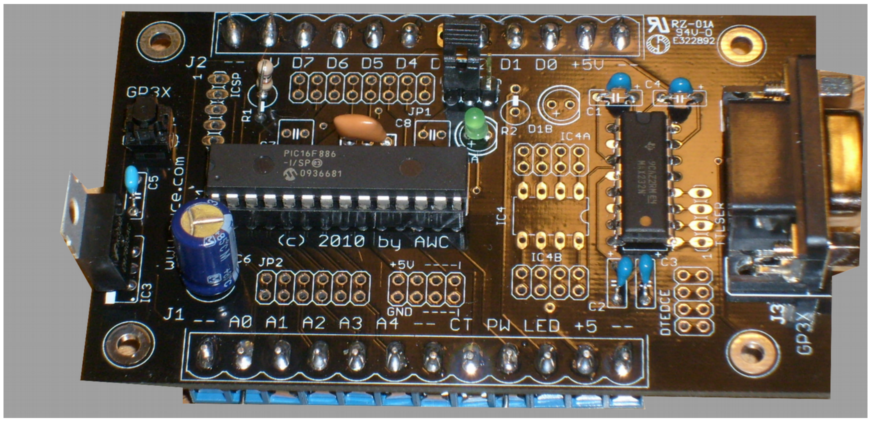
AWC

GP3 PC Analog/Digital I/O Kit



For the latest documentation see http://www.awce.com/doclib.htm#gp3

AWC
http://www.awce.com
v1.6 15 April 2015

Table of Contents

Overview

The GP3 is the easiest way to interface a PC to the real world. You can also develop using your PC and then let the GP3 sense and control the real world on its own. This kit provides a hardware interface that allows you to easily monitor and control analog and digital circuitry. Features include:

- 8 general purpose I/O lines (digital input/output with special features like PWM and pulse output)
- 5 10-bit A/D inputs
- 1 hardware PWM output (operates continuously)
- 1 hardware counter input (operates continuously)
- 1 LED under software control
- True RS232 from onboard 5V supply
- EEPROM for storing configuration or serial numbers
- Efficient control via serial protocol (57600 baud) or use supplied libraries for C, C++, Visual Basic, Java, Linux, etc.

Documentation

To ensure you have the latest documentation, please check online (http://www.awce.com/doclib.htm#gp3) for the latest versions and updates.

If You Need Help

If you require assistance, please feel free to contact us. The best way to get support is via our FAQ system (http://www.awce.com/faqs). Be sure to check out our Web page for updates at www.al-williams.com/awce.

Details

The GP-3 can be constructed on one of several base boards:

1. The GPMPU28 – This small form factor board fits nearly anywhere. Convenient to plug into a solderless breadboard.
2. The GPMPU40 – Larger and more customizable than the GPMPU28. Simple to plug into a solderless breadboard or connect in a fixed installation.
3. The GP3X – This board offers the ultimate customization capabilities for the GP3.

There are 6 sections on any GP3 system:

- CPU – The CPU chip contains the firmware and memory for the system. Note that a jumper input selects if the GP3 executes it onboard software or if it accepts commands from a PC. On the GPMPU28 this jumper is on the edge connector. On the GP3X and GPMPU40 there is a dedicated jumper.
- Clock – The CPU requires a 20MHz oscillator. On the GPMPU28 board a three pin 20MHz resonator is required. The other boards normally take the same resonator, but can accept a crystal and associated capacitors.
- Communications – By default, all versions of GP3 use RS232 connections and will work with most USB to serial adapters. In addition, the GP3X board has provisions for a USB daughter board or cable. It is also possible to fit the USB hardware in a GPMPU40 board.
- Power – The GP3 requires a 5 volt regulated supply. You can connect an external regulated supply or use the onboard power system on any of the base boards. If using the USB daughter board/cable, the GP3X can obtain power from the USB port.
- Reset – The CPU requires a pull up resistor on the reset pin. In addition, a jumper or switch can be used to manually reset.

- Interface – The whole point to the GP3 is to interface to the outside world. Each board has a variety of methods to connect it to the outside world.

The remainder of this manual will be divided into two major parts: Hardware and Software. The Hardware portion will discuss each board option in turn. You should not need to read the sections that do not apply to the board you've selected. The Software portion deals with how to communicate and control the GP3 and applies to all versions of the hardware.

Hardware

Please select the section below that applies to the base board supplied with your kit. If your kit was provided assembled, you may still wish to read the sections below to get a better understanding of the layout of the board.

GP3X

There are several variations for the GP3X. If you are using the Arduino or Raspberry version, many of the parts on the board are not populated as they are not needed. You should read and understand the GP3X build instructions and then refer to the specific sections for your particular variant to learn the details about it.

