MATRIX ORBITAL

VK204-25
Technical Manual

Revision: 1.0

Contents

1 Introduction

The VK204-25 comes equipped with the following features;

- 20 column by 4 line text display
- Built in font with provision for up to 8 user defined characters
- Speeds from 1200 bps to a lighting fast 19.2 Kbps over RS-232
- Communication over I^{2}C or RS-232
- Use of up to 16 modules on the same 2 wire I^{2}C interface
- Fully buffered so that no delays in transmission are ever necessary
- Software controlled brightness
- Display with configurable time-out setting up to 180 minutes
- Use of up to a 25 key keypad with a 10 key buffer
- Six general purpose outputs for a variety of applications
- Horizontal or vertical bar graphs
- Variable power options, +5V or +7V to +15V or +7V to +30V
- Extended temperature option
- Fits Matrix Orbital's dual PC Bay inserts without modifications
- Built in large digits

1.1 What to Expect From the VK204-25

The VK204-25 is designed as the display unit for an associated controller. The controller may be anything from a single board, special purpose micro-controller to a PC, depending on the application. This controller is responsible for what is displayed on the screen of the display.

The display provides a simple command structure to allow text and bar graphs to be displayed on the screen. Text fonts are built in and use standard ASCII mapping. Provision is made for up to 8 user defined characters.

The screen is luminous for low-light situations. Display may be turned on or off under program control. Brightness is adjustable to compensate for differing lighting conditions.

General purpose outputs allow the controller to switch up to six electronic or electro-mechanical devices by issuing commands to the display unit. These can be used for controlling LEDs, relays, etc.

1.2 What Not to Expect From the VK204-25

The display does not include bitmap graphics capability, except that permitted by defining special characters.

1.3 Keypad Interface

The keypad interface takes row / column input and converts it to ASCII characters, which are delivered out the RS-232 or I^{2}C port to the associated controller. Note that the keypad is not used to directly control any aspect of the operation of the display, which acts simply as a matrix to serial converter. To use the keypad to control the display, the controller must be programmed accordingly.

1.4   Setup for Testing

Before setting up the application the user may want to try out the display. This is easily done with a PC. If not equipped with a dual bay PC mounting kit, the following will be required;

- A 4-pin power connector of the type used to connect 3.5" floppy drive. Take care not to connect the display to an unmodified spare power connector in a PC.
- A 5V power supply.
- A PC with a spare RS-232 port (COM1 or COM2).
- A 9 or 25 pin RS-232 serial cable. If using a 25 conductor cable, a 9 to 25 pin adapter will be required.
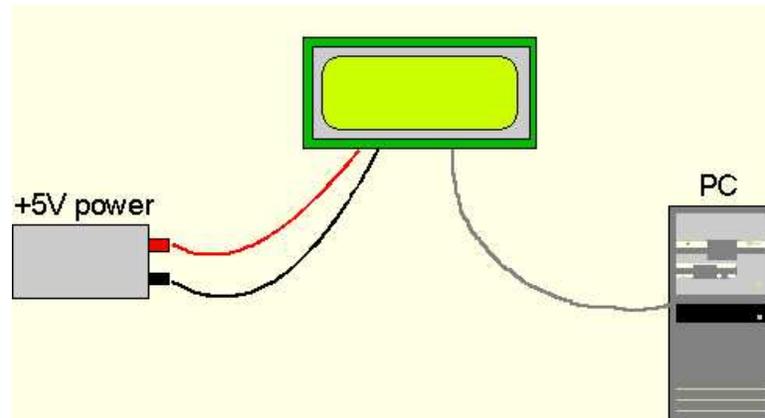
Figure 1: Connections for Testing

1. Refer to the Figure above for the following steps.
2. Wire the connector to the power supply. On most connectors the RED lead will go to +5V and the BLACK lead to GND.

> **NOTE**   The Manufacturer's Warranty becomes void if the unit is subjected to over-voltage or reversed polarity.

3. Connect the display to the PC using the serial cable and adapter if required. Make sure the RS-232 cable includes the required ground lead. There must be no voltage differential between the RS-232 ground and the power supply ground.
4. Connect the power connector, making sure that the +5V goes to V+. Turn on the power: the display should come on.

1.5   Trying Out the VK204-25

The unit should be connected to power. The PC and display should be on. To experiment with typing text, run a PC terminal program, such as Hyperterm. Make sure it's configured to use the correct port. Set

the baud rate to 19,200. If characters are typed on the keyboard, they should now appear on the display screen. Text will wrap around to the next line when the end of a line has been reached.

A few common ASCII control characters work as follows;

Table 1: Common ASCII Control Characters

Character	Hex value	Function
CR	0x0D	Moves cursor to beginning of the current line.
LF	0x0A	Moves cursor to beginning of the next (or previous) line.
FF	0x0C	Clears the display and puts the cursor at the top left.
BS	0x08	Moves the cursor one position to the left and clears that position.

NOTE   These command characters are not guaranteed to work on other Matrix Orbital display modules.

To exercise some of the other features of the display a program (in any convenient language such as Basic or C) will need to be written in order to issue the required command strings. Most terminal programs are unable to issue the 0xFE character needed as a command prefix.

1.6   Trying out a Keypad

Since a number of different keypad types can be connected to the display, the results may be a little unpredictable. At this point all we need to do is make sure that the keypad and interface work, and possibly generate an ASCII map for any programming needs. The keypad interface on the display converts a row / column connection to an ASCII character. By default, a keypress is transmitted as serial data immediately. Keypad buffering can be selected using the appropriate commands.

1.6.1   Here's what to Do

1. The PC should be running a terminal program, such as Hyperterm.

2. With the display connected to the PC, plug in the keypad. If the connector has fewer pins than the one on the display, center it as well as possible.

NOTES

- The keypad connector must be wired with columns on one side and rows on the other side of the center of the connector. If the keypad isn't wired this way an adapter must be made or the connector must be rewired to meet this requirement.
- The connector is reversible. Reversing the connector will not damage the keypad or the display, but will however, change the ASCII character map.

