00E1	11100001	E1	225	LATIN SMALL LETTER A WITH ACUTE
â	U+00E2	11100010	E2	226	LATIN SMALL LETTER A WITH CIRCUMFLEX
ã	U+00E3	11100011	E3	227	LATIN SMALL LETTER A WITH TILDE
ä	U+00E4	11100100	E4	228	LATIN SMALL LETTER A WITH DIAERESIS
å	U+00E5	11100101	E5	229	LATIN SMALL LETTER A WITH RING ABOVE
æ	U+00E6	11100110	E6	230	LATIN SMALL LETTER AE
ç	U+00E7	11100111	E7	231	LATIN SMALL LETTER C WITH CEDILLA
è	U+00E8	11101000	E8	232	LATIN SMALL LETTER E WITH GRAVE
é	U+00E9	11101001	E9	233	LATIN SMALL LETTER E WITH ACUTE
ê	U+00EA	11101010	EA	234	LATIN SMALL LETTER E WITH CIRCUMFLEX
ë	U+00EB	11101011	EB	235	LATIN SMALL LETTER E WITH DIAERESIS
ì	U+00EC	11101100	EC	236	LATIN SMALL LETTER I WITH GRAVE
í	U+00ED	11101101	ED	237	LATIN SMALL LETTER I WITH ACUTE
î	U+00EE	11101110	EE	238	LATIN SMALL LETTER I WITH CIRCUMFLEX
ï	U+00EF	11101111	EF	239	LATIN SMALL LETTER I WITH DIAERESIS
ð	U+00F0	11110000	F0	240	LATIN SMALL LETTER ETH
ñ	U+00F1	11110001	F1	241	LATIN SMALL LETTER N WITH TILDE
ò	U+00F2	11110010	F2	242	LATIN SMALL LETTER O WITH GRAVE
ó	U+00F3	11110011	F3	243	LATIN SMALL LETTER O WITH ACUTE
ô	U+00F4	11110100	F4	244	LATIN SMALL LETTER O WITH CIRCUMFLEX
õ	U+00F5	11110101	F5	245	LATIN SMALL LETTER O WITH TILDE
ö	U+00F6	11110110	F6	246	LATIN SMALL LETTER O WITH DIAERESIS
÷	U+00F7	11110111	F7	247	DIVISION SIGN
ø	U+00F8	11111000	F8	248	LATIN SMALL LETTER O WITH STROKE
ù	U+00F9	11111001	F9	249	LATIN SMALL LETTER U WITH GRAVE
ú	U+00FA	11111010	FA	250	LATIN SMALL LETTER U WITH ACUTE
û	U+00FB	11111011	FB	251	LATIN SMALL LETTER U WITH CIRCUMFLEX
ü	U+00FC	11111100	FC	252	LATIN SMALL LETTER U WITH DIAERESIS
ý	U+00FD	11111101	FD	253	LATIN SMALL LETTER Y WITH ACUTE
þ	U+00FE	11111110	FE	254	LATIN SMALL LETTER THORN

The table below is often referred to as CP-437, the OEM Code, or the Graphic Code page. This code page is the legacy code page and is (*or was*) built into the PC BIOS. This code page differs from ISO-8859-1 in that displayable graphic characters are defined for code points 1-31. Code points above 127 on this page contain some of the Latin characters of ISO-8859-1 but in a completely different arrangement.