GP3X Parts List

Designator	*Description*
C1-C4	1 uF capacitor (not included with USB kit)
C5	0.1 uF capacitor
C6	330 uF capacitor (not included with USB kit)
C7, C8	Optional. Capacitors matched to X1 if needed.
D1A	5V LED
D1B	5V or regular LED (see options)
IC1	GP3 CPU and 28 pin narrow socket
IC2	MAX232 (not included with USB kit)
IC3	7805 (not included with USB kit)
IC4	USB cable or daughter board (UB232R); If daughterboard used, includes two 4 pin socket strips and is usually mounted on bottom side of board. If not used, IC4 can be used for other purposes (see options);
J1, J2	Connectors on .2" centers; usually installed on bottom of board, but may be installed on top if preferred (see options). Standard kit comes with pin headers (which may be on .1" centers; remove extra pins with pliers) or optional screw terminals which interlock to make 12 position connectors.
J3	DB9 female (right angle, short reach; not included with USB kit)
JP1, JP2	Normally nothing is connected here. You can cut traces between the centers and rewire the digital and analog I/O to your liking (see options)
R1	10K resistor
R2	Dropping resistor for D1B, if required
X1	20Mhz ceramic resonator (three terminal) or 20Mhz crystal (outer pins only) with matching C7 and C8 installed. For Pi-compatible version, the resonator is 10MHz.
Reset	(optional) 5mm tact switch (pins 1 and 3) or 2 pin header (pins 1 and 2)
Pgm Run	3 pin header and jumper
+V	Coaxial power jack (optional; see text)
Empty	By default ICSP, JP1, JP2, R2 D1B, C7, C8, IC4, IC4A, IC4B, TTLSER, DTEDCE, and +5V/GND header (to the right of JP2) are empty. See customization for more information on how these parts of the board may be used.

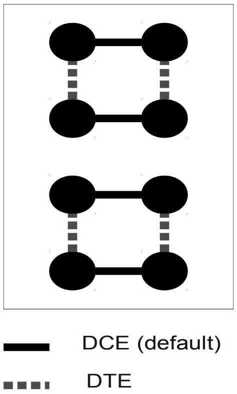
The GP3X board is highly flexible. Each of the GP3 sections has several options for customization. You should plan the options that will meet your needs before starting construction.

- CPU – The CPU should be socketed, although this is not strictly necessary. The 3 pin header marked

Pgm Run (near JP1) allows you to install a header to switch the board between the mode where it accepts commands from the PC (Pgm) or executes its internal program (Run). Normally you will install a 3 pin header in this location and a jumper will select between the two options. However, if you are certain you only want to operate the board in one mode, you can hard wire the jumper.

- Clock – By default, kits based on the GP3X board use a 20MHz ceramic resonator that includes capacitors. However, the same socket can accept a regular resonator or a crystal in which case you'll need to install C7 and C8 with the capacitors required. In addition, an 8 pin DIP oscillator (20MHz) could be installed at IC4 and its output fed into pin 9 of IC1.
- Communications – By default, the GP3 communicates via a DCE RS232 port formed by J3, IC2, and C1-C4. A DCE port mates via a straight cable to DTE ports such as those commonly found on a PC. However, there are several options available.

  First, you can rewire the port so it selectable between DTE and DCE. Near J3 there is a 2x4 pin header marked DTEDCE. The PC board has traces running across the pins. You can carefully remove the traces between each of the four pairs of pins using a hobby knife or razor (the traces are on the top side of the board – the side with the silk screen markings). Then you can install a 2x4 pin header into the holes. If you place 4 jumpers across the pins (making the same connections as the traces did) the port will be DCE. However, if you place the jumpers at right angles, the port will be DTE suitable for connection to, for example, a modem. However, if wired as DTE, you'll need a cross cable (sometimes called a null modem cable) to connect to a PC for programming. The figure below shows how the DTEDCE jumpers can be connected.



  If you are connecting the GP3 to a non-RS232 device like a microcontroller, you can use the TTLSER pins and remove IC2. The pins are:

  1 – Handshaking in to GP3 (RTS)
  2 – Handshaking out to GP3 (CTS)
  3 – Transmit from PIC (TX)
  4 – Receive from PIC (RX)

Recent GP-X boards use a special USB cable for communications. The cable has six wires. You may want to solder the cable to the top side of the board or the bottom, depending on how you plan to mount the finished unit. The table below shows where each wire is connected by color:

Color	*Function*	*GP3X connection*
Black	Ground	GND (any of the GND pins near the A4 pin on J1)
Red	+5V	+5V (any of the +5V pins near the A4 pin on J1); omit if using seperate power supply
Yellow	RTS	TTLSER pin 1
Blue	CTS	TTLSER pin 2
White	TX	TTLSER pin 3
Green	RX	TTLSER pin 4

After testing the GP3-X, you may want to use a small zip tie to strain relief the cable by passing the zip tie through the two large holes in the unused J3 footprint, ensuring that the USB cable is encircled by the zip tie. Pull the tie tight to lock the cable down to the board.

Older GP3-X boards used a daughterboard for USB communications. The IC4 socket can accept this USB daughter board (part number UB232R). Normally this board is fitted under the board using the socket strips supplied with the daughter board. The pins are soldered on the top side of the board (the side of the board with the silk screen markings). Because of the unusual pin spacing of the daughter board, one row of pins goes into IC4's pins 1-4. The other row goes into the row of holes marked IC4B that are nearest to IC4. The USB port should face the edge of the board that holds J1.

Note that because the daughter board is installed under the board, the pin numbers are reversed. That is, pin 1 of the module corresponds to IC4 pin 8 and the module's pin 8 will connect to IC4 pin 1. With the USB port on the underside of the board, it is usually a good idea to use spacers or bolts in the four corner mounting holes to form "feet" for the board so it does not rest on the USB interface.

The connections between the daughter board and the rest of the board appear in the table using the IC4 pin numbers and the module pin numbers for reference. If you wish to power the board from its own power supply or from an external 5V source, do not connect pin 2 of the USB board. However, if you wish to operate the board from the USB power supply you can connect pin 2 to one of the holes near JP2 marked +5V and omit IC2 and C6. The jumper on the daughter board should be connected between pins 1 and 2 so that the board operates on 5V instead of 3.3V.