---

3. Press a key on the keypad. An upper case ASCII character (A-Y) should appear on the PC screen. Different keys should generate different characters.

To experiment, reverse the connector and see if it generates a more logical set of characters. Ultimately, the program in the controller will have to 'map' these characters to the ones marked on the keypad, which will likely be different.

2 Connections

2.1 Connector Pinout

Refer to the diagram below for this chapter.

Figure 2: Electrical Connections

The display has four connectors;

Table 2: Connectors & Functions

Connector	Function
14 pin dual header	General purpose outputs (6)
4 pin	Power (5.0 VDC) and I^2C communications
10 pin header	Keypad
DB-9F	RS-232 / Power

2.1.1 Power and I^2C Connections

Power is applied via pins 1 and 4. Power requirement is +5 VDC $\pm$0.25V. Power may also be supplied via the RS-232 connector as described in the next section.

WARNINGS

- Do not apply any power with reversed polarity.
- Do not apply any voltage other than the specified voltage.
- Do not use any cables other than the cables supplied by Matrix Orbital, unless aware of the modifications required.
- Do not apply voltage to the DB-9 connector AND power connector
- Do not apply more than +5Vdc to pin #9 on the DB-9 connector

Connector pinout is as follows;

Figure 3: Power Connector

Table 3: Connector Pinout

Pin 4	Ground
Pin 3	SDA (I^2C data)
Pin 2	SCL (I^2C clock)
Pin 1	+5.0 VDC (+7 to +15 VDC with wide voltage option)

2.1.2 Five Volt Modules

If the display is used in a PC it becomes tempting to plug a spare power connector into the unit. **Don't do this!** Wiring for the PC power connector and that required for the display are different as shown in the Figure below.
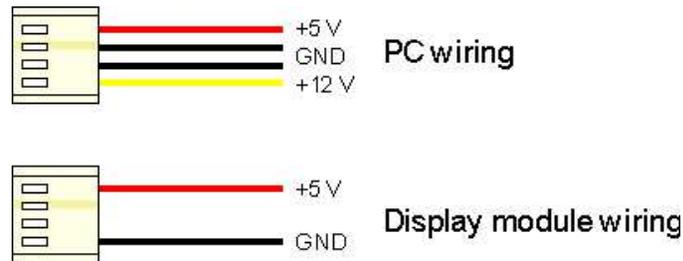
Figure 4: Wiring for 5V Modules

Matrix Orbital can supply an adapter cable designed for use with the display when it's installed in a PC. The cable is wired as shown in the Figure below.

> **NOTE**   This connector does not provide connections for I^2C.

Figure 5: 5V Power Cable

2.1.3 Wide Voltage Range Modules

> **NOTE**   Do not use this cable unless the display module has the "Wide voltage range" option (option V). Use of the 12 volt power cable with 5 volt modules will damage the module.

The 12 volt power cable is designed for use with wide voltage range display modules mounted in a PC. Wiring required for the 12 volt power connector is shown in the Figure below.
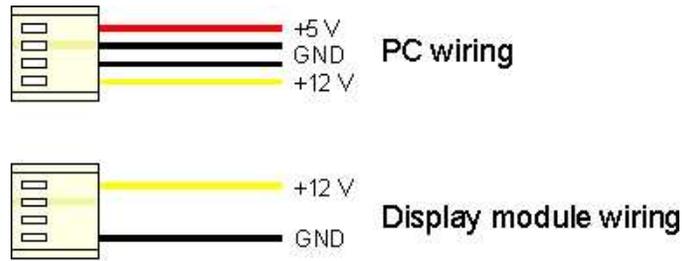
Figure 6: Wiring for 12V Modules

Matrix Orbital can supply an adapter cable designed for use with the display module when it's installed in a PC. The cable is wired as shown in the Figure below.

Figure 7: 12V Power Cable

2.1.4 RS-232 Communications

A standard DB-9F is provided for RS-232 communications. Power may also be supplied via this connector if desired.
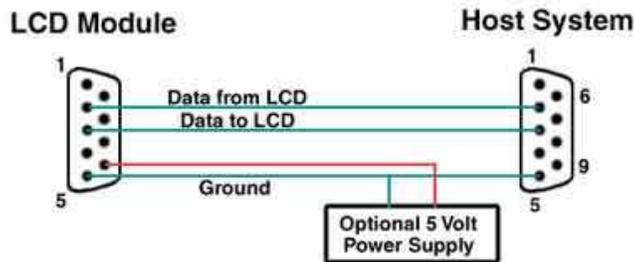
Figure 8: RS-232 and Power Connector

The RS-232 connector on the PC cable is wired so that a standard 'straight through' 9 pin D-sub cable may be used to connect the module to a standard serial port such as COM ports on PCs. Note that this device

complies with the EIA-232 standard in that it uses signal levels from +/- 3V to +/- 12V. It will not operate correctly at TTL (0 to +5V) levels.

Table 4: RS-232 Pinout

Pin Number	Direction	Description	VFD	Host
2	Data from VFD	Data out	Tx	Rx
3	Data to VFD	Data in	Rx	Tx
5	-	Ground	gnd	gnd

2.1.5 Configuring RS-232 and I^{2}C

RS-232 baud rate and I^{2}C address are configured by means of jumpers.

Figure 9: RS-232 Jumpers

The module is supplied with jumpers J1 and J2 installed, which gives an RS-232 baud rate of 19200 and an I^{2}C address of 0x5C.