Symbol	Unicode	Binary	Hex	CP437 "OEM"	Unicode name
☺	U+263A	00000001	01	1	WHITE SMILING FACE
☻	U+263B	00000010	02	2	BLACK SMILING FACE
♥	U+2665	00000011	03	3	BLACK HEART SUIT
♦	U+2666	00000100	04	4	BLACK DIAMOND SUIT
♣	U+2663	00000101	05	5	BLACK CLUB SUIT
♠	U+2660	00000110	06	6	BLACK SPADE SUIT
•	U+2022	00000111	07	7	BULLET
◘	U+25D8	00001000	08	8	INVERSE BULLET
○	U+25CB	00001001	09	9	WHITE CIRCLE
◙	U+25D9	00001010	0A	10	INVERSE WHITE CIRCLE
♂	U+2642	00001011	0B	11	MALE SIGN
♀	U+2640	00001100	0C	12	FEMALE SIGN
♪	U+266A	00001101	0D	13	EIGHTH NOTE
♫	U+266B	00001110	0E	14	BEAMED EIGHTH NOTES
☼	U+263C	00001111	0F	15	WHITE SUN WITH RAYS
►	U+25BA	00010000	10	16	BLACK RIGHT-POINTING POINTER
◄	U+25C4	00010001	11	17	BLACK LEFT-POINTING POINTER
↕	U+2195	00010010	12	18	UP DOWN ARROW
‼	U+203C	00010011	13	19	DOUBLE EXCLAMATION MARK
¶	U+00B6	00010100	14	20	PILCROW SIGN
§	U+00A7	00010101	15	21	SECTION SIGN
▬	U+25AC	00010110	16	22	BLACK RECTANGLE
↨	U+21A8	00010111	17	23	UP DOWN ARROW WITH BASE
↑	U+2191	00011000	18	24	UPWARDS ARROW
↓	U+2193	00011001	19	25	DOWNWARDS ARROW
→	U+2192	00011010	1A	26	RIGHTWARDS ARROW
←	U+2190	00011011	1B	27	LEFTWARDS ARROW
∟	U+221F	00011100	1C	28	RIGHT ANGLE
↔	U+2194	00011101	1D	29	LEFT RIGHT ARROW
▲	U+25B2	00011110	1E	30	BLACK UP-POINTING TRIANGLE
▼	U+25BC	00011111	1F	31	BLACK DOWN-POINTING TRIANGLE
⌂	U+2302	01111111	7F	127	HOUSE
Ç	U+00C7	10000000	80	128	LATIN CAPITAL LETTER C WITH CEDILLA
ü	U+00FC	10000001	81	129	LATIN SMALL LETTER U WITH DIAERESIS
é	U+00E9	10000010	82	130	LATIN SMALL LETTER E WITH ACUTE
â	U+00E2	10000011	83	131	LATIN SMALL LETTER A WITH CIRCUMFLEX
ä	U+00E4	10000100	84	132	LATIN SMALL LETTER A WITH DIAERESIS
à	U+00E0	10000101	85	133	LATIN SMALL LETTER A WITH GRAVE
å	U+00E5	10000110	86	134	LATIN SMALL LETTER A WITH RING ABOVE
ç	U+00E7	10000111	87	135	LATIN SMALL LETTER C WITH CEDILLA
ê	U+00EA	10001000	88	136	LATIN SMALL LETTER E WITH CIRCUMFLEX
ë	U+00EB	10001001	89	137	LATIN SMALL LETTER E WITH DIAERESIS
è	U+00E8	10001010	8A	138	LATIN SMALL LETTER E WITH GRAVE
ï	U+00EF	10001011	8B	139	LATIN SMALL LETTER I WITH DIAERESIS
î	U+00EE	10001100	8C	140	LATIN SMALL LETTER I WITH CIRCUMFLEX
ì	U+00EC	10001101	8D	141	LATIN SMALL LETTER I WITH GRAVE
Ä	U+00C4	10001110	8E	142	LATIN CAPITAL LETTER A WITH DIAERESIS
Å	U+00C5	10001111	8F	143	LATIN CAPITAL LETTER A WITH RING ABOVE
É	U+00C9	10010000	90	144	LATIN CAPITAL LETTER E WITH ACUTE
æ	U+00E6	10010001	91	145	LATIN SMALL LETTER AE
Æ	U+00C6	10010010	92	146	LATIN CAPITAL LETTER AE
ô	U+00F4	10010011	93	147	LATIN SMALL LETTER O WITH CIRCUMFLEX
ö	U+00F6	10010100	94	148	LATIN SMALL LETTER O WITH DIAERESIS
ò	U+00F2	10010101	95	149	LATIN SMALL LETTER O WITH GRAVE
û	U+00FB	10010110	96	150	LATIN SMALL LETTER U WITH CIRCUMFLEX
ù	U+00F9	10010111	97	151	LATIN SMALL LETTER U WITH GRAVE
ÿ	U+00FF	10011000	98	152	LATIN SMALL LETTER Y WITH DIAERESIS
Ö	U+00D6	10011001	99	153	LATIN CAPITAL LETTER O WITH DIAERESIS
Ü	U+00DC	10011010	9A	154	LATIN CAPITAL LETTER U WITH DIAERESIS

