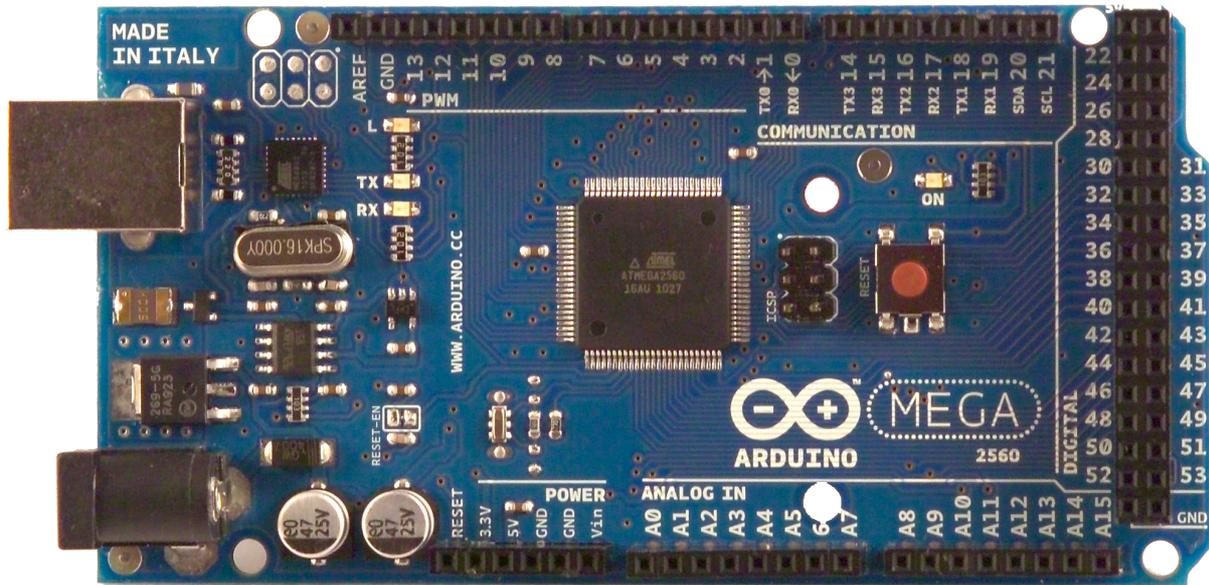


Arduino MEGA 2560



Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Index

Technical Specifications

Page 2

How to use Arduino
Programming Enviroment, Basic Tutorials

Page 6

Terms & Conditions

Page 7

Enviromental Policies
half sqm of green via Impatto Zero®

Page 7



RADIOSPARES

RADIONICS



Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I²C: 20 (SDA) and 21 (SCL).** Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.



radiospares

RADIONICS



Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega's digital pins.

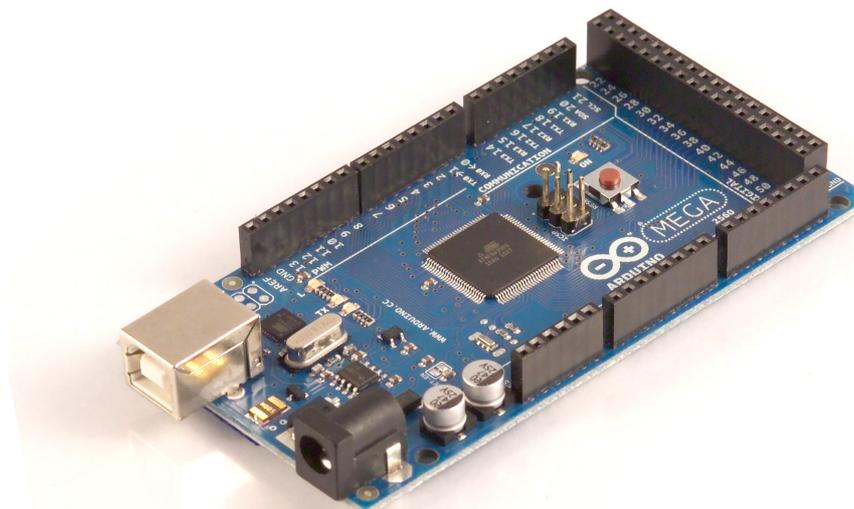
The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. To use the SPI communication, please see the ATmega2560 datasheet.

Programming

The Arduino Mega2560 can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.



radiospares

RADIONICS



Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. **Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).**



radiospares

RADIONICS



How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink**

Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board** select MEGA

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.

```
Blink | Arduino 0017
File Edit Sketch Tools Help
Blink $
int ledPin = 13; // LED connected to digital pin 13

// The setup() method runs once, when the sketch starts

void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}

// the loop() method runs over and over again,
// as long as the Arduino has power

void loop()
{
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(1000); // wait for a second
  digitalWrite(ledPin, LOW); // set the LED off
  delay(1000); // wait for a second
}
```



Done compiling.

Press Compile button
(to check for errors)



Upload



TX RX Flashing



Blinking Led!

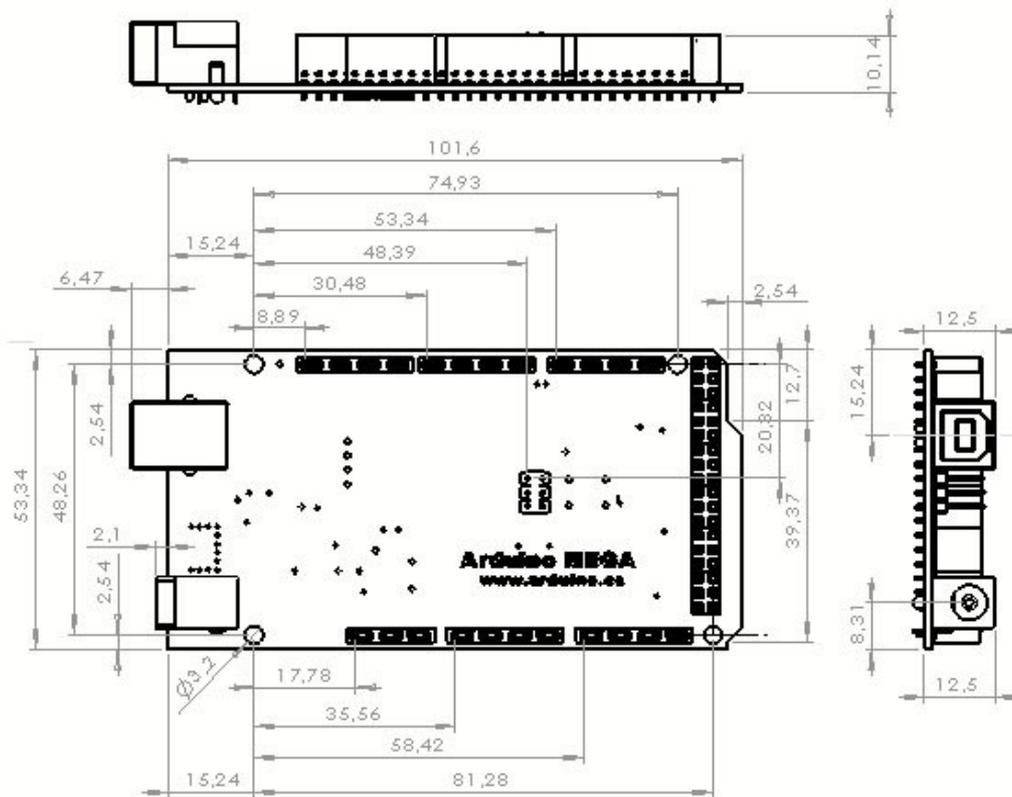
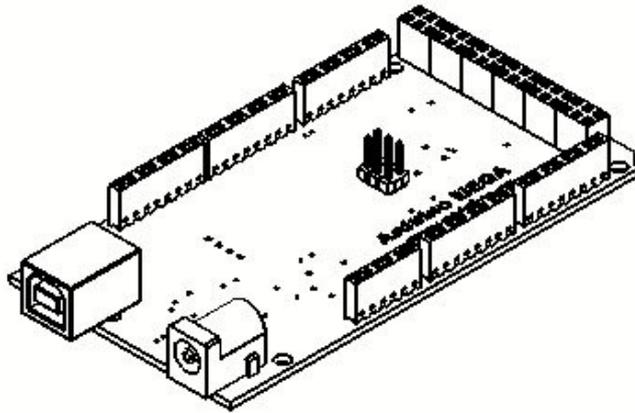


radiospares

RADIONICS



Dimensioned Drawing



radiospares

RADIONICS



Terms & Conditions



1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



Environmental Policies



The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.