- RS-232 port: J0, J1, J2 - control baud rate. RS-232 format is 8N1 (8 bits, no parity, one stop bit)

- I^{2}C port: J0, J1, J2, J3 - sets slave peripheral address

Table 5: Serial Baud Rates

Baud Rate	Slave Address	J3	J2	J1	J0
1200	50H	out	out	out	out
	52H	out	out	out	in
2400	54H	out	out	in	out
	56H	out	out	in	in
9600	58H	out	in	out	out
	5AH	out	in	out	in
19200	5CH	out	in	in	out
	5EH	out	in	in	in
1200	60H	in	out	out	out
	62H	in	out	out	in
2400	64H	in	out	in	out
	66H	in	out	in	in
9600	68H	in	in	out	out
	6AH	in	in	out	in
19200	6CH	in	in	in	out
	6EH	in	in	in	in

2.2   I²C Communications

I^{2}C communications runs at 100 Kbps and supports up to 16 units on a single communications line. The I^{2}C data line operates on 5 volt CMOS levels.

2.2.1   ACK

The idea of ACK is to indicate when the data has been received correctly. ACK does not indicate data incorrectly received. ACK simply fails to indicate when data is correctly received. Clearly, this is of limited usefulness and even less so with Matrix Orbital modules. Matrix orbital modules are not capable of failing to acknowledge an incorrectly received byte in response to that bytes transition. They are only capable of failing to acknowledge the bytes following the byte, which was not received. To fully understand the reasons for this one needs to understand something about how a Matrix Orbital module processes data. Basically the reason why a Matrix Orbital module might fail to receive a byte correctly is that it was unable to process the byte previous before the failed byte was transmitted. Because the module cannot possibly know that it would be unable to store the byte before the next byte was received it cannot know to not ACK. The reason for this situation in deference to situations one might be familiar with (i.e., memory chips, etc) is that the Matrix Orbital module employs a micro-processor to perform these data storage functions. A memory chip takes care of these things entirely within hardware subsystems that operate at the same speed as the transmission themselves.

The display uses a standard Phillips 7bit address as defined by Phillips. However, Matrix Orbital specifies I^{2}C address in 8bits. The 8th bit, least significant bit (LSB or Low Order Bit) of the 8bit address is read

/ write bit. If we take a standard Phillips 7bit address of 45hex this would be in binary 1000101. This is 7bits. Matrix Orbital would describe the Phillips I^{2}C address of 45hex as 8Ahex. The read address would be 8Bhex.

For more information on Phillips I^{2}C please visit;

`http://www.ping.be/~ping0751/i2cfaq/i2cindex.htm`

2.3 General Purpose Outputs

The display has six general purpose outputs. These are provided to control relays or other electronic devices. This allows external devices to be turned on or off using the PC or controller and software commands.

Each output is wired as shown in the Figure below. The + terminal is connected directly to the module positive supply, the - terminal is connected through a 240 ohm current limiting resistor and the electronic switch to ground.
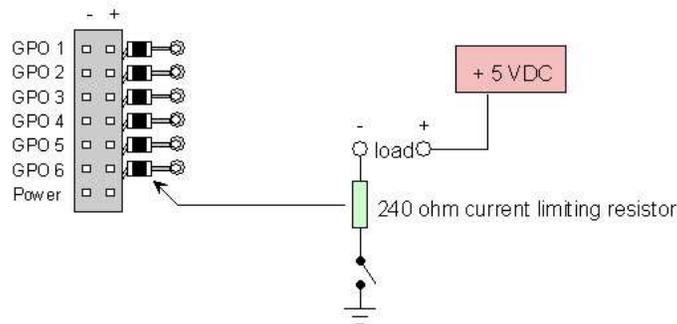
Figure 10: General Purpose Outputs

Maximum allowable current is 20 mA, which is enforced by the current limiting resistor. If the device being switched has a resistance of 240 ohms or more the corresponding resistor may be shorted. Solder a small jumper wire (wirewrap wire is good) from the feedthrough hole to the corresponding negative pin for the GPO in question.
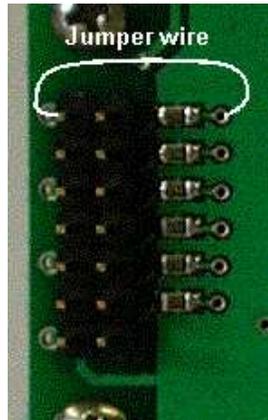
Figure 11: Bypassing 240 ohm Resistor

NOTE   The GPOs do not have any over current or over / under voltage protection so care must be taken when using them. For instance if the external device is a relay it must be fully clamped using a diode and capacitor to absorb any generated back electro-motive force (EMF).
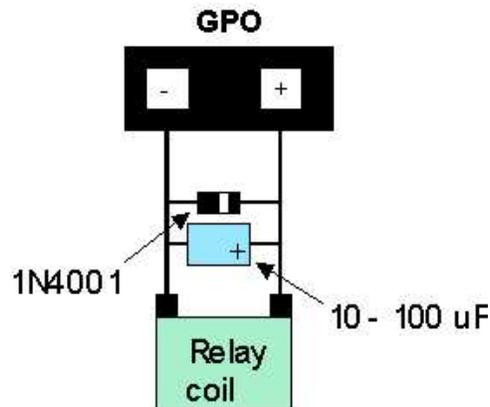
Figure 12: Clamping a Relay

3   Displaying Text

This chapter describes the various text display commands in detail.

3.1 General

Text is displayed on the VK204-25 using the built in 5x7 dot matrix font, in addition to up to 8 user defined characters.

3.2 The Built In Character Font

The display includes a built in 5x7 dot matrix font with the full range of ASCII characters plus a variety of extended characters, as shown in the Figure below.

Figure 13: Character Set

In addition to the built in characters, users may define up to 8 special characters which, once defined, occupy positions 0x00 to 0x07 in the above chart. The display does not have provision to download other

fonts.

3.3 Writing Text to the Display

When the display receives a character, it displays that character at the position currently defined. The next character sent to the module then advances to the following position on the display. Characters are drawn using the built in font, and only characters defined in the font are actually displayed. Characters that are not defined by the built in font print as a space (i.e., the cursor is advanced for the next character). The position where text is to be inserted is a character location stored in the display's volatile memory and maintained internally by the display's firmware. This position is manipulated by the commands shown in the following section.

3.4 Text Commands

In this section commands are identified by their names and decimal values.