Symbol	Unicode	Binary	Hex	CP437 "OEM"	Unicode name
¢	U+00A2	10011011	9B	155	CENT SIGN
£	U+00A3	10011100	9C	156	POUND SIGN
¥	U+00A5	10011101	9D	157	YEN SIGN
₧	U+20A7	10011110	9E	158	PESETA SIGN
ƒ	U+0192	10011111	9F	159	LATIN SMALL LETTER F WITH HOOK
á	U+00E1	10100000	A0	160	LATIN SMALL LETTER A WITH ACUTE
í	U+00ED	10100001	A1	161	LATIN SMALL LETTER I WITH ACUTE
ó	U+00F3	10100010	A2	162	LATIN SMALL LETTER O WITH ACUTE
ú	U+00FA	10100011	A3	163	LATIN SMALL LETTER U WITH ACUTE
ñ	U+00F1	10100100	A4	164	LATIN SMALL LETTER N WITH TILDE
Ñ	U+00D1	10100101	A5	165	LATIN CAPITAL LETTER N WITH TILDE
ª	U+00AA	10100110	A6	166	FEMININE ORDINAL INDICATOR
º	U+00BA	10100111	A7	167	MASCULINE ORDINAL INDICATOR
¿	U+00BF	10101000	A8	168	INVERTED QUESTION MARK
⌐	U+2310	10101001	A9	169	REVERSED NOT SIGN
¬	U+00AC	10101010	AA	170	NOT SIGN
½	U+00BD	10101011	AB	171	VULGAR FRACTION ONE HALF
¼	U+00BC	10101100	AC	172	VULGAR FRACTION ONE QUARTER
¡	U+00A1	10101101	AD	173	INVERTED EXCLAMATION MARK
«	U+00AB	10101110	AE	174	LEFT-POINTING DOUBLE ANGLE QUOTATION MARK
»	U+00BB	10101111	AF	175	RIGHT-POINTING DOUBLE ANGLE QUOTATION MARK
░	U+2591	10110000	B0	176	LIGHT SHADE
▒	U+2592	10110001	B1	177	MEDIUM SHADE
▓	U+2593	10110010	B2	178	DARK SHADE
│	U+2502	10110011	B3	179	BOX DRAWINGS LIGHT VERTICAL
┤	U+2524	10110100	B4	180	BOX DRAWINGS LIGHT VERTICAL AND LEFT
Á	U+00C1	10110101	B5	181	LATIN CAPITAL LETTER A WITH ACUTE
╡	U+2562	10110110	B6	182	BOX DRAWINGS VERTICAL DOUBLE AND LEFT SINGLE
╖	U+2556	10110111	B7	183	BOX DRAWINGS DOWN DOUBLE AND LEFT SINGLE
╕	U+2555	10111000	B8	184	BOX DRAWINGS DOWN SINGLE AND LEFT DOUBLE
╣	U+2563	10111001	B9	185	BOX DRAWINGS DOUBLE VERTICAL AND LEFT
║	U+2551	10111010	BA	186	BOX DRAWINGS DOUBLE VERTICAL
╗	U+2557	10111011	BB	187	BOX DRAWINGS DOUBLE DOWN AND LEFT
╝	U+255D	10111100	BC	188	BOX DRAWINGS DOUBLE UP AND LEFT
╜	U+255C	10111101	BD	189	BOX DRAWINGS UP DOUBLE AND LEFT SINGLE
╛	U+255B	10111110	BE	190	BOX DRAWINGS UP SINGLE AND LEFT DOUBLE
┐	U+2510	10111111	BF	191	BOX DRAWINGS LIGHT DOWN AND LEFT
└	U+2514	11000000	C0	192	BOX DRAWINGS LIGHT UP AND RIGHT
┴	U+2534	11000001	C1	193	BOX DRAWINGS LIGHT UP AND HORIZONTAL
┬	U+252C	11000010	C2	194	BOX DRAWINGS LIGHT DOWN AND HORIZONTAL
├	U+251C	11000011	C3	195	BOX DRAWINGS LIGHT VERTICAL AND RIGHT
─	U+2500	11000100	C4	196	BOX DRAWINGS LIGHT HORIZONTAL
┼	U+253C	11000101	C5	197	BOX DRAWINGS LIGHT VERTICAL AND HORIZONTAL
╞	U+255E	11000110	C6	198	BOX DRAWINGS VERTICAL SINGLE AND RIGHT DOUBLE
╟	U+255F	11000111	C7	199	BOX DRAWINGS VERTICAL DOUBLE AND RIGHT SINGLE
╚	U+255A	11001000	C8	200	BOX DRAWINGS DOUBLE UP AND RIGHT
╔	U+2554	11001001	C9	201	BOX DRAWINGS DOUBLE DOWN AND RIGHT
╩	U+2569	11001010	CA	202	BOX DRAWINGS DOUBLE UP AND HORIZONTAL
╦	U+2566	11001011	CB	203	BOX DRAWINGS DOUBLE DOWN AND HORIZONTAL
╠	U+2560	11001100	CC	204	BOX DRAWINGS DOUBLE VERTICAL AND RIGHT
═	U+2550	11001101	CD	205	BOX DRAWINGS DOUBLE HORIZONTAL
╬	U+256C	11001110	CE	206	BOX DRAWINGS DOUBLE VERTICAL AND HORIZONTAL
╧	U+2567	11001111	CF	207	BOX DRAWINGS UP SINGLE AND HORIZONTAL DOUBLE
╨	U+2568	11010000	D0	208	BOX DRAWINGS UP DOUBLE AND HORIZONTAL SINGLE
╤	U+2564	11010001	D1	209	BOX DRAWINGS DOWN SINGLE AND HORIZONTAL DOUBLE
╥	U+2565	11010010	D2	210	BOX DRAWINGS DOWN DOUBLE AND HORIZONTAL SINGLE
╙	U+2559	11010011	D3	211	BOX DRAWINGS UP DOUBLE AND RIGHT SINGLE
╘	U+2558	11010100	D4	212	BOX DRAWINGS UP SINGLE AND RIGHT DOUBLE
╒	U+2552	11010101	D5	213	BOX DRAWINGS DOWN SINGLE AND RIGHT DOUBLE
╓	U+2553	11010110	D6	214	BOX DRAWINGS DOWN DOUBLE AND RIGHT SINGLE
╫	U+256B	11010111	D7	215	BOX DRAWINGS VERTICAL DOUBLE AND HORIZONTAL SINGLE
╪	U+256A	11011000	D8	216	BOX DRAWINGS VERTICAL SINGLE AND HORIZONTAL DOUBLE
┘	U+2518	11011001	D9	217	BOX DRAWINGS LIGHT UP AND LEFT
┌	U+250C	11011010	DA	218	BOX DRAWINGS LIGHT DOWN AND RIGHT
█	U+2588	11011011	DB	219	FULL BLOCK
▄	U+2584	11011100	DC	220	LOWER HALF BLOCK
▌	U+258C	11011101	DD	221	LEFT HALF BLOCK