radiospares

RADIONICS



Graphic LCD Serial Backpack Datasheet

2009.09.09

Figure 1

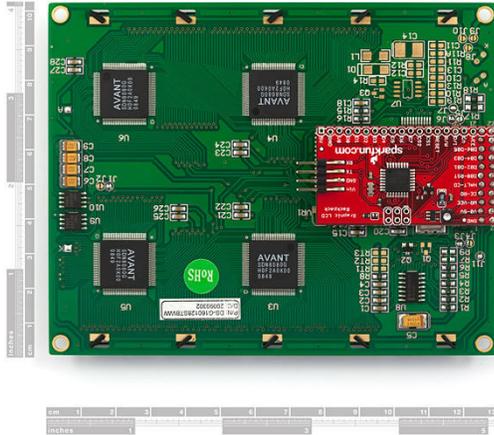
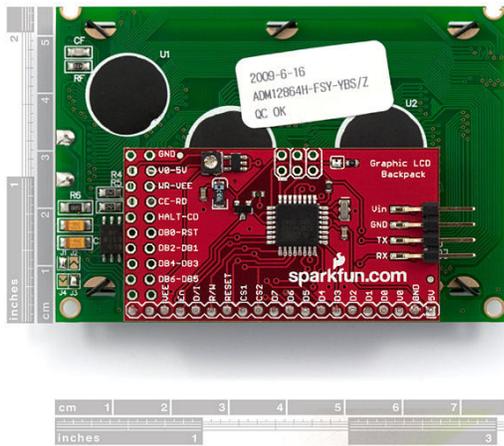


Figure 2



Overview

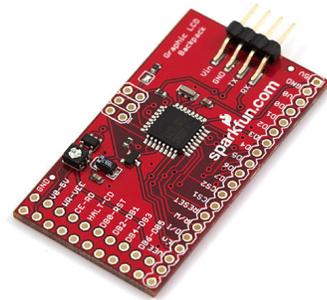
The Serial Graphic LCD backpack interfaces to 160x128 pixel "Huge" Graphic LCD (sku: LCD-08799) and provides the user a simple serial interface to a full range of controls. Besides writing text, the backpack allows the user to draw lines, circles and boxes, set or reset individual pixels, erase specific blocks of the display and control the backlight. There's also a reverse mode that allows the screen to operate blue on white instead of white on blue.

All source code for the ATmega168 processor is compiled using the free WinAVR compiler and is free for downloading on the product description [page](#).

Specifications:

- Voltage: 6V – 7V DC
- Current: 220mA (backlight at 100%)
Input: 0-5V, 115,200bps (adjustable), 8 data bits, 1 stop bit, no parity
- Dimensions: 2.02" x 1.11" (51mm x 28mm)

Figure 3



Setup

Setting up the display is as easy as connecting 6V to Vin, 0V to GND and a serial TX line from your source to the RX line on the backpack. The TX line from the backpack has been left in the final design for future code revisions, debugging and user development, but it is not currently utilized as of this writing.

Voltages of up to 9V may be used to power the backpack, however care should be taken to reduce the backlight duty cycle in such cases to reduce the chance of overloading the voltage regulator on the backpack.

The backpack is primarily intended for embedded applications, but it can easily be connected to a computer and written to with a terminal emulator (the FT232RL USB to Serial Breakout, SKU:BOB-00718, is very good for this – just connect grounds and TX on the break out board to RX on the backpack).

If you're running on a terminal emulator, just open the port to 115,200bps, 8 data bits, 1 stop bit, no parity, and start typing.

Note: There is a small potentiometer on the backpack

Graphic LCD Serial Backpack Datasheet

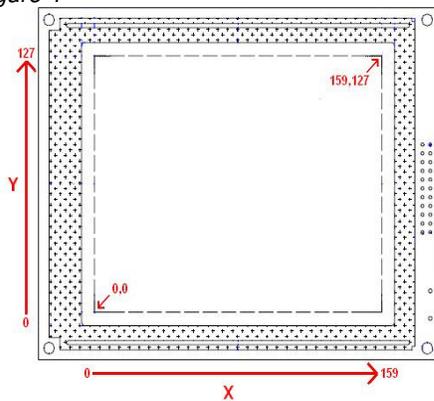
2009.09.09

that allows for contrast adjustments. This should already be adjusted for you, but if text is not readily apparent or otherwise does not suit your needs, feel free to adjust to your liking.

Operation

The graphic LCD is mapped out in Cartesian coordinates as shown in the following picture:

Figure 4



ASCII characters are printed to the screen with respect to two user-changeable settings, `x_offset` and `y_offset`. These two settings define the top-left corner bit of a character space, which is 6x8 bits. By changing `x_offset` and `y_offset` the user can place text anywhere on the screen.

Printing characters to the screen happens left to right, top to bottom, without adjusting `x_offset` and `y_offset`. Further, changing the offsets will change the entire frame of the text, meaning that writing to the end of one line and onto the next will happen seamlessly as the text has no predefined locations where it can or can't be written (except for locations close to the left and bottom edges of the display).

Backspace is also functional and tries to maintain the reference frame as set by the user.

The backpack has an input buffer of 416 bytes, the number of characters that fill the screen. Streaming more than 416 characters at a time may cause a buffer overrun, and the time it takes for the firmware to process the entire buffer is dependent on what combination of characters and special commands that

the user sends. Some experimentation may be necessary to find the best operation.

Special Commands

All commands are preceded with "|", or ASCII decimal 124 (0x7C). This tells the display that a command sequence follows. Before any of the following commands are given, they must be preceded with "|". The actual character "|" is not (and cannot) be printed to the screen.

Clear Screen:

Sending "`<control>@` (0x00)" clears the screen of all written pixels. If you're operating in normal mode, all pixels are reset. If you're operating in reverse mode, all pixels are set (white background). An example of a "clear screen" command would be to send 0x7C 0x00, or from a keyboard send "|" and `<control>@`.

Demo code:

Sending "`<control>d` (0x04)" runs demonstration code. This is in the firmware just as an example of what the display can do. To see the demonstration, send 0x7C 0x04, or from a keyboard send "|" and `<control>d`.

Reverse Mode:

Sending "`<control>r` (0x12)" toggles between white on blue display and blue on white display. Setting the reverse mode causes the screen to immediately clear with the new background. To set the reverse mode, send 0x7C 0x12, or from a keyboard send "|" and `<control>r`. This setting is saved between power cycles. If the display is turned off while in reverse mode, it will next power up in reverse mode.

Splash Screen:

Sending "`<control>s` (0x13)" allows or disallows the SparkFun logo to be displayed at power up. The splash screen serves two purposes. One is obviously to put our mark on the product, but the second is to allow a short time at power up where the display can be recovered from errant baud rate changes (see Baud Rate for more info). Disabling the splash screen suppresses the logo, but the delay remains active. To disable the splash screen, send 0x7C, 0x13, or from a keyboard send "|" and `<control>s`.