3.4.1 Auto line wrap on (254 67)

Enables automatic line wrapping. Note that this is not 'word wrapping' and wraps may occur in the middle of a word.

3.4.2 Auto line wrap off (254 68)

Disables automatic line wrapping. Characters beyond the end of a line will be lost.

3.4.3 Auto scroll on (254 81)

When auto scrolling is on, it causes the display to shift the entire display's contents up to make room for a new line of text when the text reaches the scroll position (the bottom right character position).

3.4.4 Auto scroll off (254 82)

When auto scrolling is disabled, text will wrap to the top left corner of the display area. Existing text in the display area is not erased before new text is placed. A series of 'spaces' followed by a "Cursor home" command may be used to erase the top line of text.

3.4.5   Set cursor position (254 71 [column] [row])

This command sets the cursor position (text insertion point) to the [column] and [row] specified. Columns have values from 1 to 20 (0x01 to 0x14) and rows have values of 1 and 2 (0x01 and 0x02).

3.4.6   Send cursor home (254 72)

This command moves the cursor position (text insertion point) to the top left of the display.

3.4.7   Turn on block (blinking) cursor (254 83)

Turns on the blinking block cursor. The cursor shows the current text insertion point. Both blinking and underline cursors may be turned on or off independently. The cursor is off by default

3.4.8   Turn off block (blinking) cursor (254 84)

Turns off the blinking block cursor. Does not affect the underline cursor.

3.4.9   Cursor left (254 76)

Moves the cursor one position to the left but does not erase any character that may be in that position. Note that this command moves the text insertion point even if the cursor is turned off.

> **NOTE**   A 'destructive backspace', which erases the character to the left of the original position, may be done by issuing the following sequence: cursor left, space, cursor left.

3.4.10   Cursor right (254 77)

Moves the cursor one position to the right but does not erase any character that may be in that position. Note that this command moves the text insertion point even if the cursor is turned off.

4   Keypad Interface

This chapter describes the keypad interface and associated commands in detail.

4.1 General

The display keypad interface processes the keypad row / column matrix into a serial (RS-232 or I^{2}C) data byte stream. Aside from this processing, the keypad has no effect on the display. To send keystrokes to the display, they must be routed through the controller.

4.2 Connections
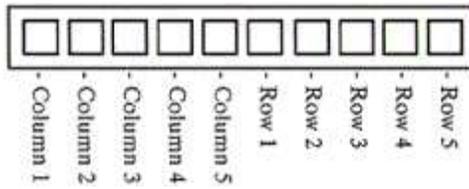
Figure 14: Keypad Connector

The connector is not 'keyed' so the keypad will probably plug in either of two ways. The display will not be damaged by reversing the connector, but the keypad will generate a different ASCII character mapping for each position. If the connector has fewer than 10 pins it should be centered on the display connector.

The diagram shows the logical layout (row 1, column 1 in upper left). The connector for the keypad is a 10 pin 0.1" spacing male header. Pins 1 through 5 are columns and pins 6 through 10 are rows. The keypad is scanned whenever a key is pressed: there is no continuous key scan. This means that key presses are dealt with immediately without any appreciable latency. This also prevents electrical noise which is often caused by continuous key scans.

NOTE The keypads may be laid out in a different pattern. If this is the case, the user will need to interpret the key codes differently.

Table 6: Keypad Layout

		Columns				
		1	2	3	4	5
	1	A	B	C	D	E
Rows	2	F	G	H	I	J
	3	K	L	M	N	O
	4	P	Q	R	S	T
	5	U	V	W	X	Y

> **NOTE** The keypad connector must be wired with columns on one side and rows on the other side of the center of the connector. In situations where the keypad isn't wired this way an adapter will need to be made, or the user should rewire the connector to meet this requirement.

4.3   I²C Interface

The keypad is read by I²C master read. In short, this means that a read of the module will always return the first unread key press. A read is initiated by writing to the module with its base address plus 1, then clocking the module's return byte after the module releases the SDA line. Much more detail on this basic I²C function can be found in the I²C specification by Phillips.

A good reference is also available at;

`http://www.ping.be/~ping0751/i2cfaq/i2cindex.htm`

The module contains a ten key press buffer so that it can be polled for key presses at an infrequent rate (every 500 to 1000 mS is typical). All returned key presses indicate the presence or absence of additional logged key presses by the most significant bit (MSB - bit 7). If the user has pressed two keys since the last poll of the keypad interface, the first read will return the key code with bit 7 set and the second read will return the key code with bit 7 clear. The application must take into account this bit to keep up with user key presses. If there are no keypresses detected, the module will return zero (0x00).

4.4   RS-232 Interface

By default on any press of a key, the module will immediately send out the key code at the selected baud rate. This behavior can be modified using commands found in the next section.

4.5   Keypad Commands

4.5.1   Auto repeat mode on (254 126 [mode])

[mode] = 0x00 gives Resend Key Code mode
[mode] = 0x01 gives Key Down / Key Up Code mode
Two Modes of auto repeat are available and are set via the same command.

- **Resend Key Code:** This mode is similar to the action of a keyboard on a PC. In this mode, when a key is held down, the key code is transmitted immediately followed by a 1/2 second delay. After this delay, key codes will be sent via the RS-232 interface at a rate of about 5 codes per second. This mode has no effect if polling or if using the I²C interface.
- **Key Down / Key Up Codes:** This mode may be used when the typematic parameters of the "Resend key code" mode are unacceptable or if the unit is being operated in polled mode. The host system detects the press of a key and simulates an auto repeat inside the host system until the key release is detected.