Symbol	Unicode	Binary	Hex	CP437 "OEM"	Unicode name
▐	U+2590	11011110	DE	222	RIGHT HALF BLOCK
▀	U+2580	11011111	DF	223	UPPER HALF BLOCK
α	U+03B1	11100000	E0	224	GREEK SMALL LETTER ALPHA
ß	U+00DF	11100001	E1	225	LATIN SMALL LETTER SHARP S
Γ	U+0393	11100010	E2	226	GREEK CAPITAL LETTER GAMMA
π	U+03C0	11100011	E3	227	GREEK SMALL LETTER PI
Σ	U+03A3	11100100	E4	228	GREEK CAPITAL LETTER SIGMA
σ	U+03C3	11100101	E5	229	GREEK SMALL LETTER SIGMA
µ	U+00B5	11100110	E6	230	MICRO SIGN
τ	U+03C4	11100111	E7	231	GREEK SMALL LETTER TAU
Φ	U+03A6	11101000	E8	232	GREEK CAPITAL LETTER PHI
Θ	U+0398	11101001	E9	233	GREEK CAPITAL LETTER THETA
Ω	U+03A9	11101010	EA	234	GREEK CAPITAL LETTER OMEGA
δ	U+03B4	11101011	EB	235	GREEK SMALL LETTER DELTA
∞	U+221E	11101100	EC	236	INFINITY
φ	U+03C6	11101101	ED	237	GREEK SMALL LETTER PHI
ε	U+03B5	11101110	EE	238	GREEK SMALL LETTER EPSILON
∩	U+2229	11101111	EF	239	INTERSECTION
≡	U+2261	11110000	F0	240	IDENTICAL TO
±	U+00B1	11110001	F1	241	PLUS-MINUS SIGN
≥	U+2265	11110010	F2	242	GREATER-THAN OR EQUAL TO
≤	U+2264	11110011	F3	243	LESS-THAN OR EQUAL TO
⌠	U+2320	11110100	F4	244	TOP HALF INTEGRAL
⌡	U+2321	11110101	F5	245	BOTTOM HALF INTEGRAL
÷	U+00F7	11110110	F6	246	DIVISION SIGN
≈	U+2248	11110111	F7	247	ALMOST EQUAL TO
°	U+00B0	11111000	F8	248	DEGREE SIGN
∙	U+2219	11111001	F9	249	BULLET OPERATOR
·	U+00B7	11111010	FA	250	MIDDLE DOT
√	U+221A	11111011	FB	251	SQUARE ROOT
ⁿ	U+207F	11111100	FC	252	SUPERSCRIPT LATIN SMALL LETTER N
²	U+00B2	11111101	FD	253	SUPERSCRIPT TWO
■	U+25A0	11111110	FE	254	BLACK SQUARE
	U+00A0	11111111	FF	255	NO-BREAK SPACE