Graphic LCD Serial Backpack Datasheet

2009.09.09

Set Backlight Duty Cycle:

Sending "<control>b (0x02)" followed by a number from 0 to 100 will change the backlight intensity (and therefore current draw). Setting the value to zero turns the back light off, setting it at 100 or above turns it full on, and intermediate values set it somewhere in-between. The number setting in the command sequence is an 8-bit ASCII value. As an example, to set the backlight duty cycle to 50, send 0x7C 0x02 0x32, or from a keyboard send "|", <control>b and "2".

Change Baud Rate:

Sending "<control>g (0x07)" followed by an ASCII character from "1" to "6" changes the baud rate. The default baud rate is 115,200bps, but the backpack can be set to a variety of communication speeds:

```

"1" = 4800bps
"2" = 9600bps
"3" = 19,200bps
"4" = 38,400bps
"5" = 57,600bps
"6" = 115,200bps

```

As an example, to set the baud rate to 19,200bps, send 0x7C 0x07 0x33, or from a keyboard send "|", <control>g and "3". The baud rate setting is retained during power cycling, so if it powers down at 19,200bps, it will next power up with that setting.

In a pinch, the baud rate can be reset to 115,200. During the one second delay at power up, send the display any character at 115,200bps. The number "115200" will be shown in the upper left corner of the display, and the backpack will revert to that setting until otherwise changed.

Set X or Y Coordinates:

Sending "<control>x (0x18)" or "<control>y (0x19)" followed by a number representing a new reference coordinate changes the X or Y coordinates. The X and Y reference coordinates (x_offset and y_offset in the source code) are used by the text generator to place text at specific locations on the screen. As stated earlier, the coordinates refer to the upper left most pixel in the character space. If the offsets are within 6 pixels of the right edge of the screen or 8 pixels of the bottom, the text generator will revert to the next logical line for text so as to print a whole

character and not parts. As an example, to set x_offset to 80 (the middle of the horizontal axis) send 0x7C 0x18 0x50, or from a keyboard send "|", <control>x and "P". Attempting to set values greater than the length of each axis result in maximizing the respective offsets.

Set/Reset Pixel:

Sending "<control>p (0x10)" followed by x and y coordinates, and a 0 or 1 to determine setting or resetting of the pixel. Any pixel on the display can independently set or reset with this command. As an example, to set the pixel at (80, 64) send 0x7C 0x10 0x50 0x40 0x01, or from a keyboard send "|", <control>p, "P", "@", and <control>a. Remember that setting a pixel doesn't necessarily mean writing a one to that location, it means to write the opposite of the background. So if you're operating in reverse mode, setting a pixel actually clears the pixel and sets it apart from the white background. Resetting that pixel causes it to be white like the background.

Draw Line:

Sending "<control>l (0x0C)" followed by two sets of (x, y) coordinates defining the line's start and stop, followed by a 0 or 1 determines whether to draw or erase the line. As an example, to draw a line from (0,10) to (50,60) send 0x7C 0x0C 0x00 (x1) 0x0A (y1) 0x32 (x2) 0x3C (y2) 0x01, or from a keyboard send "|", <control>l, <control>@, <control>j, "2", "<" and <control>a. To erase the line (and leave surrounding text and graphics unchanged), submit the same command but changing the last <control>a to <control>@.

Draw Circle:

Sending "<control>c (0x03)" followed by x and y coordinates defining the center of the circle, followed by a number representing the radius of the circle, followed by a 0 or 1 determines whether to draw or erase the circle. As an example, to draw a circle at center (80, 64) with radius 10 send 0x7C 0x03 0x50 0x40 0x0A 0x01, or from a keyboard send "|", <control>c, "P", "@", <control>j and <control>a. To erase the circle (and leave surrounding text and graphics unchanged), submit the same command but changing the last <control>a to <control>@. Circles can be drawn off-grid, but only those pixels that fall within the display boundaries will be written.

Graphic LCD Serial Backpack Datasheet

2009.09.09

Draw Box:

Sending "<control>o (0x0F)" followed by two sets of (x, y) coordinates defining opposite corners of the box, followed by a 0 or 1 determines whether to draw or erase the box. This command is exactly like the draw line command, but instead of drawing a line you get a box that exactly contains the line between the given coordinates. As an example, to draw a rectangular box around the line from (0,10) to (50,60) send 0x7C 0x0F 0x00 (x1) 0x0A (y1) 0x32 (x2) 0x3C (y2) 0x01, or from a keyboard send "|", <control>o, <control>@, <control>j, "2", "<" and <control>a. To erase the box (and leave surrounding text and graphics unchanged), submit the same command but changing the last <control>a to <control>@.

Erase Block:

Sending "<control>e (0x05)" followed by two sets of (x, y) coordinates defines opposite corners of the block to be erased. This is just like the draw box command, except the contents of the box are erased to the background color. As an example, to erase a rectangular block around the line from (0,10) to (50,60) send "0x7C 0x05 0x00 (x1) 0x0A (y1) 0x32 (x2) 0x3C (y2)", or from a keyboard send "|", <control>e, <control>@, <control>j, "2" and "<".

DG5 VHand 2.0 OEM Technical Datasheet

DGTech Engineering Solutions
www.dg-tech.it

Release 1.1 November, 2007



DG5 -VHand 2.0

DGTech
Engineering Solutions
www.dg-tech.it

Table of Contents

Product Description.....	3
Technical Characteristics.....	3
Interfacing to dataglove.....	4
The output connector.....	4
Pin description:.....	4
Communication Protocol:.....	4
Serial Port Setting:.....	4
Package structure:.....	4
Finger reference:.....	6
Right Version:.....	6
Left Version:.....	6
Hand movement measures.....	7
Acceleration axis reference:.....	7
Hand orientation: roll and pitch.....	7
Accessories.....	8
VHand USB Cable	8
VHand Bluetooth Kit.....	8
DG5 Vhand 2.0 DDK:.....	8
DG5 VHand 2.0 USB DDK:.....	8
DG5 Vhand 2.0 Bluetooth DDK:.....	8
Software and Manuals Download:.....	9
Contacts.....	9

Illustration Index

Illustration 1: External connector, Pin numeration.....	4
Illustration 2: Right dataglove: bend sensor reference.....	6
Illustration 3: Data glove axes reference.....	7
Illustration 4: Roll and Pitch angle.....	7

Product Description

The new DG5 VHand data glove is a complete and innovative sensor. Thanks to its five embedded bend sensors, it is possible to accurately measure the finger movements, while the embedded 3 axes accelerometer allows to sense both the hand movements and the hand orientation (roll and pitch).

The glove communicates with external devices via a 4 wires connector. The transmission is made via TTL levels and it uses a standard RS232 protocol at 115200 bps, so interfacing the device to a PC or a microcontroller is really immediate.

The glove can be powered from 3.3V to 5V and it is really power safe, it consumes less than 20mA. It has been developed for wireless and autonomous operations and it can be powered with a battery, guaranteeing a long operative period.

The data glove can be used in different applications: robotics, motion capture, virtual reality, innovative games, rehabilitation and also as an innovative aid for disabled people.