In this mode, when a key is held down, the key code is transmitted immediately and no other codes will be sent until the key is released. On the release of the key, the key release code transmitted will be a value equal to the key down code plus 20 hex. For example, the key code associated with key 'P' (0x50) is pressed, the release code is 'p' (0x70). In RS-232 polled mode or via the I^{2}C interface, the "Key down / key up codes" are used. However, the user should be careful of timing details. If the poll rate is slower than the simulated auto-repeat it is possible that polling for a key up code will be delayed long enough for an unwanted key repeat to be generated.
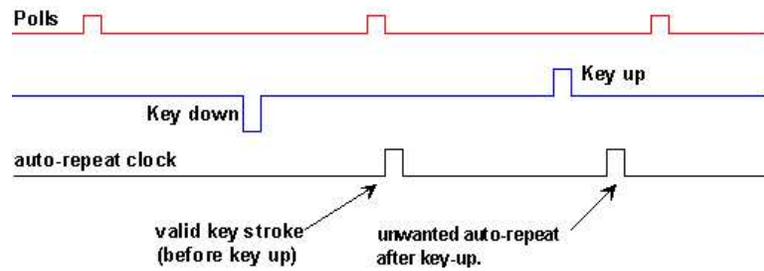
Figure 15: Poll Timing

4.5.2   Auto repeat mode off (254 96)

This command turns off auto repeat mode.

4.5.3   Auto transmit keypresses on (254 65)

In this mode, all keypresses are sent immediately to the host system without the use of the poll keypad command. This is the default mode on power up.

4.5.4   Auto transmit keypresses off (254 79)

In this mode, up to 10 keypresses are buffered until the unit is polled by the host system via the poll keypad command. Issuing this command places the unit in polled mode.

4.5.5   Clear key buffer (254 69)

This command clears any unread keypresses. In a menuing application, if the user presses a key which changes the menu context, any following key presses may be inaccurate and can be cleared out of the buffer between menu changes to prevent jumping around the menu tree. It may also be used to, in effect, reset the keypad in case the host application resets for whatever reason.

4.5.6 Poll keypad (254 38)

This command returns any unbuffered keypresses via the RS-232 interface. The host system must be set up to receive the key codes. When the display receives this command it will immediately return any unbuffered keypresses which may have not been read already. If there is more than one keypress buffered, then the high order bit (MSB) of this returned keycode will be set (1). If this is the only buffered keypress, then the MSB will be reset (0). If there are no buffered keypresses, then the returned code will be 0x00. Please note to make use of this command, the "Auto transmit keypress" mode should be off.

4.5.7 Set debounce time (254 85 [time])

[time] is in increments of 6.554 milliseconds.

This command sets the time between key press and key read. All key types with the exception of latched piezo switches will 'bounce' for a varying time, depending on their physical characteristics. The default debounce time for the module is about 52 mS, which is adequate for most membrane keypads.

5   Bar Graphs and Special Characters

The display includes the ability to draw bar graphs (either horizontal or vertical) and allows users to define up to eight special characters.

Eight characters (ASCII values 0x00 to 0x07) are set aside for use with bar graphs, user defined characters, and big numbers. Since the same 8 characters are used for each function, the functions may not be used simultaneously. The characters may be defined or redefined at any time by issuing the commands shown in this section. Once defined, they may be used either by means of the bar graph commands, or by simply issuing one of the ASCII values 0x00 to 0x07 (which is not prefixed by the command byte, 254).

5.1 Command List

5.1.1 Initialize wide vertical bar graph (254 118)

This command defines the 8 special / user characters to be blocks suitable for use in drawing wide (5 pixel) vertical bar graphs. Any previously existing definitions will be lost. Once this command has been issued, any number of vertical bar graphs may be drawn unless the characters are redefined by another command.

5.1.2 Initialize narrow vertical bar graph (254 115)

This command defines the 8 special / user characters to be blocks suitable for use in drawing narrow (2 pixel) vertical bar graphs. Any previously existing definitions will be lost. Once this command has been issued, any number of vertical bar graphs may be drawn unless the characters are redefined by another command

5.1.3 Draw vertical bar graph (254 61 [column] [height])

Draws a vertical bar graph in [column] having a height of [height] pixels. The height may range from 0 to 20 (0x00 to 0x14) pixels. The necessary characters must first be initialized by either of the commands shown in section 5.1.1 or 5.1.2, which will determine the width of the graph drawn. Graph may be erased by drawing a bar graph of height = 0 in the same column.

5.1.4 Initialize horizontal bar graph (254 104)

This command defines the 8 special / user characters to be blocks suitable for use in drawing horizontal bar graphs. Any previously existing definitions will be lost. Once this command has been issued, any number of horizontal bar graphs may be drawn unless the characters are redefined by another command.

5.1.5 Draw horizontal bar graph (254 124 [column] [row] [dir] [length])

Draws a horizontal bar graph in [row] starting at [column] with a length of [length] pixels. [row] may have a value of 0x01 or 0x02, column may range from 0x01 to 0x14 and length may be from 0x00 to 0x64 (0 to 100) if the graph can extend the full width of the screen. Each column is 5 pixels wide (spaces between the columns don't count).

[dir] specifies the direction: 0x00 goes from left to right, 0x01 goes from right to left.

5.1.6 Define custom character (254 78 [c] [8 bytes])

The display allows up to 8 user defined (custom) characters. These characters occupy the first 8 (0x00 to 0x07) places in the character set.

Custom characters occupy a 5x8 pixel matrix. Built in characters are 5x7: the bottom row of pixels is normally reserved for the underline cursor. The underline cursor should be turned off if the bottom row of pixels forms part of a custom character.

The characters are defined by issuing the command 254 78 [c] followed by 8 bytes to define the character. [c] is the character number (0x00 to 0x07). The 8 bytes are mapped as shown below;

Table 7: 8 Byte Map

MSB LSB								
*	*	*	1	2	3	4	5	Data Byte 1
*	*	*	6	7	8	9	10	Data Byte 2
*	*	*	11	12	13	14	15	Data Byte 3
*	*	*	16	17	18	19	20	Data Byte 4
*	*	*	21	22	23	24	25	Data Byte 5
*	*	*	26	27	28	29	30	Data Byte 6
*	*	*	31	32	33	34	35	Data Byte 7
*	*	*	36	37	38	39	40	Data Byte 8

A "1" bit indicates an on (black) pixel, a "0" bit indicates an off (clear) pixel.