IC4 pin #	*USB232 Pin #*	*GP3X connection*
8	1	Ground
7	2	+5V (if USB power desired; no connection if using separate power supply)
6	3	TTLSER-2
5	4	TTLSER-1
2	7	TTLSER-3
1	8	TTLSER-4

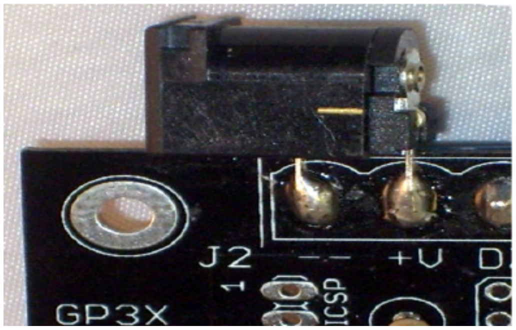
Note that it is possible to socket IC2 and the USB module so that the board can operate in either mode. With the power removed, simply install either IC2 or the USB module as desired. To switch the board to use the other type of communications, remove the power. Then remove either IC2 or the USB module and install the other unit.

- Power – There are several options to power the GP3. The only actual requirement is a regulated 5 volt (5V) supply. Feeding such a regulated voltage into any of the pins on J1 or J2 marked 5V (and ground on any of the pins marked –) is sufficient to power the board (with all I/O quiescent, the board draws roughly 11 to 15 mA).

If you prefer, you can install the regulator (IC3) and C6 and feed a higher voltage (usually 8 to 12V) into V+ on J2. Then the board will produce 5V for itself and provide it at the 5V points on J1 and J2. There are also holes where you can tap (or supply) 5V and ground near JP2 marked +5V and GND.

If you wish to use a coaxial power jack, one is included although its use is optional. The jack has three pins, two that line up and one that is offset. Using pliers, bend the offset pinup against the body of the jack. This pin is the same as the pin nearest the open end of the connector and it is customary to use this pin for ground. The bent pin can be used as a ground point for testing or it can be clipped.

The jack can be placed against the – and V+ pins of J2 along the top side of the board. The pins will line up with the holes and you can tack solder the pins (see Figure below). In this configuration, a standard DC “wall wart” with the sleeve grounded (the center pin should have between 8 and 12 volts) will power the onboard regulator.



- Reset – The chip requires a 10K resistor at R1. In addition, the RESET header near IC3 can accommodate a 2 pin header (between pins 1 and 2) so a jumper can reset the board. By default, the GP3X kits, however, have a 5mm tact switch that can install in the holes (pins 1 and 3) to provide a true reset switch.
- Interface – The primary external access to the board's functions is via J1 and J2. The markings on the board explain what each pin does. The 4 corners (marked with –) are ground points. There is another ground on J1 right after the analog channels. The analog channels are marked A0 to A4. The other pins on J1 are the hardware counter (CT), the hardware PWM output (PW), the LED output (LED), and a connection to the 5V supply rail (+5). J2 also has a 5V supply rail connection, an input to the voltage regulator (+V), and the digital pins (D0 to D7).

  You can install headers in these locations either on the top of the board or the bottom (the bottom is handy for plugging into solderless breadboards). In addition, you can use standard “screw terminals” which come in interlocking blocks of 2 or 3 terminals, or use any other type of 12 pin connector with 0.2” spacing. If you ordered the **screw terminals and install them on the bottom of the board** (soldering them on the same side as the silk screen) they will form “feet” that will hold the board up which is handy and also leave the other components on the board more accessible. However, you can also install them on the top of the board if you prefer. In particular, if you install the USB daughter board (which mounts on the bottom) you may want to keep the pins on the top. The choice between top and bottom is purely one of personal preference – the circuit doesn't care either way.

The analog and digital pins route through JP1 and JP2. If you wish, you can cut the traces between the sections of JP1 (digital) or JP2 (analog) and reroute the signals. For example, you might put a series resistor between the holes to limit current. Or you could route a wire from an analog input to an op-amp in the spare IC4 socket and then back from the op-amp to the GP3's CPU. Note that on JP2, the 5th pin (near J1's A3) is not connected through and only connects to an unused GP3 pin.

IC4 is available for use with any circuitry you might want to add unless you are using it for the USB daughter board. An op amp, a temperature sensor, a solid state relay, a DIP switch, or anything else you can arrange to fit in an 8 pin DIP factor can be placed at IC4. The IC4A and IC4B holes connect to IC4 to provide places you can connect your circuitry to the board. In addition, there are convenient holes that carry +5V and ground nearby for your use.

The LED output is brought out to JP1 and also to D1A which normally takes a 5V LED. You can also fit D1B as a normal LED with an appropriate dropping resistor at R2. It is possible to use both LEDs if the output current of the GP3 pin is not exceeded (in which case both LEDs will light or extinguish together). Another option is to install a 5V LED at D1B and omit R1. Then you can use the R1 pad with the circle around it as a connection point. For example, you might wire from the circle pad of R1 to one of the JP1 digital connections to get a second LED output. It is not necessary to install either D1A or D1B for the GP3's operation.

Once you have decided on the options you wish to use, you can assemble the board. In general it is best to start with the “low” components first. IC1's socket and IC2, for example are easiest to install first. Then you can install discrete components like the smaller capacitors, resistors, and LEDs. Large components like the connectors, C6, and IC3 should be installed last.