See: https://en.wikipedia.org/wiki/Alt_code

ISO 8859 Parts 1-16

Comparison of the various parts (1–16) of ISO/IEC 8859

Binary	Oct	Dec	Hex	1	2	3	4	5	6	7	8	9	10	11	13	14	15	16
1010 0000	240	160	A0	Non-breaking space (NBSP)														
1010 0001	241	161	A1	¡	Ą	Ħ	Ą	Ë		‘		¡	Ą	ก	”	Ḃ	¡	Ą
1010 0010	242	162	A2	¢	˘		к	Ђ		’	¢	¢	Ē	ข	¢	ḃ	¢	ą
1010 0011	243	163	A3	£	Ł	£	Ŗ	Ѓ		£			Ģ	ฃ	£			Ł
1010 0100	244	164	A4	¤				Є	¤	€	¤		Ī	ค	¤	Ċ	€	
1010 0101	245	165	A5	¥	Ľ		Ĩ	S		Đ	¥		Ĩ	ฅ	„	ċ	¥	„
1010 0110	246	166	A6	¦	Ś	Ĥ	Ļ	І		¦			Ķ	ฆ	¦	Ḋ	Š	
1010 0111	247	167	A7	§				Ї		§				ง	§			
1010 1000	250	168	A8	¨				J		¨			Ļ	จ	Ø	Ẁ	š	
1010 1001	251	169	A9	©	Š	İ	Š	Љ		©			Đ	ฉ	©			
1010 1010	252	170	AA	ª	Ş		Ē	Њ		˛	×	ª	Š	ช	Ŗ	Ẃ	ª	Ş
1010 1011	253	171	AB	«	Ť	Ğ	Ģ	Ћ		«			Ŧ	ซ	«	ḋ	«	
1010 1100	254	172	AC	¬	Ź	Ĵ	Ŧ	Ќ	،	¬			Ž	ฌ	¬	Ỳ	¬	Ź
1010 1101	255	173	AD	Soft hyphen (SHY)										ญ	SHY			
1010 1110	256	174	AE	®	Ž		Ž	Ў			®		Ū	ฎ	®			ź
1010 1111	257	175	AF	¯	Ż		¯	Џ		—	¯		Ŋ	ฏ	Æ	Ÿ	¯	Ż
1011 0000	260	176	B0	°				А		°				ฐ	°	Ḟ	°	
1011 0001	261	177	B1	±	ą	ħ	ą	Б		±			ą	ฑ	±	ḟ	±	
1011 0010	262	178	B2	²	˛	²	˛	В		²			ē	ฒ	²	Ġ	²	Č
1011 0011	263	179	B3	³	ł	³	ŗ	Г		³			ģ	ณ	³	ġ	³	ł
1011 0100	264	180	B4	´				Д		´	´		ī	ด	“	Ṁ	Ž	
1011 0101	265	181	B5	µ	ľ	µ	ĩ	Е		̋	µ		ĩ	ต	µ	ṁ	µ	”
1011 0110	266	182	B6	¶	ś	ĥ	ļ	Ж		Ά	¶		ķ	ถ	¶			
1011 0111	267	183	B7	·	ˇ	·	ˇ	З		·				ท	·	Ṗ	·	
1011 1000	270	184	B8	¸				И		Έ	¸		ļ	ธ	ø	ẁ	ž	
1011 1001	271	185	B9	¹	š	ı	š	Й		Ή	¹		đ	น	¹	ṗ	¹	č
1011 1010	272	186	BA	º	ş		ē	К		Ί	÷	º	š	บ	ŗ	ẃ	º	ş
1011 1011	273	187	BB	»	ť	ğ	ģ	Л	؛	»			ŧ	ป	»	Ṡ	»	
1011 1100	274	188	BC	¼	ź	ĵ	ŧ	М		Ό	¼		ž	ผ	¼	ỳ	Œ	
1011 1101	275	189	BD	½	˝	½	Ŋ	Н		½			—	ฝ	½	Ẅ	œ	
1011 1110	276	190	BE	¾	ž		ž	О		Ύ	¾		ū	พ	¾	ẅ	Ÿ	
1011 1111	277	191	BF	¿	ż		ŋ	П	؟	Ώ		¿	ŋ	ฟ	æ	ṡ	¿	ż
1100 0000	300	192	C0	À	Ŕ	À	Ā	Р		Ï		À	Ā	ภ	Ą	À		