Once defined, a character is displayed simply by issuing a value (0x00 to 0x07) corresponding to the character number. The character will be laid out as follows;

Table 8: Character Values

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35
36	37	38	39	40
Cursor Line				

NOTE   Custom characters will be erased if any of the "Initialize bar graph" commands are issued.

5.1.7   Initialize large digits (254 110)

This command defines the 8 special / user characters to be blocks suitable for use in drawing large digits. Any previously existing definitions will be lost. Once this command has been issued, any number of large characters may be placed until the characters are redefined by another command.

5.1.8   Place large digit 254 35 [col] [digit]

This command allows the large digits to be drawn on the LCD screen. Numbers of almost full display height may be placed along side regular text on four row displays. The column number has a maximum value which is less than the display width because the digits are all three columns wide.

Before using this command the "Initialize large digits" command must be issued to define the blocks necessary to make up the digits. If regular text and large digits are mixed on one screen, the user should always set the display cursor position before placing regular text because the creation of a large digit will leave the cursor position to the bottom right of the large digit and not at the last regular text write position.

[col] can have values from 0x01 to 0x12 (1 to 18).

[digit] has values from 0x00 to 0x09 (0 to 9).

6   Miscellaneous Commands

The commands listed in this chapter don't readily fit in any of the other categories, or are used in more than one category.

6.1   Command List

6.1.1   Clear display (254 88)

This command clears the display and resets the text insertion point to the top left of the screen.

6.1.2   Set Brightness (254 89 [Brightness])

This command sets the display's brightness to [brightness], where [brightness] is a value between 0x00 and 0x03 (between 0 and 3) according to the table below.

Table 9: Brightness Settings

Hex Values	Brightness
0x00	100%
0x01	75%
0x02	50%
0x03	25%

6.1.3 Display on (254 66[minutes])

This command turns on the display for a time of [minutes] minutes. If [minutes] is zero (0), the display will remain on indefinitely.

NOTE	The factory default for display is on.

6.1.4 Display off (254 70)

This command turns the display off.

6.1.5 General purpose output off (254 86 [gpo #])

This command turns OFF any of the general purpose outputs. [gpo #] is 1 to 6.

NOTE	OFF means that the output floats.

6.1.6 General purpose output on (254 87 [gpo #])

This command turns ON any of the general purpose outputs. [gpo #] is 1 to 6. ON means that the output is pulled low (ground via 240 ohms).

6.1.7 Read module type (254 55)

This command will return, over the RS-232 interface, the 1-byte model type value of the module. This command will return a 1-byte hex value. Values for various modules at the time of this publication are as follows;

Table 10: Module Values

LCD0821 - 0x01	LCD2021 - 0x03	LCD2041 - 0x05
LCD4021 - 0x06	LCD4041 - 0x07	LK202-25 - 0x08
LK204-25 - 0x09	LK404-55 - 0x0A	VFD2021 - 0x0B
VFD2041 - 0x0C	VFD4041 - 0x0D	VK202-25 - 0x0E
VK204-25 - 0x0F	GLC12232 - 0x10	GLC24064 - 0x13
GLK24064-25 - 0x15	GLK12232-25 - 0x22	LK404-AT - 0x31
LK402-12 - 0x33	LK162-12 - 0x34	VK204-25PC - 0x35

6.1.8 Read Version Number (254 54)

This command will return the firmware version number of the display. The command will return a 1-byte hex value.

7 Appendix: Command Summary

7.1 General

The operation of the display is controlled by a simple and consistent command set. Commands control;

- Text display
- Graphics display
- Keypad interface
- Miscellaneous operating parameters

This chapter includes summary tables of all commands.

7.2 Issuing Commands

Commands are issued to the display by the controller. In a test setup, commands can be issued to the display by means of a BASIC program, using the chr$( ) function. In the tables below, we've shown commands in hex, ASCII and decimal form. All commands begin with the prefix character 0xFE (254 decimal). These commands are issued on the serial communications link (I^2C or RS-232) at the currently defined baud rate.

For example (using BASIC in a test setup), the user could issue the command to clear the screen on the display by including the line;

```
PRINT#1,chr$(254);chr$(88)
```

in the BASIC program.

Or, with C the user could (using Zcomm serial library)

```
ZComm1->WriteCommByte(0xfe);
ZComm1->WriteCommByte('X');
```

7.3 On Numbers

Like all computerized devices, the display operates with commands and values in the form of binary numbers. These binary numbers are arranged in 8 digit (i.e., 8 bit) groups called bytes. The decimal value of a byte may have any value from 0 to 255.

Bytes are usually specified in either decimal or hexadecimal (base 16) form for convenience, since binary numbers are confusing to deal with directly. Hexadecimal (hex) numbers are particularly convenient because

---

exactly two hexadecimal digits make up one byte, each hex digit representing 4 binary digits (4 bits) as shown here;

Table 11: Hex Value Table

Binary	Hex	Decimal	Binary	Hex	Decimal
0000	0	0	1000	8	8
0001	1	1	1001	9	9
0010	2	2	1010	A	10
0011	3	3	1011	B	11
0100	4	4	1100	C	12
0101	5	5	1101	D	13
0110	6	6	1110	E	14
0111	7	7	1111	F	15

Based on the table, the byte 01001011 can be represented in hex as 4B, which is usually written as any of 4Bh, 4BH, 4B hex or 0x4B.

The numbers can also be expressed in decimal form if preferred.