Construction Notes

- If your kit is supplied with two 24 pin .1 inch headers simply remove every other pin with pliers so that the headers fit in the J1 and J2 slots.
- C1-C4 have polarity markings but it is not necessary to use electrolytic capacitors. If your kit is supplied with ceramic capacitors for C1-C4 you can ignore the polarity markings on the board.
- C6, the LEDs, and the ICs are the only devices that you have to be careful to observe the orientation marked on the board. C6 should have its + or – lead clearly marked and the + side is marked on the board. The longer lead of LEDs will go in the hole marked with the plus sign.
- IC1 and IC2 should be oriented as shown on the silk screen. The silk screen for IC3 shows a thick bar near the edge which is the back of the package (the side without the part markings).
- Because IC4 is uncommitted you can install something in IC4 any way you like (including on the back side of the board) as long as you do the rest of the wiring to IC4 to match.
- Although it is customary to put the body of the resistors in the hole that has the white circle around it, putting one in backwards will not affect circuit operation.
- Although it is customary to put the ceramic resonator or crystal so that the markings face away from IC1, putting it backwards will not affect circuit operation.
- Don't forget that JP1, JP2, IC4A, IC4B, and DTEDCE all have traces connecting pins “across” from each other. If you are making any modifications that require these holes to be electrically separate you must use a hobby knife or razor to carefully cut through the trace on the top side of the board that connects them. You should check that you have made the cut by using an ohm meter before and after the cut. Of course, be careful not to cut other parts of the board.

You should check your work before installing IC1 in its socket. Power the board up using your choice of methods (9V to the +V pin on J2, 5V regulated to any of the +5V pins, or USB power) and verify that 5V appears on the +5V connector near JP2.

Next, proceed to the Final Checkout section below.

Raspberry PI-Compatible Notes

Important Note: Although the GP3X board has pins labeled +5V, when using the Raspberry PI-compatible version, these pins are at 3.3V. It will not hurt the GP3 to power it at 5V, but it may damage the host computer.

In addition, all time-related measurements for the Raspberry PI-compatible version are at 1/2 resolution compared to the normal GP-3 (however, the baud rate is still 57600 baud). So a command that specifies time milliseconds on a normal GP-3 will use 2mS units on the Raspberry PI-compatible version.

The PI-compatible board contains the following components: C5, D1, IC1 (and socket), R1, and X1. In addition there is a 3 pin header and a 2 pin header. Screw terminals are normally supplied.

The board is assembled as usual, ignoring any parts not supplied. A wire jumper must be put in the PGM/RUN jumper going from PGM to the center pin. You should also install a wire jumper (insulated) between pin 1 of the TTLSER connector to any ground point (pin 5 of the DB-9 connector, J3, is handy).

The 3 pin header is placed in the TTLSER pins (pin 2-4) and the two pin header is place in any of the 4 holes marked +5V and GND near the A4 pin on the edge of the board. Place the header so that one pin is in the +5V hole (which is actually 3.3V) and the other is in the GND hole.

To connect to the Pi, connect 3.3V from the Pi (pin P1-1) to the +5V pin on the GP3X. Also connect the ground (pin P1-6). You can do this via the 2 pin header installed or the screw terminals, as you prefer (if you plan to only use the screw terminals, you don't need to install the two pin header).

Connect the PI's P1-8 pin to the GP3X TTLSER pin 3 and P1-10 to TTLSER pin 4. This enables serial communications with the Pi.

The serial port on the Pi is used as the system console. This has to be disabled in order to free the port for use with the GP3. Edit /etc/inittab and find the line that contains /sbin/getty and ttyAMA0. Using a text editor (and your root login) place a # sign at the start of this line and save the file. You will have to reboot the Pi to free the serial port. If you ever want to enable the console, edit the file again and remove the # sign.

For software, see http://www.awce.com/gp3pi.htm for the latest software and demos.

Arduino-Compatible Notes

The Arduino-compatible version only uses the following parts: C5, D1A, IC1 (and socket), R1, and X1. In addition, there is a 3 pin header and a 2 pin header. Screw terminals are normally supplied.

The board is assembled as usual, ignoring any parts not supplied. A wire jumper must be put in the PGM/RUN jumper going from PGM to the center pin. You should also install a wire jumper (insulated) between pin 1 of the TTLSER connector to any ground point (pin 5 of the DB-9 connector, J3, is handy).

The 3 pin header is placed in the TTLSER pins (pin 2-4) and the two pin header is place in any of the 4 holes marked +5V and GND near the A4 pin on the edge of the board. Place the header so that one pin is in the +5V hole (which is actually 3.3V) and the other is in the GND hole.

To connect to the Arduino, you must connect power to the 2 pin header or the terminals marked +5 and – (usually from the Arduino; if you use a separate 5V supply, be sure the Arduino and the GP3 share a common ground). The serial port connects to pin 3 and 4 of the TTLSER connector. Depending on your sketch, you may be able to use any digital pins for the serial port (it depends on what serial library you use). The example programs use pins 8 and 12 on the Arduino. Connect pin 8 to pin 3 of TTLSER and pin 12 to pin 4 on TTLSER.

For software, download the Arduino-compatible library at http://www.awce.com/gp3ard.htm. Unzip the files to your Arduino/libraries directory. Once you restart (or start) the Arduino IDE, you will be able to use the library like any other library. Example code is installed and you can access it via the Arduino Examples menu. In addition, online documentation is available at http://www.awce.com/gp3adoc.

GPMPU28

GPMPU28 Parts List

Designator	*Description*
C1-C4	1 uF capacitor
C5	0.1 uF capacitor
C6	330 uF capacitor
D1	5V LED
IC1	GP3 CPU and 28 pin narrow socket
IC2	MAX232
IC3	7805
JP1	11 pin header; normally installed so the plastic and long pins are on the bottom of the board and soldered on the top of the board.
J1	DB9 connector (female, short reach)
R1	10K resistor
X1	3 pin 20MHz resonator
Reset	(optional) 2 pin header for reset jumper
Vin/Gnd	9V battery snap

The board requires no modifications to work with the GP3 However, you may want to consider any special power supply connections or serial connections you'd like to make. In addition, you'll need to connect the outside world to the GP3 as appropriate for your situation.