Comparison of the various parts (1–16) of ISO/IEC 8859																		
Binary	Oct	Dec	Hex	1	2	3	4	5	6	7	8	9	10	11	13	14	15	16
1100 0001	301	193	C1	Á				С	ء	А		Á		ม	Į	Á		
1100 0010	302	194	C2	Â				Т	آ	В		Â		ย	Ā	Â		
1100 0011	303	195	C3	Ã	Ă		Ã	У	أ	Г		Ã		ร	Ć	Ã		Ă
1100 0100	304	196	C4	Ä				Ф	ؤ	Δ		Ä		ฤ	Ä			
1100 0101	305	197	C5	Å	Ĺ	Ċ	Å	Х	إ	Е		Å		ล	Å			Ć
1100 0110	306	198	C6	Æ	Ć	Ĉ	Æ	Ц	ئ	Ζ		Æ		ฦ	Ę	Æ		
1100 0111	307	199	C7	Ç			Į	Ч	ا	Н		Ç	Į	ว	Ē	Ç		
1100 1000	310	200	C8	È	Č	È	Č	Ш	ب	Θ		È	Č	ศ	Č	È		
1100 1001	311	201	C9	É				Щ	ة	І		É		ษ	É			
1100 1010	312	202	CA	Ê	Ę	Ê	Ę	Ъ	ت	К		Ê	Ę	ส	Ź	Ê		
1100 1011	313	203	CB	Ë				Ы	ث	Λ		Ë		ห	Ė	Ë		
1100 1100	314	204	CC	Ì	Ě	Ì	Ė	Ь	ج	М		Ì	Ė	ฬ	Ģ	Ì		
1100 1101	315	205	CD	Í				Э	ح	N		Í		อ	Ķ	Í		
1100 1110	316	206	CE	Î				Ю	خ	Ξ		Î		ฮ	Ī	Î		
1100 1111	317	207	CF	Ï	Ď	Ï	Ī	Я	د	О		Ï		ฯ	Ļ	Ï		
1101 0000	320	208	D0	Đ	Đ		Đ	а	ذ	П		Ğ	Đ	ะ	Š	Ŵ	Đ	
1101 0001	321	209	D1	Ñ	Ń	Ñ	Ņ	б	ر	Р		Ñ	Ņ	ั	Ń	Ñ		Ń
1101 0010	322	210	D2	Ò	Ň	Ò	Ō	в	ز			Ò	Ō	า	Ņ	Ò		
1101 0011	323	211	D3	Ó			Ķ	г	س	Σ		Ó		ำ	Ó			
1101 0100	324	212	D4	Ô				д	ش	Т		Ô		ิ	Õ	Ô		
1101 0101	325	213	D5	Õ	Ő	Ġ	Õ	е	ص	У		Õ		ี	Ő			
1101 0110	326	214	D6	Ö				ж	ض	Ф		Ö		ึ	Ö			
1101 0111	327	215	D7	×				з	ط	Х		×	Ũ	ื	×	Ť	×	Ś
1101 1000	330	216	D8	Ø	Ř	Ĝ	Ø	и	ظ	Ψ		Ø		ุ	Ų	Ø		Ű
1101 1001	331	217	D9	Ù	Ů	Ù	Ų	й	ع	Ω		Ù	Ų	ู	Ł	Ù		
1101 1010	332	218	DA	Ú				к	غ	Ї		Ú		ฺ	Ś	Ú		
1101 1011	333	219	DB	Û	Ű	Û		л		Ÿ		Û			Ū	Û		
1101 1100	334	220	DC	Ü				м		ά		Ü			Ü			
1101 1101	335	221	DD	Ý		Ŭ	Ũ	н		έ		İ	Ý		Ż	Ý		Ę
1101 1110	336	222	DE	Þ	Ţ	Ŝ	Ū	о		ή		Ş	Þ		Ž	Ŷ	Þ	Ţ
1101 1111	337	223	DF	ß				п		ί	_	ß		฿	ß			
1110 0000	340	224	E0	à	ŕ	à	ā	р	-	ΰ	א	à	ā	เ	ą	à		
1110 0001	341	225	E1	á				с	ف	α	ב	á		แ	į	á		
1110 0010	342	226	E2	â				т	ق	β	ג	â		โ	ā	â		
1110 0011	343	227	E3	ã	ă		ã	у	ك	γ	ד	ã		ใ	ć	ã		ă
1110 0100	344	228	E4	ä				ф	ل	δ	ה	ä		ไ	ä			