7.3.1 ASCII Characters

Since computers deal internally with numbers only, but externally with both letters and numbers, several schemes were developed to 'map' written characters to numeric values. One such scheme has become universal, the American Standard Code for Information Interchange, or ASCII. ASCII tables are readily available from a number of sources. A few examples will do here;

Table 12: Example of an ASCII Table

The letter	A	Has a value of	65 Decimal or	41 Hex
The letter	A	Has a value of	97 Decimal or	61 Hex
The number	0	Has a value of	48 Decimal or	30 Hex
The number	9	Has a value of	57 Decimal or	39 Hex

This gives rise to the possibility of confusion when parameters are being set on the display. For example, the GPO ON and OFF commands use a number to indicate which GPO is being controlled. We're told that acceptable values are 0 to 6. All such parameters must use numeric values (i.e., the actual byte values). If we send the ASCII number 0 by mistake it will actually give the value 48 decimal (30 hex) to the parameter, which is wrong.

In the tables given in the following sections ASCII characters are shown as 'A', with single quotes.

7.4 Text Commands

Syntax in the tables below is given in hex, decimal and decimal with ASCII, in that order, one per line.

Table 13: Text Commands

Command	Syntax	Default	Notes
Auto line wrap on	FE 43 254 67 254 'C'	On	Enables line wrapping (not word wrap)
Auto line wrap off	FE 44 254 68 254 'D'	On	Disables line wrapping
Auto scroll on	FE 51 254 81 254 'Q'	Off	Enables scroll at bottom of screen. Text will push display up one line to make room for new line
Auto scroll off	FE 52 254 82 254 'R'	Off	Disables auto scroll. Text will wrap to top left and overwrite existing text
Set cursor position	FE 47 [col] [row] 254 71 [col] [row] 254 'G' [col] [row]	N/A	Moves cursor to the specified column and row. The cursor marks the text insertion point in this and all commands
Send cursor home	FE 48 254 72 254 'H'		This command moves the cursor to the top left of the display area
Block cursor on	FE 53 254 83 254 'S'	On	Turns on the blinking block cursor
Block cursor off	FE 54 254 84 254 'T'		Turns off the blinking block cursor

Command	Syntax	Default	Notes
Cursor left	FE 4C 254 76 254 'L'		Moves the cursor one position to the left. If the cursor is already at the beginning of a line it will move to the end of the other line
Cursor right	FE 4D 254 77 254 'M'		Moves the cursor one position to the right. If the cursor is already at the end of a line it will move to the beginning of the other line

7.5 Keypad Interface Commands

NOTE   The letter "R" in the "Default" column indicates that this state can be saved to nonvolatile memory with the "Remember" command.

Table 15: Keypad Interface Commands

Command	Syntax	Default	Notes
Auto repeat mode on	FE 7E [0x00|0x01] 254 126 [0|1] 254 '~' [0|1]	Off	Applies to keypad only.0x00 = 200 ms typematic,0x01 = key down/key up codes sent.
Auto repeat mode off	FE 60 254 96 254 '`'	Off	Applies to keypad only.

Command	Syntax	Default	Notes
Auto transmit key presses on	FE 41 254 65 254 'A'	On	Sets auto transmit mode for keypad. Key presses are transmitted to host without polling.
Auto transmit key presses off	FE 4F 254 79 254 'O'	Off	Up to 10 key presses buffered until polled.
Clear key buffer	FE 45 254 69 254 'E'	N/A	Clear unread key presses.
Poll keypad	FE 26 254 38 254 '&'	N/A	Returns buffered key presses to application. Returns 0x00 if no key presses. High order bit set unless this is the last/only key press.
Set debounce time	FE 55 [time] 254 85 [time] 2 54 'U' [time]	52ms	Resolution: 1 = 0.6554 ms [time] is a numeric multiplier.

7.6 Bar Graphs and Special Characters

The commands in this section are used to define and display bar graphs and special characters.

Table 17: Bar Graphs and Special Characters

Command	Syntax	Notes
Initialize thick vertical bar graph	FE 76 254 118 254 'v'	Initializes the user character set to make wide vertical bar graphs.
Initialize thin vertical bar graph	FE 73 254 115 254 's'	Initializes the user character set to make narrow vertical bar graphs.

Command	Syntax	Notes
Initialize horizontal bar graph	FE 68 254 104 254 'h'	Initializes the user character set to make horizontal bar graphs.
Define custom character	FE 4E [c][8 bytes] 254 78 [c][8 bytes] 254 'N' [c][8 bytes]	Defines one of 8 custom "user" characters. Character number is [c] between 0x00 and 0x07.
Draw vertical bar graph	FE 3D [col][length] 254 61 [col][length] 254 '=' [col][length]	Draws a vertical bar graph at column [col] of length [length]. Length is measured in pixels (0x00 to 0x14). User must first use the 'v' or 's' command to initialize characters.
Draw horizontal bar graph	FE 7C [c][r][d][length] 254 124 [c][r][d][length] 254 'l' [c][r][d][length]	Draws a horizontal bar graph starting at column [c] on row [r] with direction [d] (0 is right, 1 is left) of length [length]. Length is measured in pixels (0x00 to 0x64 if starting in column 1). User must first use the 'h' command to initialize characters.
Initialize large digits	FE 6E 254 110 254 'n'	Initializes the user character set to make large digits.
Place large digits	FE 23 [col] [digit] 254 23 [col] [digit] 254 '#' [col] [digit]	Place large digit number [digit] in column [col] of the display. Cursor moves to bottom right of large digit. [digit] is 0x00 to 0x09, [col] is 0x01 to 0x12 (i.e. 1 to 18 decimal).

7.7 Miscellaneous Commands

Table 19: Miscellaneous Commands

Command	Syntax	Default	Notes
Clear display	FE 58 254 88 254 'X'	N/A	Clears screen of text and graphics, places text cursor at top left.