If you are prototyping, you may want to install the header at JP1 and use it to connect to a solderless breadboard. For best results, install JP1 on the bottom of the board so that the plastic and long part of the pins are under the board. Then solder the short pins as the protrude through the top side of the board (the side with silk screen markings). You can also solder wires to the holes along the long edge of the board near IC1 to connect to external circuitry or use any other connection on 0.1” centers.

GPMPU28 External Connections

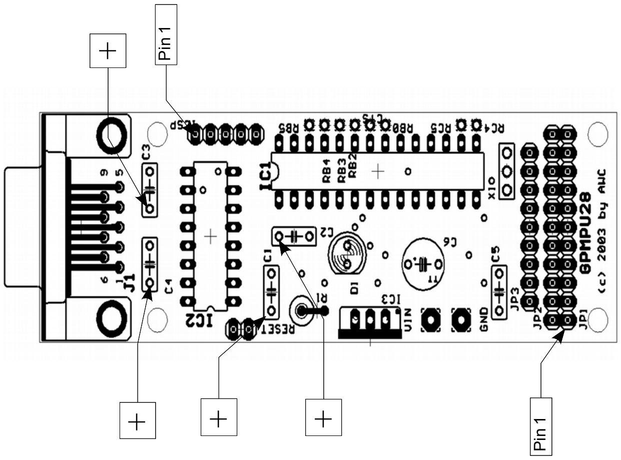
Pin	*Description*
JP1-1	Ground
JP1-2	Hardware PWM output
JP1-3	Ground for normal operation; +5V to run compiled and downloaded script (GP3A1 and above only). Note GP3 chips have a simple LED blinking test program preloaded, so connecting JP1-3 to +5V is a good way to test your GP3.
JP1-4	External count input
JP1-5	Analog input #4
JP1-7	Analog input #3
JP1-8	Analog input #2

Pin	*Description*
JP1-9	Analog input #1
JP1-10	Analog input #0
JP1-11	+5V
RB0	Digital input/output #0
RC4	Digital input/output #1
RC5	Digital input/output #2
RB3	Digital input/output #3
RB4	Digital input/output #4
RB5	Digital input/output #5
ICSP-5	Digital input/output #6
ICSP-4	Digital input/output #7

Board assembly is straight forward. Install the shorter components such as the IC1 socket and IC2 first. Then install the smaller components such as the reset jumper, X1, R1, etc. Finally, install the larger components such as IC3 and J1. You can wire a source of 8-12V to the Vin and GND pins or use the supplied 9V battery connector. Since the 9V battery wires take stress, you should solder both sides of the wire and you may wish to provide strain relief such as a drop of hot glue over the wires.

The only polarity-sensitive components normally placed on the board are the ICs, the LED, and the large electrolytic capacitor. The capacitor is clearly marked as are the ICs (IC3's metal back side is closest to the edge of the board). The flat side of the LED (D1) is the ground side, so the long lead goes in the other hole. Although it is customary to put the body of the resistors in the hole that has the white circle around it, putting one in backwards will not affect circuit operation. Also, it is customary to put the ceramic resonator or crystal so that the markings face away from IC1, but putting it backwards will not affect circuit operation.

Normally, C1-C4 are not polarized capacitors. However, if you use polarized capacitors the negative side of C1, C3, and C4 are all to the right as you read the markings correctly. That is, if you consider the DB9 connector “up” and IC1 is on the right, the negative leads of C1, C3, and C4 are to the right. C2's negative lead is to the bottom using the same orientation (that is, facing the LED).



JP2 and JP3 mirror the inner pins of JP1 (that is JP2-1 and JP3-1 are connected to JP1-2) and are connected together along the top side of the board. If you wish to customize what signals appear on JP1 you can use a hobby knife or razor blade to carefully cut the traces between JP2 and JP3. Then you can use wire to connect JP3 to a different connection, or pick up the signal that used to go to JP1 in the JP2 hole. For example, if you wanted JP1-2 to connect to digital I/O #7 you could cut the trace between JP2-1 and JP3-1 (which correspond to JP1-2) and jumper JP3-1 to ICSP-4.

You should check your work before installing IC1 in its socket. Power the board up using your choice of methods (9V to the Vin connector or 5V regulated to JP1-11) and verify that 5V appears either at JP1-11 or another 5V test point (e.g, the exposed lead of R1 nearest IC3).

Next, proceed to the Final Checkout section below.

GPMPU40

GPMPU40 Parts List

Designator	*Description*
C1-C4	1 uF capacitor
C5	0.1 uF capacitor
C6	330 uF capacitor
C7, C8	Optional. Capacitors matched to X1 if needed.
D1	5V LED
IC1	GP3 CPU and 28 pin narrow socket; install so pin 1 matches pin 1
IC2	40 pin footprint; used for wiring connections in this case
IC3	MAX232
IC4	7805
IC5 (not marked)	An 8 pin socket can be placed so pin 1 matches pin 17 of IC1. An MCP602 may be supplied (see text). This socket is not wired to anything by default so you can place anything you wish here (or nothing).
JP1, JP2	Forms one 40 pin connector with JP1-1 being pin 1 and JP2-1 being pin 21. These are normally installed on the bottom of the board so the board can plug into a solderless breadboard. However, it is possible to install on the top side of the board or use different connectors on .1 inch centers.
J1	DB9 female (right angle; short reach)
R1	10K resistor
R2 (not marked)	Dropping resistor for D1, if required. LED supplied does not require R2.
X1	3 pin 20MHz resonator (supplied) or 20MHz crystal with matching capacitors at C7 and C8
Reset	(optional) 2 pin header for reset jumper
Pgm Run (not marked)	3 pin header and jumper
Vin/J2	9V battery snap or optional coaxial power jack

The GPMPU40 board is not specifically designed for the GP3 – it is simply a general purpose board and you can read more about it in the GPMPU40 manual (see http://www.awce.com/doclib.htm#GPMPU40). The goal is to essentially duplicate the GP3X or GPMPU28 schematics (at the end of this manual) on the board.