Technical Characteristics

- **Power Supply:** from 3.3V to 5.0 V
- **Load Current:** 20mA
- **Operating Temperature:** from 0 to 50 °C
- **Storage Temperature:** from 0 to 70 °C
- **Finger Sensing Resolution:** 10 bit (1024 step)
- **Number of finger sensor:** 5 (one per finger)
- **Hand orientation resolution:** 0.5°
- **Measured Hand Acceleration:** from -2g to 2g
- **Sampling Rate:** 25Hz
- **Glove dimension:** one size fits many size (elastic). The glove is available both in right and left version
- **Output Connector:** standard AMP connector, 4 ways (GND, Vcc, TX and RX signals)
- **Output High (1) signal level (RX and TX):** from 2.8 to 3.3V
- **Output Low (0) signal level (RX and TX):** from 0.0V to 0.3V

Interfacing the dataglove

The output connector

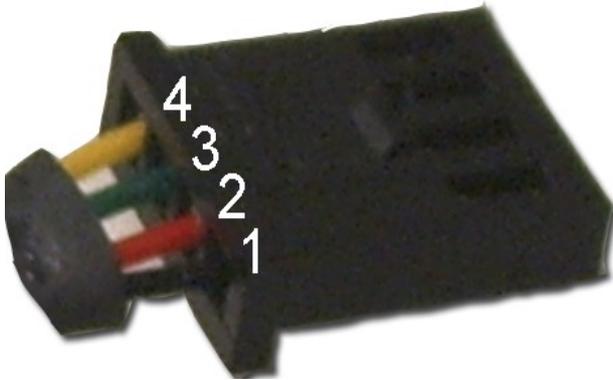


Illustration 1: External connector, Pin numeration

Pin description:

- 1 (Black): GND
- 2 (Red): Vcc
- 3 (Green): RX
- 4 (Yellow): TX

In order to power the data glove, it is needed to provide a supply from 3.3V to 5.0V on the Vcc pin. The RX and the TX pin are used for the communication;

RX: receive, the dataglove listens here the data from the controlling device;

TX: transmit, the dataglove sends on this pin the data package, containing the dataglove status;

Communication Protocol:

Serial Port Setting:

Baud Rate: 115200 BPS
Data Bit: 8
Stop Bit: 1
Parity: NONE

- start transmission (controller to dataglove): send 's' to the glove;
- (Data glove to controller): the glove transmits the package continuously;
- stop transmission (controller to dataglove): send 'e' to the glove;

Package structure:

The glove continuously transmits to the host device the following 20 byte package:

```
1 - header = 0x20
2 - header = 0x0A
3 - length = 0x14 (20 byte)
4 - acceleration axis ax_l
5 - acceleration axis ax_h
6 - acceleration axis ay_l
7 - acceleration axis ay_h
8 - acceleration axis az_l
9 - acceleration axis az_h
10- bend 0_l
```

11 - bend 0_h
12 - bend 1_l
13- bend 1_h
14- bend 2_l
15- bend 2_h
16 - bend 3_l
17 - bend 3_h
18 - bend 4_l
19 - bend 4_h
20 - crc

CRC represents the XOR of the first 19 bytes.
Bend value are from 0 to 1023 so only 10 bit are used.

The acceleration values are from -32767 to +32767.
The relative accelerations can be computed with the following relations:

$$Acc_x = 2 * (ax_l + ax_h * 128) * sign(ax_h) / 32767 ;$$

$$Acc_y = 2 * (ay_l + ay_h * 128) * sign(ay_h) / 32767 ;$$

$$Acc_z = 2 * (az_l + az_h * 128) * sign(az_h) / 32767 ;$$

The value are from -2g to +2g.

Finger reference:

In the following picture are reported the finger reference for both the right and the left dataglove versions.

Right Version:

Thumb	Bend 0
Index	Bend 1
Middle	Bend 2
Ring	Bend 3
Little	Bend 4

Left Version:

Thumb	Bend 4
Index	Bend 3
Middle	Bend 2
Ring	Bend 1
Little	Bend 0

Right Dataglove Bend Sensor Reference



Illustration 2: Right dataglove: bend sensor reference

Hand movement measures

Acceleration axis reference:

The dataglove incorporates a 3 axes accelerometer able to measure the hand acceleration along the 3 main axes. The hand orientations (roll and pitch) can be extracted from these information. In the following picture it is explained the data glove axes reference.

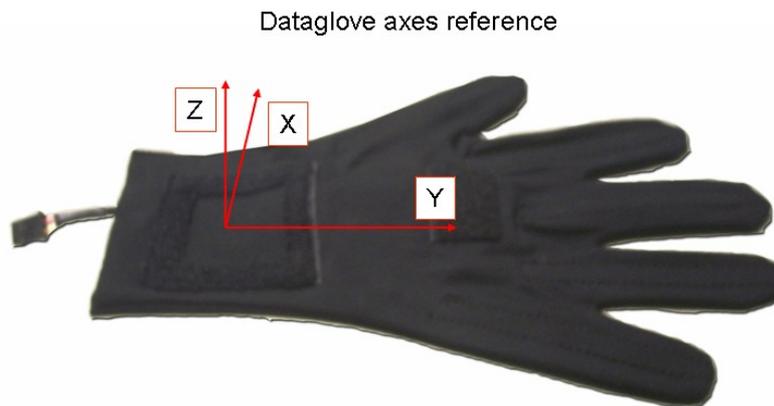


Illustration 3: Data glove axes reference

Hand orientation: roll and pitch

The accelerations can be divided in two types: dynamic and static. The 3 axes accelerometer measures both the types. A dynamic acceleration is a measure of the instantaneous movement of the hand along one of the three axes. Static accelerations are due to the gravity force, for example rolling the hand around the y axis (roll) will modify the acceleration measured along the x axis and along the z axis, since the component of the gravity along these axes is modified, then the pitch angle can be deduced, for example, by getting the static measure of the x acceleration. Vice versa, rotating the hand along the x axis (pitch) will modify the measured y acceleration, so the pitch angle can be measured.

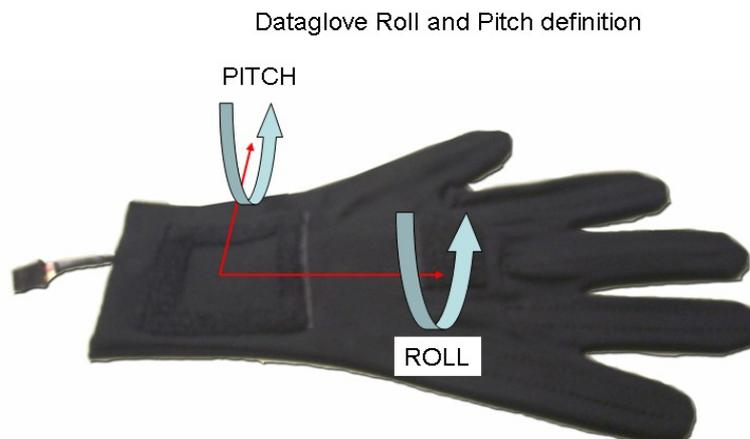


Illustration 4: Roll and Pitch angle

Accessories

The DG5 VHand 2.0 OEM is provided as is, so the customer has to develop its own communication board.

In order to facilitate and accelerate the developing, some accessories are available; this accessories can be bought separately or with the data glove DDK package (see later in this section).

VHand USB Cable

A ttl to usb cable which permits to immediately connect the data glove to a USB port of a PC. The cable provide also the Vcc, so an external power supply is not necessary;

RIF: VUSB

VHand Bluetooth Kit

A TTL to Bluetooth adapter with an external battery; simply connect the glove and the battery to the adapter and start communicating with the glove in a wireless way. A BlueTooth receiver or a PC with a BlueTooth connection is needed.

RIF: VBLUE

DG5 Vhand 2.0 DDK:

To facilitate the developers, two complete DDK (Device Developing Kit) are available.

DG5 VHand 2.0 USB DDK:

The kit comprises:

1 DG5 Vhand 2.0 OEM

1 TTL to USB adapter cable

1 CD containing all the software and the SDK (Software Developing Kit), with all the libraries needed to fast develop custom application.

Manuals and Datasheets

RIF: VUSB-DDK

DG5 Vhand 2.0 Bluetooth DDK:

The kit comprises:

1 DG5 Vhand 2.0 OEM

1 Bluetooth Kit (adaptor and battery)

1 CD containing all the software and the SDK (Software Developing Kit), with all the libraries needed to fast develop custom application.