Command	Syntax	Default	Notes
Set brightness	FE 59 [brightness] 254 89 [brightness] 254 'Y' [brightness]	0x00 0x00	Sets display brightness. Compensates for viewing angle. brightness is a value between 0x00 and 0x03. 0x00=100%, 0x01=75%, 0x02=50%, 0x03=25%.
Display on	FE 42 [minutes] 254 66 [minutes] 254 'B' [minutes]	On	Display will stay on for [minutes]. If [minutes] = 0 display will stay on permanently. Maximum value for [minutes] is 100.
Display off	FE 46 254 70 254 'F'	On	Turns the display off.
General purpose output off	FE 56 [gpo #] 254 86 [gpo #] 254 'V' [gpo #]		Turns a general purpose output OFF. [gpo #] may be from 1 to 6.
General purpose output on	FE 57 [gpo #] 254 87 [gpo #] 254 'W' [gpo #]		Turns a general purpose output ON. [gpo #] may be from 1 to 6.
Read module type	FE 37 254 55 254 '7'	see table	Reads the module type. This command will return a 1-byte hex value.
Read Version Number	FE 36 254 54 254 '6'		Reads the firmware version number of the module. This command will return a 1-byte hex value.

8   Appendix: Specifications and Options

8.1   Specifications

Table 21: Environmental Specifications

	Standard Temperature	Extended Temperature
Operating Temperature	-20°C to +70°C	-40°C to +85°C
Storage Temperature	-40°C to +80°C	-50°C to +85°C
Operating Relative Humidity	20 to 80% non condensing	
Vibration (non-operating)	10 to 55 to 10 Hz (Frequency)	
	1.0 mm (Total Amplitudes)	
	30 minutes (Duration)	
	X, Y, Z each direction	
Shock (Non-operation)	593 m/S^2, 10 mS	

Table 22: Electrical Specifications

Supply Voltage	4.75 - 5.25 Vdc (optional 7- 15 and 7 - 30VDC)
Supply Current	220 mA typical
Supply Current (power up)	Up to 550mA inrush

Table 23: Optical Characteristics

Number of Characters	40 (20 characters by 2 lines)
Matrix format	5 x 7 with underline
Display Area	70.8 x 20.9mm XxY
Character Size	2.4 x 4.7mm (XxY)
Character Pitch	3.6 x 5.4 mm (XxY)
Line pitch	5.95 mm
Dot Size	0.4 x 0.5 mm (XxY)
Dot Pitch	0.5 x 0.7 mm (XxY)
Luminance	Min. 350cd/m^2 (100fL) min
Color of Illumination	Blue/Green
Display Life	100,000 hours typical

WARNINGS

- Heavy shock of more then 100G, thermal shock greater then 10°C/minute, direct hit with a hard material to the glass surface (especially to the exhaust pipe) may CRACK the glass.
- Steady repetition of a static message displaying, longer then 5 hours in a day may cause a phosphor burnout problem.

Figure 16: Physical Layout

8.2 Options

Table 24: Options Available on VK204-25

Extended Temperature	E
Wide Voltage	V
Wide Voltage with Efficient Switching Power Supply	VPT

9 Appendix: Glossary

Table 25: Appendix: Glossary

ASCII	American Standard Code for Information Interchange. A 7 bit binary code representing the English alphabet, decimal numbers and common punctuation marks. Also includes control characters such as carriage return or end of text. An 8 bit superset of the standard ASCII codes is often used today to include foreign characters and other symbols. These supersets are often called extended ASCII character sets
Backlight	A backlit display is illuminated from behind to provide nighttime and improved daytime readability
Binary Number	The (data and signaling) bit transmission rate of an RS232 device
Bit	A number written using binary notation which only uses zeros and ones
Bitmap	A representation, consisting of rows and columns of dots, of a graphics image in computer memory. The value of each dot (whether it is filled in or not) is stored in one or more bits of data
Byte	A grouping of eight binary bits
CCFL	Cold Cathode Fluorescent Lamp. A high brightness backlighting source consists of a fluorescent tube powered by a high voltage A.C. source
Configuration	The way a system is set up, or the assortment of components that make up the system. Configuration can refer to either hardware or software, or the combination of both
Contrast	The ratio of luminance between the light state of the display to the dark state of the display

Controller	The micro-controller or PC used to control the Matrix Orbital display unit
DB-9	The designation of a connector used in the RS232 interface: 9 pin connector
Firmware	Software (programs or data) that has been written onto read-only memory (ROM). Firmware is a combination of software and hardware. ROMs, PROMs and EPROMs and flash EEPROMs that have data or programs recorded on them are firmware
Font	A design for a set of characters. A font is the combination of typeface and other qualities, such as size, pitch, and spacing
Font Metric	A definition of where font is to be placed, such as margins and spacing between characters and lines
Hexadecimal	Refers to the base-16 number system, which consists of 16 unique symbols: the numbers 0 to 9 and the letters A to F. For example, the decimal number 15 is represented as F in the hexadecimal numbering system. The hexadecimal system is useful because it can represent every byte (8 bits) as two consecutive hexadecimal digits. It is easier for humans to read hexadecimal numbers than binary numbers
I^{2}C	Short for Inter-IC, a type of bus designed by Phillips Semiconductors in the early 1980s, which is used to connect integrated circuits (ICs). I^{2}C is a multi-master bus, which means that multiple chips can be connected to the same bus and each one can act as a master by initiating a data transfer
Interface	A means by which two systems interact
LCD	Liquid Crystal Display
Module Type Value	This refers to the model number of the module
Pixel	The smallest individually controllable element of a display
Pre-Generated Fonts	Pre-determined fonts which can be downloaded into graphic liquid crystal displays
Primitive	A low-level object or operation from which higher-level, more complex objects and operations can be constructed. In graphics, primitives are basic elements, such as lines, curves, and polygons, which you can combine to create more complex graphical images
RS-232	Short for recommended standard-232C, a standard interface approved by the Electronic Industries Association (EIA) for connecting serial devices

Scroll	To view consecutive lines of data on the display screen. The term scroll means that once the screen is full, each new line appears at the bottom edge of the screen and all other lines move up one position
Serial Number	A number that is one of a series and is used for identification of the module
Serial Port	A port, or interface, that can be used for serial communication, in which only 1 bit is transmitted at a time
Version Number	This refers to the firmware revision number of the module
Volatile Memory	Temporary memory. Once the power supply is turned off volatile memory is then erased