Building the GPMPU40 version of the GP3 requires you to install parts and then wire the different areas of the board together (see table below). You can also customize the board in several ways.

The 28-pin IC socket goes in the IC1 footprint with pin 1 aligned. If you wish to use the optional 8 pin socket, place it in IC1 also with pin 1 of the socket lined up with pin 17 of the footprint.

Other required components for the CPU are X1 and R1. Note that X1 is normally a 20MHz 3 pin resonator. If you use a 2 pin resonator or a crystal you'll need matching capacitors (C7 and C8) depending on the crystal you select. R1 is the reset resistor and you should also install a 2 pin header at the RESET holes unless you prefer to wire an external switch to this port (not supplied).

If you are using RS232 (the default for this board) install J1, C1-C4, and IC3. It is possible to connect a UB232R or other USB to serial board into the spare area under IC1 (the optional 8 pin socket).

The board can be powered by a 5V regulated supply (connected to the Vcc holes or the 5V connector on the edge). You can also install IC4 and C6 to provide power on board. In this case you need to supply 8-12V on the

Vin pin (for example, by installing the included 9V battery connector to Vin and the nearby GND terminal). You can also install J2 for a coaxial power connector (ground on the sleeve, 8-12V on the center pin). You should always install C5, regardless.

Be sure to observe all IC polarities as well as the polarity for C6 (and C1-C4 if they are provided as polarized capacitors).

Usually, you'll want to solder two 20 pin headers at J1 and J2. These form a 40 pin connector (J1-1 is pin 1, J2-1 is pin 21). Normally these are soldered on top of the board so the plastic part and the pins are on the bottom. However, if you prefer you can install them on the top side or use other connectors with .1 inch centers.

To complete the GP3 on this board, you can use the following connection guidelines, using wire to make the connections. You may make the connection on the top or bottom of the board (or both). The connection wire the RS232 port, the clock, the reset circuit, the power supply, the LED, and the Pgm Run jumper to the CPU.

GPMPU40 Connections

Connect	*To*	*Required/Optional*
IC2-1	RST (either hole)	Required
IC2-8	Ground (any hole)	Required
IC2-9	CLK (either hold)	Required
IC2-10	CLK (hole not used by IC2-9)	Required
JP1-14	LED (+; long lead)	Optional
JP1-15	LED (short lead)	Optional
IC2-15, IC2-26	1K resistor	Optional (used with normal LED)
IC2-15	Ground (any hole)	Optional (used with 5V LED)
JP4-26	Ground (any hole)	Optional (used with normal LED)
IC2-29	T	Required
IC2-30	R	Required
IC2-31	Ground (any hole)	Required
IC2-32	Vcc	Required
IC2-34	r	Required
IC2-35	t	Required
JP7-1, JP7-2, JP7-3	3 pin header	Required (Pgm Run jumper) Note JP7 and JP8 are not connected to anything else by default, so there is no “wrong way” to wire this header, however it is easiest to think of the pin nearest the JP7 or JP8 marking as pin 1.
JP8-1	Ground	Required
JP8-2	IC2-12	Required
JP8-3	Vcc	Required
JP5-2	JP5-T	Required
JP5-3	JP5-R	Required
JP6-6	JP6-7	Required
JP6-9	JP6-10	Required
J2-1 (if NOT using J2)	IC2-25	Optional (brings unregulated voltage to JP2)
Vin (if using J2)	IC2-25	Optional (brings unregulated voltage to JP2)

JP7 is the run/program jumper. If you connect a shunt from pin 1 to 2 and reset the board, the GP3 will respond to commands from the PC. If you connect the shunt between 2 and 3 and reset, the GP3 will execute its preprogrammed script. By default, the test script blinks the LED (see Final Checkout, below).

Note that JP6 has pin 1 marked. JP6 has 2 long rows and 6 short columns. The numbers go by columns. So the first short row with pin 1 contains pins 1 and 2. The next short row has pin 3 (directly next to pin 1) and pin 4 (directly next to pin 2). Another way to think about this is that the odd numbered pins are all on the long row with pin 1 and the even numbered pins are all on the other long row with pin 2 directly below pin 1.

The connector along the edge (JP1+JP2 counts as one connector with pins 1 to 40) contains all the GP3 signals as seen in the table below.