Manuals and Datasheets

RIF: VBLUE-DDK

Software and Manuals Download:

The latest software and manual versions can be found here:

www.dg-tech.it/vhand/

Contacts

For further information please contact us:

DGTech Engineering Solution

via Calzolaria, 20 40053 Bazzano (BO) – Italy

Phone: +39 (0)51 832149

Fax: +39 (0)51 832149

Email: info@dg-tech.it

www.dg-tech.it



www.dg-tech.it

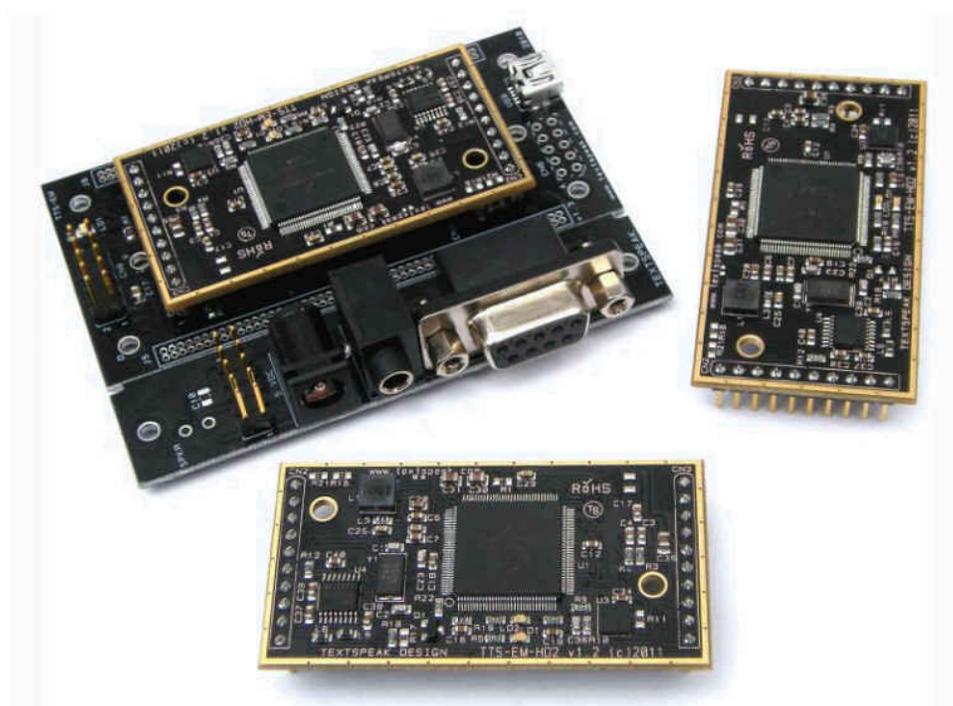
TextSpeak™

Users' Guide

TTS-EM-HD2

High Definition Speech
Synthesizer Systems

Version 2.0 Release



Advanced Information
V 2.0 Rev A December 2012

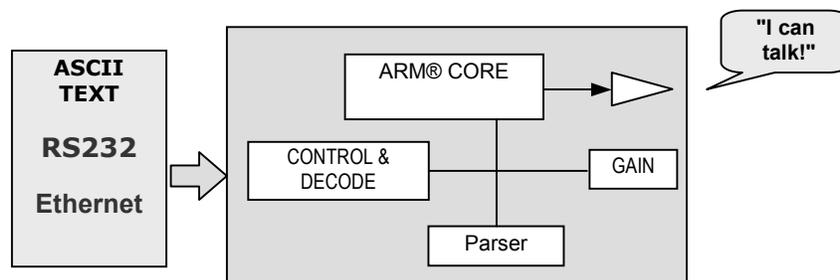
Consult factory for final specifications
prior to design-in

Version 3.3.21 or greater revision level

TTS-EM OVERVIEW

The TextSpeak 'EM' Text-To-Speech processor converts ASCII text directly to voice with unlimited vocabulary. The platform boasts multi-language capability on a sophisticated board level solution that accepts RS-232 and optional Ethernet input to provide high quality speech from text based data streams.

- Unlimited vocabulary Text to Speech generation
- Multi-language capability with 'a-la-carte' feature scalability
- Integrated command language and in stream control codes
- Dual audio output; Line Out and Speaker Amplifier (1w/8ohm)
- Switches seamlessly from TTS to pre-recorded files on the fly.
- Low power operation with auto power down and sleep modes
- Flash upgradeable for both revision and language enhancements
- RS-232 expandable to accept USB, Ethernet or 802.11 ASCII coded data
- Small footprint, 1" x 2" board incorporating standard .1" 1x10 headers
- Optional Ethernet, Automatic Level Control & integrated 25 watt amplifiers
- RoHS and Industrial Temp standard.



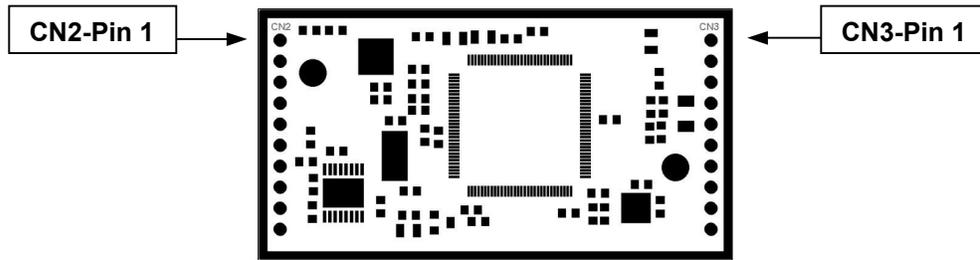
Applications

- Emergency Notification
- Alarm and security
- Transportation systems
- Signage to Speech
- Machine control interfaces
- Streaming text reader

Other Applications include:

Wireless audio transmitters and information systems, Email/SMS Readers, GPS/Navigation Systems, Warning systems, Machine accessories, Smart Pagers, Ticketing and Kiosks, disability compliance for hearing impaired individuals (ADA).

CONTROLS AND CONNECTIONS



CN3	Signal	1x10 Header Pin Description
CN3-1	GND	Analog ground
CN3-2	GND	System ground
CN3-3	RX	RS-232 Signal from HOST to TTS-EM
CN3-4	TX	RS-232 Signal from TTS-EM to HOST
CN3-5,6	VCC	5 VDC Power
CN3-7	Speaker +	Speaker out + 400mw (8ohm) > Refer to Note 1!
CN3-8	Speaker -	Speaker out - 400mw (8ohm) > Refer to Note 1!
CN3-9	GND	System ground
CN3-10	Line Out	Unbalanced TTS audio output



Note 1: Speaker outputs pins are floated and balanced. They must be directly connected to speaker terminals.
Do not connect or reference pins 7 or 8 to ground!

CN2	Signal	1x10 Header Pin Description
CN2-1	n/a	Reserved for 3.6V powering option (consult factory)
CN2-2	RX-DATA	Non-Inverted Serial Data from HOST to TTS-EM
CN2-3	TX-DATA	Non-Inverted Serial Data from TTS-EM to HOST
CN2-4	nTALKING	Digital Output 1=Idle 0= Speech generation active
5,6,7,8	Test	Test Pins/ Do not connect
CN2-9,10	RS-Select	Connect 9 & 10 to enable pins 2/3 for non-inverted digital serial I/O. Also disables the RS-232 signal stream