Comparison of the various parts (1–16) of ISO/IEC 8859																		
Binary	**Oct**	**Dec**	**Hex**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**13**	**14**	**15**	**16**
1110 0101	345	229	E5	å	Í	ċ	å	х	م	ε	ı	å		ገ	å			ć
1110 0110	346	230	E6	æ	ć	ĉ	æ	ц	ن	ζ	ɪ	æ		ๅ	ę	æ		
1110 0111	347	231	E7	ç			į	ч	ه	ŋ	n	ç	į	◌̃	ē	ç		
1110 1000	350	232	E8	è	č	è	č	ш	و	θ	ʊ	è	č	◌́	č	è		
1110 1001	351	233	E9	é				щ	ى	ι	ˈ	é		◌̉	é			
1110 1010	352	234	EA	ê	ę	ê	ę	ъ	ي	к	ገ	ê	ę	◌̃	ź	ê		
1110 1011	353	235	EB	ë				ы	ً	λ	ɔ	ë		◌̇	ė	ë		
1110 1100	354	236	EC	ì	ě	ì	ė	ь	ٌ	μ	somethingelse	ì	ė	◌̀	ģ	ì		
1110 1101	355	237	ED	í				э	ٍ	ν	ɒ	í		◌̊	ķ	í		
1110 1110	356	238	EE	î				ю	َ	ξ	ɯ	î		◌̂	ī	î		
1110 1111	357	239	EF	ï	ď	ï	ī	я	ُ	ο	ı	ï		◌	ļ	ï		
1111 0000	360	240	F0	ð	đ		đ	№	ِ	π	ɲ	ğ	ð	ο	š	ŵ	ð	đ
1111 0001	361	241	F1	ñ	ń	ñ	ņ	ё	ّ	ρ	o	ñ	ņ	๑	ń	ñ		ń
1111 0010	362	242	F2	ò	ň	ò	ō	ђ	ْ	ς	ע	ò	ō	๒	ņ	ò		
1111 0011	363	243	F3	ó			ķ	ѓ		σ	ɳ	ó		๓	ó			
1111 0100	364	244	F4	ô				є		τ	ɵ	ô		๔	ō	ô		
1111 0101	365	245	F5	õ	ő	ġ	õ	ѕ		υ	γ	õ		๕	ő			
1111 0110	366	246	F6	ö				і		φ	ɤ	ö		๖	ö			
1111 0111	367	247	F7	÷				ї		χ	ק	÷	ũ	๗	÷	ṫ	÷	ś
1111 1000	370	248	F8	ø	ř	ĝ	ø	ј		ψ	ר	ø		๘	ų	ø		ű
1111 1001	371	249	F9	ù	ů	ù	ų	љ		ω	ש	ù	ų	๙	ł	ù		
1111 1010	372	250	FA	ú				њ		ї	ת	ú		๚	ś	ú		
1111 1011	373	251	FB	û	ű	û		ћ		ü		û		๛	ū	û		
1111 1100	374	252	FC	ü				ќ		ó		ü			ü			
1111 1101	375	253	FD	ý		ŭ	ū	§		ú	LRM	ı	ý		ż	ý		ę
1111 1110	376	254	FE	þ	ţ	ŝ	ū	ў		ώ	RLM	ş	þ		ž	ŷ	þ	ţ
1111 1111	377	255	FF	ÿ	·			џ				ÿ	к		’	ÿ		
Binary	**Oct**	**Dec**	**Hex**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**13**	**14**	**15**	**16**

At position 0xA0 there's always the non breaking space and 0xAD is mostly the soft hyphen, which only shows at line breaks. Other empty fields are either ▯ unassigned or the system used is not able to display them.
There are ▯ new additions as ISO/IEC 8859-7:2003 and ISO/IEC 8859-8:1999 versions. LRM stands for left-to-right mark (U+200E) and RLM stands for right-to-left mark (U+200F).

Appendix E Com Port parameters

Serial Master uses the 'newer' syntax in setting Device Control Block (DCB) parameters.

RS232_Settings = COM<n>[:] [baud=<b>] [parity=<p>] [data=<d>] [stop=<s>] [to={on|off}] [xon={on|off}] [odsr={on|off}] [octs={on|off}] [dtr={on|off|hs}] [rts={on|off|hs|tg}] [idsr={on|off}]

Parameter	Description
com<m>[:]	Specifies the number of the asynchronous communications port.
Baud={110|300|600|1200|2400|4800|9600|19200|38400|57600|115200}	Specifies the bit rate of the serial interface.
parity={n|o|e|m|s}	Specifies n= no parity, o=odd parity, e = even parity, m = Mark parity bit, s = Space partity bit. Not all ports support m or s and Serial Master will not set them.
data={5|6|7|8}	Specifies the number of data bits transmitted. Not all devices support 5 or 6.
Stop={0|1|2}	Specifies the number of stop bits. 0=one stop bit, 1= 1.5 stop bit, 2=2 stop bits
to={on | off}	Specifies whether the device uses infinite time out processing. The default value is off. Turning this option on means that the device will never stop waiting to receive a response from a host or client computer. Serial Master sets time-outs separately so that the driver returns any received data available immediately. Character transmit is set to 'not used' .
xon={on | off}	Specifies whether the system allows the XON/XOFF protocol. This protocol provides flow control character codes for serial communications, enhancing reliability, but reducing performance.
odsr={on | off}	Tx_Flow Control 'DSR' sets this parameter on. When 'on' and DSR is turned off, output is suspended until DSR is turned on.
octs={on | off}	Tx_Flow Control 'CTS' sets this parameter on. When 'on' and CTS is turned off, output is suspended until CTS is turned on.
dtr={on | off | hs}	Specifies whether the system turns on the Data Terminal Ready (DTR) output handshake. Setting this value to 'on', provides a constant signal to show the terminal is ready to receive data. Rx_Flow Control 'DTR' sets this value to 'hs' providing a receive flow control signal based on receive buffer space availability.
rts={on | off | hs | tg}	Setting this value to 'on' or 'off', provides a constant signal to show the terminal is ready or not ready to send data. Rx_Flow Control 'RTS' sets this value to 'hs' provides a receive flow control signal based on receive buffer space availability. Tx_Flow Control 'RTS' sets this value to 'tg' that turns RTS on when data is in the Tx buffer and turns RTS off when the transmit buffer is empty.
idsr={on | off}	The driver ignores any bytes received, unless the DSR modem input line is 'On/Active'. You must turn Rx_Flow Control 'DSR' to this option on to use DSR handshaking.