GPMPU40 Edge Connector

Pin	*Function*
1	Reset
2	Analog #0
3	Analog #1
4	Analog #2
5	Analog #3
6	NOT USED
7	Analog #4
8	Ground
9 and 10	Must be disconnected (clock pins)
11	Hardware counter
12	Program/run jumper (wired to JP8/JP7)
13	Hardware PWM output
14	Onboard LED output
27	Digital I/O #1
28	Digital I/O #2
29-30	Must be disconnected (serial I/O)
31	Ground
32	+5V
33	Digital I/O #0
34-35	Must be disconnected (serial I/O)
36	Digital I/O #3
37	Digital I/O #4
38	Digital I/O #5
39	Digital I/O #6
40	Digital I/O #7

You can use the extra pins in the IC2 foot print to mount an op-amp (MCP602 is optionally supplied; note the MCP602 is designed to operate off 5V so do **NOT** connect Vin to this particular op amp). You can also use this spare area to connect more LEDs, transistors, or any other small device or circuit. The connections will be available at the board edge:

8 pin socket (mounted with pin 1 at IC1's pin 17)	***40 pin connector (J1+J2)***
Pin 1	Pin 17
Pin 2	Pin 18
Pin 3	Pin 19
Pin 4	Pin 20
Pin 5	Pin 21
Pin 6	Pin 22
Pin 7	Pin 23
Pin 8	Pin 24

You could also fit a USB daughterboard in this socket. Refer to the GP3X instructions for more on how this would be possible. However, if you wish to use USB, a better choice would be to use the GP3X board which requires less custom wiring.

You should check your work before installing IC1 in its socket. Power the board up using your choice of methods and verify that 5V appears on the +5V connector.

Next, proceed to the Final Checkout section below.

Final Checkout

If you are using a MAX232, you can also perform an echo test to verify the port circuitry before you install IC1. Connect your PC to the DB9 connector using a straight RS232 cable (assuming you did not modify the DTEDCE jumper on the GP3X; otherwise, use a cross cable). Open a terminal program to the correct port and make sure local echo is off as well as hardware handshaking. The baud rate is not important (use 57600 baud if you like since that is the speed you will use later as well). Typing on the terminal should have no effect at this time. If you see what you are typing you don't have local echo turned off in your terminal program. Short pins 17 and 18 on the IC1 socket and apply power to the board. Now when you type, you should see the characters your type appear. Removing the short should stop your terminal program from showing you what you type. This test will show that the RS232 communications is working correctly. You can perform the same test with the USB daughter board or cable using a USB cable and the appropriate COM port.

Once these checks are complete (or you are feeling lucky) you can fit a jumper on the Pgm Run header so that the center pin connects to the pin closest to the word Run on the GP3X. For the GPMPU40 the jumper at JP7 should have a short between JP7-2 and JP-3. For the GPMPU28 connect JP1-3 to the 5 volt supply (pin JP1-11, for example). Install IC1 and power up the board. The GP3 chip has a program built in that will blink the onboard LED 10 times, pause and repeat. If you did not install either D1A or D1B as built-in LEDs, you can temporarily connect an LED to the LED pin of J1. You can also monitor the output with a voltmeter as the output is slow (about 2Hz).

If the communications test passed and the LED blinks, congratulations, your GP3 is working and you can proceed to the Software section of this manual.

Software

The GP3 connects to a PC or other host device via the RS-232 port. The GP3 is a DCE device, so you can use a straight cable to connect directly to a PC. The PC sends commands to the board using 57600 baud serial data. To make things simpler, you can download libraries that interface your programming language to the GP3 seamlessly. Most Windows-based languages can use ActiveX controls or DLLs, and both are supplied.

To use the GP3 for a particular function, you can either write custom software in a variety of languages, use an application provided (such as GP3DAQ) or use GP3EZ to build a “script” without programming. In addition, GP3 scripts can be downloaded to the GP3 where it can execute without connection to a PC.

You can find more about different techniques to use the GP3 at http://www.awce.com/gp3intro.htm. The current methods available are:

Method	*Description*	*Pros*	*Cons*	*Links*
GP3EZ	Use a simple point and click interface to create "scripts". You can run the scripts from the PC, or download them to the GP3 so it will run standalone.	Very easy to use. Integrated debugging. Only way to compile to the GP3.	Not as flexible as conventional programming. Requires Windows or Mono under Linux (Linux version available May 2010, although V1.5 works under Mono with only a few cosmetic problems)	• Info center including videos, help files, and downloads • Online Help
GP3DAQ	Use an example program to plot analog inputs and save them to your PC as files	No programming required	Windows only. Only plots analog values. Although you can scale data using a polynomial, generally not as flexible as writing custom software or scripts	• More info
Microsoft (ActiveX) languages (C++, C#, VB, VBA, ASP, etc.)	The provided ActiveX control provides an easy way to control the GP3 from most Microsoft languages and many other Windows-based languages.	Simple integration with common Windows languages. Very simple to use.	ActiveX not portable. Must have correct MS dependencies.	• This page has more info (see below). • Core installer has ActiveX and standard DLLs. • Example with VB Express
Windows DLL or Library	Standard library can produce a standard library, a DLL, or you can just include the source code in your C or C++ program.	Portable (same code usable on Linux, Windows, Cygwin, Windows CE). DLL usable with Java via JNI or JNA calls.	Integrating DLLs slightly more difficult in some languages).	• Download library • The Java page has information about using JNA to call the library from Java • Online documentation
Linux Library	Standard library can produce a standard library, a shared object, or you can just include the source code in your C or C++ program.	Portable (same code usable on Linux, Windows, Cygwin, Windows CE). The .so is usable with Java via JNI or JNA calls. Also used for Raspberry-Pi.	Any 3.3V board requires level conversion or GP3/Pi hardware. Serial port can't be used for console.	• Download library • The Java page has information about using JNA to call the library from Java • Online documentation
Pure Java Object	Standard Java Object wraps GP3 functionality	No native code required. Example of embeddable Javascripting.	Requires finicky Java Serial libraries (javax.comm or RXTX)	• The Java object and example • This page has an example.