OEM-TTS-EM RS-232 interface - low level commands

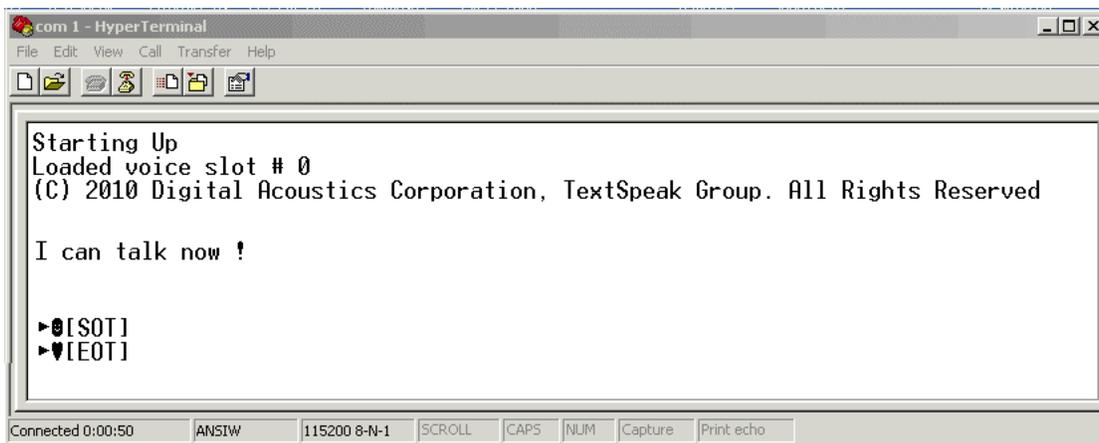
Real time text may be continuously sent to the TTS-EM using RS-232 text transfer. TextSpeak TTS-EM will also speak phrases as you type in real time.

In manual typing mode pressing the Enter key <CR> to begin a speech playback. In streaming mode both ASCII <CR> or <DLE>1 will begin a speech playback.

Initial Setup and Manual Speech Generation

- Launch a standard Terminal program and select an RS-232 port.
- Set Communications to 115,200 bps (8,n,1)
- Insure that Hardware Flow control is set OFF
- Set Terminal Emulation to ANSI (optional)
- Power-up your TTS-EM (wait 10 seconds for self test)
- Type any phrase, followed by a the Enter key

<CR> is the convention for *Enter, Carriage Return* (ASCII 13, 0xD)
This is invokes sending a phrase to the TTS conversion routine



```
com 1 - HyperTerminal
File Edit View Call Transfer Help
Starting Up
Loaded voice slot # 0
(C) 2010 Digital Acoustics Corporation, TextSpeak Group. All Rights Reserved

I can talk now !

▶[SOT]
▶[EOT]
```

Connected 0:00:50 ANSIW 115200 8-N-1 SCROLL CAPS NUM Capture Print echo

Menu Operation and Command Mode

Setup, information and special commands are available via the RS-232 COMMAND MODE interface

Options available in Command mode are:

- View current voice selections
- Change the default mode voice if several voices are activated
- Playback demonstrations of multi-language voices
- Edit the spoken power-up prompt
- Update programs features and revision
- Change serial baud rate

After Entering COMMAND MODE all TTS functions will be temporarily suspended

To enter COMMAND MODE a required sequence of 3 "+" characters must be manually typed (with 500-700ms pause between each character)

Example:

+++

will result in an RS-232 response:

CMD>

Press **H** to view available choices. The RS-232 screen will respond with a help menu.

```
(C) 2010-2013 Digital Acoustics Corp. TextSpeak Group. All Rights Reserved

Unique Identifier '{44B446BB-5AE6-A2EC-E94B-A5612AAAA1B6}'

HELP:

A          Adaptive Level      Set Adaptive Level Control (ALC), view ambient noise.
B"text"    Boot TTS             Enter the text to be spoken at power-up (ex: b"hello")
C          Clear Screen        This will clear an ANSI terminal screen
D          Display Files       Show listing of files and memory in folder "UserData"
E          ignore ESC          Ignore ESC commands in the TTS stream
F          Flash Drive         Enable USB access to manage Flash memory
H          Help                This list
I          Information         Get current settings information
P#         Play Audio          Generate test tones (1=1khz,2=sweep,3=noise)
Q#         Quiet Mode          'Q?' for options
S          Change Serial       115200,n,8,1
T          TTS Mode            Go from CMD mode to TTS mode
U          Update              Update Firmware
V#         Select Voice       Select a voice slot
?          Review              Review current settings and information

Cmd>
```

Command Mode Overview

A	ALC Automatic Level Control and Amplifier control (available on ENCx and TTS-HD3)
B	Boot phrase to be spoken in TTS on power-up. (ex: B"anyphrase" <CR>)
C	Clears terminal screen (ex: C <CR>)
D	Display audio files stored in local memory default folder "UserData"
E	ESC ON/OFF. Disable ESC sequence processing in data stream
F	Flash Drive Enable USB port for audio or upgrade file transfer
H	Help Displays Help Menu
I	Information Display version and settings
L#	Level sets and saves audio volume output level #(0-9 range @ 4db/step)
P#	Playback Generate playback of tones or test patterns #(0-3)
Q#	Quiet Modes. Set power-up audio announcement & terminal duplex echo #(0-2)
S	Serial RS-232 Speed Change (ex: S115200 <CR>)
T	TTS mode return. Exits command mode (CMD>)
U	Upgrade ; reserved for field upgrades
V#	Voice provides a means to choose a voice and set the power-up default voice. Refer to '?' (below) to list available voices (ex: V1 <CR> selects voice "James") #(0-400)
?	Review voices. Provides information on available languages and voices.

Voice File Index Lists

Command Mode option "V#" provides a unique method to select and preview samples of voices in various languages. The "?" command will display a list of voices and their current availability. Active voices may be selected for TTS operation, or licensed. The screen below is a sample of voices for selection for 5 available languages.

```
Cmd> ?
Voices:
Slot   Language      Status      Name
0      U.S. English  Not Installed  Carol
1      U.S. English  Activated    James
2      French/Canada Not Installed  Celine
3      French/Canada Not Installed  Antoine
4      German        Not Installed  Babschi
5      German        Not Installed  Manfred
6      U.K English   Not Installed  Melanie
7      U.K English   Not Installed  Leland
8      Spanish       Not Installed  Maria
9      Spanish       Not Installed  Santos
101    U.S. Premium  Activated    Susan
102    U.S. Premium  Not Installed  Steven

Current Voice = Slot 1

To purchase and activate a voice, contact sales@textspeak.com

Press H for Help Menu

Cmd>
```

The example cited above displays a list of voices. Status information displayed provides the following information:

- **Activated** Voices are available for selection using "V" Command. The selected voice is saved in memory and started in power-up.
- **Not Activated** The voice is installed, but not licensed for use. Contact sales@textspeak.com for purchase and licensing procedures.
- **Not Installed** The voice is not installed in Flash Memory at the time of purchase. Contact TextSpeak for additional information.

Commonly used menu commands are **V#** and **?** to change voices and **T** return to TTS Mode from CMD prompt. For inline commands that may be used embedded within the spoken text stream refer to the next section.

For languages not shown please contact sales@textspeak.com for the latest pricing and availability of additional worldwide languages

Inline and Keyboard Control Codes

The table below reflects Keyboard/TYPED and RS-232/ASCII control
 Numeral shown with `d' indicate ASCII decimal values not character "d"