Device control block (DCB) Parameters

Parameter size	Parameter label	Allowed settings	Pre Windows XP mode ends with			
			x	p	others	
DWORD	DCBlength;	set to sizeof(DCB)				
DWORD	BaudRate;					baud=
DWORD	fBinary:1;	must be TRUE.				
DWORD	fParity : 1;	True= checked, False =no check				parity=
DWORD	fOutxCtsFlow : 1;	TRUE, the CTS signal is monitored for output flow control.	FALSE	TRUE	FALSE	octs=on|off
DWORD	fOutxDsrFlow : 1;	TRUE, the DSR signal is monitored for output flow control	FALSE	TRUE	FALSE	odsr=on|off
DWORD	fDtrControl : 2;	Off=00, on=01, hs=02	on	hs	on	dtr=
DWORD	fDsrSensitivity : 1;					idsr=on|off
DWORD	fTXContinueOnXoff : 1;					
DWORD	fOutX : 1;	XON/XOFF flow control is used during transmission.	TRUE	FALSE	FALSE	Xon=on|off
DWORD	fInX : 1;	XON/XOFF flow control is used during reception.	TRUE	FALSE	FALSE	Xon=on|off
DWORD	fErrorChar : 1;	If true, bytes with parity errors are replaced. See ErrorChar				
DWORD	fNull : 1;	if TRUE, null bytes are discarded.				
DWORD	fRtsControl : 2;	See note 1:	on	hs	on	Rts=
DWORD	fAbortOnError : 1;	TRUE, on error driver terminates all R/W operations. See ClearCommError .				
DWORD	fDummy2 : 17;	Reserved; do not use				
WORD	wReserved;	Reserved; must be zero				
WORD	XonLim;	Min buffer bytes to send Xon after Xoff				
WORD	XoffLim;	Remaining buffer bytes to send Xoff				
BYTE	ByteSize;	Data=5,6,7, or 8 bits				data=
BYTE	Parity;	parity=n=00, o=01, e=02, m=03, s=04				parity=
BYTE	StopBits;	0=one bit, 1= 1.5 bits, 2=2 stop bits				Stop=
char	XonChar;					
char	XoffChar;					
char	ErrorChar;	Replaces bytes received with a parity error. fParity and fErrorChar must be true.				
char	EofChar;					
char	EvtChar;					
WORD	wReserved1;	Reserved; must be zero				

See https://learn.microsoft.com/en-us/windows/win32/api/winbase/ns-winbase-dcb

Note1:

Value	Meaning
RTS_CONTROL_DISABLE 0x00	Disables the RTS line when the device is opened and leaves it disabled. Setting value is selected by Mode setting rts=off.
RTS_CONTROL_ENABLE 0x01	Enables the RTS line when the device is opened and leaves it on. Setting value is selected by Mode setting rts=on.
RTS_CONTROL_HANDSHAKE 0x02	Enables RTS handshaking. (RTR functionality) The driver raises the RTS line when the "type-ahead" (input) buffer is less than one-half full and lowers the RTS line when the buffer is more than three-quarters full. If handshaking is enabled, it is an error for the application to adjust the line by using the EscapeCommFunction function. Setting value is selected by Mode setting rts=hs.
RTS_CONTROL_TOGGLE 0x03	Specifies that the RTS line will be high if bytes are available for transmission. After all buffered bytes have been sent, the RTS line will be low. Setting value is selected by Mode setting rts=tg.
DTR_CONTROL_DISABLE 0x00	Disables the DTR line when the device is opened and leaves it disabled. Setting value is selected by Mode setting rts=off.
DTR_CONTROL_ENABLE 0x01	Enables the DTR line when the device is opened and leaves it on. Setting value is selected by Mode setting rts=on.
DTR_CONTROL_HANDSHAKE 0x02	Enables DTR handshaking. (receive flow control) If handshaking is enabled, it is an error for the application to adjust the line by using the EscapeCommFunction function. Control conditions are not well defined, but if the setting is supported are likely to be the same as RTS_CONTROL_HANDSHAKE. Setting value is selected by Mode setting rts=hs.