Conventions: ▶ = **16d** and <esc> = **27d** or a keyboard **ESC** key

Example: ▶**2** is ASCII data decimal 16 and 2 (0x10,0x02) <esc> **!** is 23

ACTION	TYPED	ASCII	ECHO	REMARK
INVOKE SPEECH	<CR>	▶<02d>	▶<2d> [SOT]	Begin a TTS of text in buffer.
STOP CAPTURE	n/a	▶<25d>	▶<25d> [EM]	End of Medium. Stop adding text to speech buffer
RESTART CAPTURE	n/a	▶<30d>	▶<30d> [RS]	Start capturing text, add text to buffer.
PLAY FILE	<esc> p "any.wav"	▶p"any.wav"	[ACK] <06d>	Begin synchronous playback of recorded audio file "any.wav"
CANCEL	<esc> <esc>	▶<24d>	▶<24d> [CAN]	Cancel/Abort speech in progress and clear buffer.
REPEAT	<esc>/	n/a	n/a	Repeat (must follow a space and precede the <CR)
VOL DN	<esc> (▶<14d>	[Vol=#]	Volume Down, 0-9, 10 steps
VOL UP	<esc>)	▶<15d>	[Vol=#]	Volume Up 0-9, 0-9, 10 steps
VOL SET	n/a	▶<18d>#	▶<18d> # [DC2]	Volume Set, #=0-9
SLEEP	<esc> !	▶<23d>	▶<23d> [SLP]	Enter Suspend mode. Wake on next (dummy) character
RESPONSES (INDICATES STATUS, SUCH AS SPEECH CONVERTING OR FINISHED TTS)				
ACTIVE TTS	n/a	n/a	▶<01d> [SOH]	TTS Speech generation in progress
IDLE	n/a	n/a	▶<03d> [EOT]	End of TTS Speech generation.
OK	n/a	n/a	[ACK] <06d>	Command Acknowledgment
WAKE	n/a	n/a	▶<26d> [RDY]	Awake from sleep, characters will be processed

Inline (real-time) Language/Voice/Audio File Playback

The code sequence is **<ESC> [digit]** in ASCII will execute a real-time voice change. The [digit] numbers, ASCII "0"- "9" are the voices, in order, as listed in the voice listing menu. Voices change in about 2-3 seconds.

One exception for switching times is transitions between Steven and Susan... (101 and 102). These changes require 6-8 seconds.

EXAMPLE

This sample text below demonstrates both spoken and non-spoken data with pre-recorded audio files. (tone1.wav and tone2.wav are factory recorded files). Sample below uses default root, indicated by "\". Default user folder is "UserData" and does not require the "\".

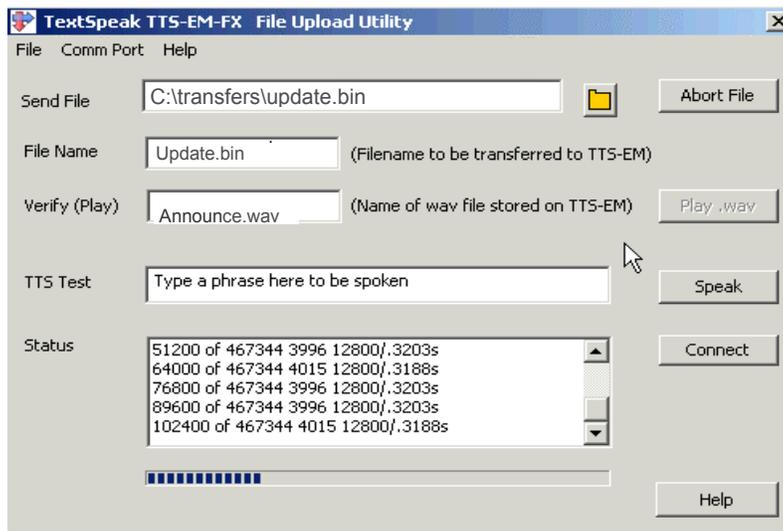
```
ESCp"\tone1.wav" Express 32, arriving in 2 minutes on track 12, for all points
South. DLEEM This text is in-line data that is not spoken by the TTS processor
DLERS Please expect delays ESCp"\tone2.wav". End of test. CR
```

[download online sample](#)

Software Utilities

TextSpeak **EM-FX** software utility simplifies software upgrades and audio file transfers for TTS-EM-HD2 and HD3 products using the TTS speech data channel.

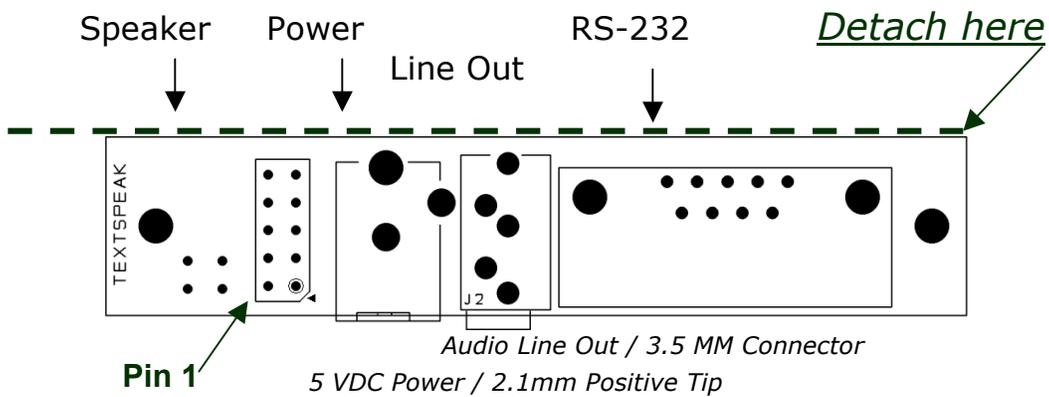
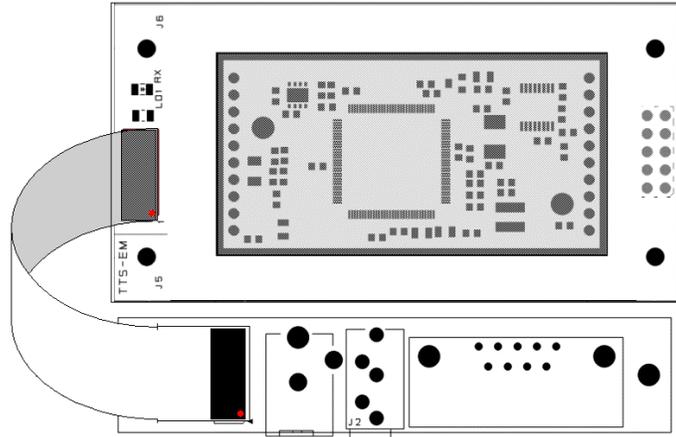
Downloads and additional information is at www.textspeak.com/doc.htm



Using the TTS-HD2-EK motherboard

TTS-EM optional mini-motherboard provides all the connections for test and evaluation of your TTS-EM module.

The motherboard is detachable from the main PCB. *To separate, manually snap PCB at scored line)*



Pin	Signal	2x5 Header Pin Description
1	GND	Analog ground
2	GND	System ground
3	RX	RS-232 Signal from HOST to TTS-EM
4	TX	RS-232 Signal from TTS-EM to HOST
5,6	VCC	5 VDC Power
7	Speaker +	Speaker out + 400mw (8ohm) > Refer to Note 1!
8	Speaker -	Speaker out - 400mw (8ohm) > Refer to Note 1!
9	GND	System ground
10	Line Out	Speech output nominal 1v p-p into 10K



Note 1: Speaker outputs pins are floated and balanced. They must be directly connected to speaker terminals.
Do not connect or reference pins 7 or 8 to ground!

Accessories and Options

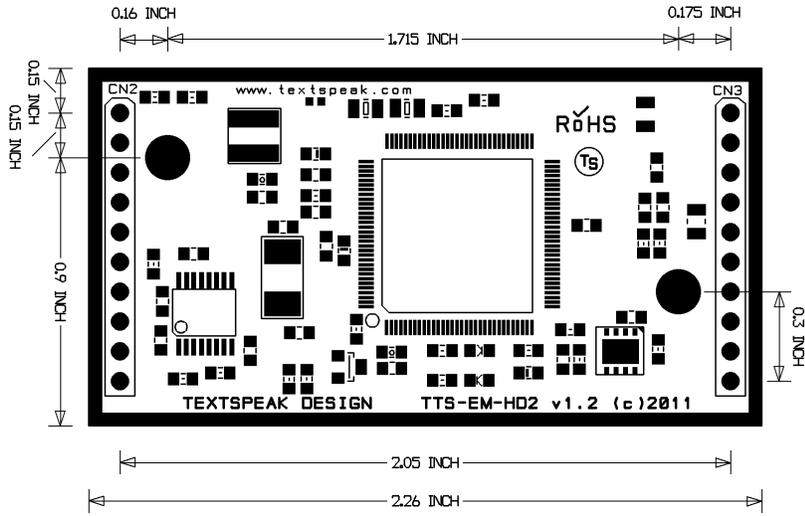
Your TextSpeak TTS-EM-HD2 is part of a family of TextSpeak products.

Access www.textspeak.com to learn more

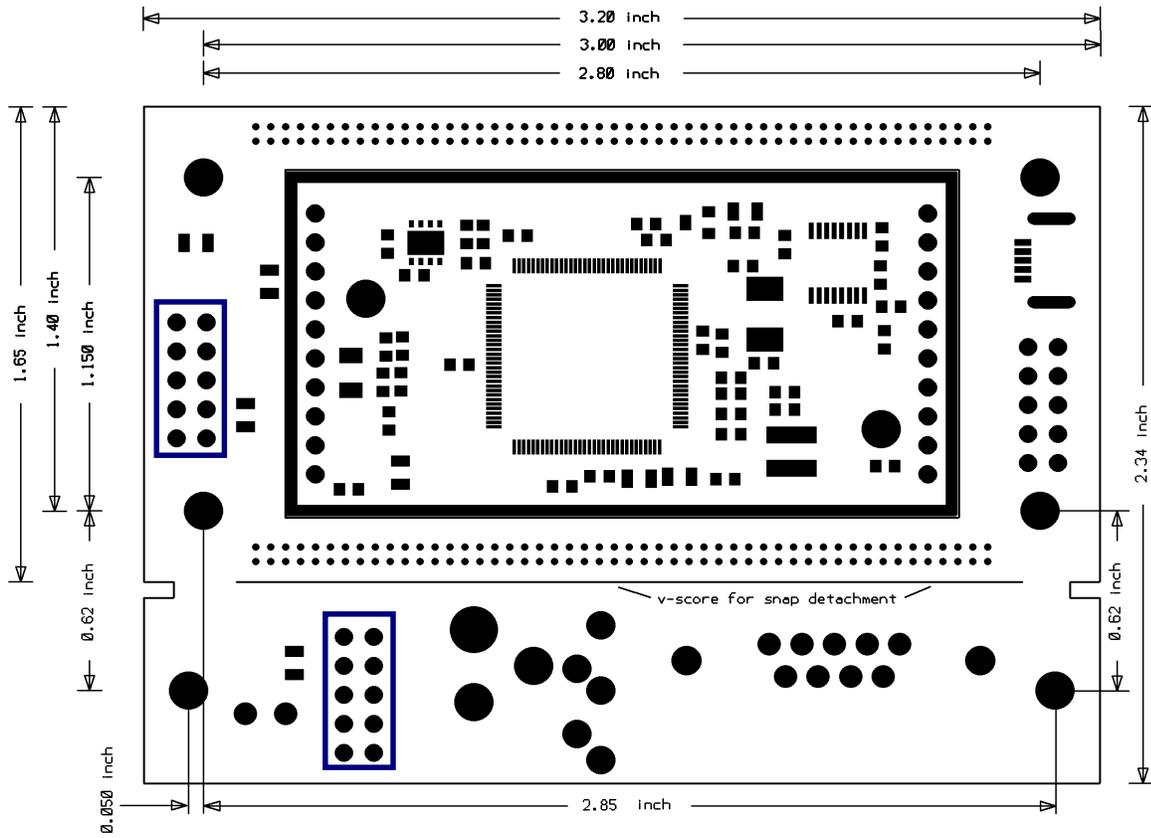
2013 Speech Product Overview (subject to change)

PRODUCT NAME	TTS-03	TTS-06	TTS-08	TTS-EM-HD1	TTS-EM-HD2	TTS-EM-HD3	TTS-EM-ENC1	TTS-EM-ENC2	AUD-EM-3
Physical Package	Module	Module	Enclosure	Module	Module	Module	Enclosure	Enclosure	Enclosure
Enclosure Style	x	x	Plastic	x	x	x	Industrial	Industrial	Industrial
Dimension	1x2"	1x2"	2x2"	1x2"	2x3"	2x3"	4x5"	4x5"	4x5"
Languages Supported	1 US	2	2	4	20+	24+	24+	24+	x
Standard Language*	English	Eng/Span	Eng/Span	World TTS	World TTS	World TTS	World TTS	World TTS	n/a
Intelligibility	Medium	Medium		HD Quality	HD Quality	HD Quality	HD Quality	HD Quality	n/a
Dynamic voice swap	x	x	x	x	x	✓	✓	✓	✓
Dictionary support	x	x	x	x	x	✓	✓	✓	n/a
Audio File Playback	x	x	x	.wav	.wav	.wav	.wav	.wav	.wav/.mp3
Line Out	x	✓	✓	✓	✓	✓	✓	✓	✓
Audio Amp (W RMS)	0.8	0.8	1	1	1	3/8/25w	3/8/25w	3/8/25w	3/8/25w
Ambient noise detection	x	x	x	x	x	✓	✓	✓	✓
Auto Volume Adjust	x	x	x	x	x	12/20db	12/20db	12/20db	12/20db
USART TTL I/O	x	x	✓	✓	✓	✓	✓	✓	✓
RS232/Ethernet	RS232	RS232	RS232	RS232	RS232	RS232	RS232	TCP/UDP	RS232/UDP
USB File Transfer	x	x	x	x	x	✓	✓	✓	✓
Remote File Transfer	x	x	x	x	x	✓	✓	✓	✓
Remote Upgrades	x	x	x	x	x	✓	✓	✓	✓
Power	9v	5v	7-12v	5v	5v	9-24v	9-24v	9-24v	9-24v
Standby modes	x	x	x	x	x	✓	✓	x	✓
Solar sleep modes	x	x	x	x	x	✓	✓	x	✓
Technology Core	WB701	TTS06	TTS06	HD1	HD2	HD3	HD3	HD3	HD3
Ordering notes	* Select/Choose WorldTTS languages when ordering (-x)								
Ordering code	TTS03	TTS06	TTS08	TTSHD1-x	TTSHD2-x	TTSHD3-x	TTSENC1-x	TTSENC2-x	WPAUD3

Dimensions HD1



Dimensions HD2



Technical Support

Information online www.textspeak.com
Email tsupport@textspeak.com
Telephone +1 (203)-803-1069 M-F 9-5 (GMT-5)

Specifications HD1

Voltage In	
Sleep Mode ** (consult factory for 35 mw sleep option)	100mw 5V@20ma
Idle Mode	350mw 5V@70ma
TTS Active Line Out Only (Speaker Amp Idle)	700mw
TTS Active Speaker Amp @ 1W/8ohm	2.5W
Nominal Line Out	-10 db
Speaker Out Max	~800mw/8ohm 10% THD
Volume Steps	7
RS-232 Command Buffer	1024 bytes
RoHS	Yes
Weight	.35 oz (9 g)
Size (HD1 Module)	2.25 x 1.25 in
** Version 3.2.x or later OS	

ELUA

Hardware End User License Restrictions

The end user is licensed to use the TTS-EM for speech generation for real-time Text to Speech conversation and playback in a public place. The end user may not use the TTS-EM to record and/or save audio in stored files to be used to playback or broadcast in any public place.

#

Digital Acoustics Corporation TextSpeak Design Group, Westport, CT 06880 U.S.A.

This document may contain advance information. Contact factory for technical specification before product design and/or use. Design and specifications are subject to change without notice. Digital Acoustics' product designs are packaged by leading manufacturers around the world. Digital Acoustics® is a registered trademark.
© 2012 Digital Acoustics Corporation TextSpeak Design Group (TS_EM-UserGude-HD2-V2.doc)