Java via Library	Using the portable DLL or .so file from Linux is simple	No need for finicky serial libraries	Not "pure" Java	• Download library • The Java page has information about using JNA to call the library from Java • Online documentation
LabView	LabView (by National Instruments) is a very popular program for managing data acquisition and simple control. LabView can easily drive the GP-3 hardware via a serial or USB port.	Popular environment. Interfaces with many other tools.	Not inexpensive for all versions. Experienced programs often prefer working in traditional languages.	• More Info
Gambas (Linux)	Gambas is a VB-like RAD tool for Linux that supports the GP3	Easy visual development	Linux only	• Gambas Example
GNUPlot (Linux)	This article shows a simple way to stream data from a GP-3 to GNUPlot	No real programming; flexible and easy	Linux only; not the fastest solution	• GNUPlot Example
Arduino	Libraries to interface the GP3/A to Arduino. Works with any GP3 board if interfaced to Arduino.	Install like any Arduino library by copying into your libraries folder		• Download • Documentation

Analog inputs read raw counts from 0 to 1023 (0 to 5V with the default reference) and have a maximum recommended source impedance of 10K. If your source exceeds this 10K maximum (or you require a different voltage range), you may wish to buffer the analog inputs with an appropriate op amp circuit.

Reference

The GP3 has a simple command structure. When using the libraries, the commands are even simpler. Each section below describes a command and how to execute it either directly or via the standard libraries. Unless you are trying to write your own library for a different platform, you probably won't care about the direct commands.

When using either ActiveX library, you must set the commport parameter to match the port the GP3 is using (e.g., set to 1 for COM1). In addition, you must set portopen to TRUE to connect to the hardware.

If you are directly controlling the GP3 (and not using the library), you need to know a little bit about how the protocol works. To prevent synchronization errors, each command byte starts with a 0 bit. Each data byte starts with a 1 bit. This requires numbers to be packed to fit. In general a command has the binary format (the raw commands are specified in binary):

0 C C C P P P L

Where C C C is the command code (0 to 7), P P P is typically a pin number (0-7) and L is the LSB of any data byte. If additional data is required, the next byte sent will have the format:

1 N N N N N N N

Since this is only 7 bits, the convention is that the GP3 accepts the value divided by 2. The L bit from the command makes the entire 8 bit number. All communications are at 57600 baud, 8 bits, 1 stop bit, and no parity.

In addition, there is a 16-bit argument (ARG) that can be set by special commands. Some commands use this argument word as an additional source of data. When the GP3 sends data to the PC, it is not encoded in any particular way since the GP3 only responds to queries from the PC there is no chance of desynchronizing. All data back and forth is in binary (although the results from a raw inp command happen to be an ASCII "0" and "1").

By default, all digital I/O pins are inputs until you change them explicitly to outputs. Hardware handshaking allows you to determine when the GP3 is busy. The GP3 will remain busy while performing any operation, although hardware PWM and the hardware counter works without tieing up the GP3.

For the purposes of this manual, a byte has 8 bits and a word has 16 bits. Raw commands are shown in binary. Each raw command is a single byte, not 8 ASCII characters.

Common Commands

These commands are provided in the standard ActiveX libraries. In addition, each command discusses the raw command bytes you would send to duplicate the command if you are writing your own libraries.

high(pin), low(pin) – These commands are the easiest way to manipulate the output pins of the GP3. These commands force the pin (from 0 to 7) to be an output and set the logic state to high, low respectively. *Raw command: 0 0 0 1 P P P H where P P P is the pin number and H=0 for low or 1 for high.*

toggle(pin) – Sets an output pin (from 0 to 7) to an output and inverts its current state. *Raw command: 0 0 1 0 P P P 0 where P P P is the pin number.*

inp(pin) – Switches a pin to an input and reads the value. The raw command version returns an ASCII "0" or ASCII "1". The library versions return 0 or 1 as you would expect. *Raw command: 0 0 1 0 P P P 1 where P P P is the pin number.*

tris – This property sets the direction bits for the 8 digital I/O pins. Each 1 in this byte sets the corresponding pin to an input (the default). A 0 sets the corresponding pin to an output. You only need to use this property if you want to manage the pin directions all at the same time (in other words, you don't need to use tris if you are making calls like high, low, etc.). As an example, if you wanted to set all pins to inputs except for pins 0 and 1 you might write:

```
        io.tris = &HFC   ' FC hex = 11111100
```

Raw commands: To read use 0 1 0 1 0 1 1 0; to set use 0 0 0 0 0 0 1 L where L is the LSB of the value to set.

pins - This property sets and reads the I/O pins as a single byte. You only need to use this property if you want to manage the pin data all at the same time (in other words, you don't need to use pins if you are making calls like high, low, etc.). As an example, if you wanted to set all output pins to 0 except for pins 7 and 1 you might write:

```
        io.pins = &H7D   ' 7D hex = 01111101
```

Naturally, in this case, the command only affects pins already set as outputs.
Raw commands: To read use 0 1 0 1 0 1 1 1; to set use 0 0 0 0 0 1 0 L where L is the LSB of the value to set.

a2d – This function accepts an A/D channel number (from 0 to 4) and returns a 16-bit number that contains the 10-bit A/D result. The first byte sent by the GP3 is the top 8 bits of the result. *Raw command: 0 0 1 1 P P P 0 where P P P is the A/D channel number*.

a2dconf – Configure the A/D for result justification and voltage reference selection. The function call allows you to set the justify parameter to get results right or left justified (default is left and has value 0; a 1 selects right justification). The cselect argument can be 0, 1, or 4. The default, 0, uses ground and the supply voltage as references. A value of 1 selects analog input 3 as the positive reference (the negative reference is still ground in this case). A value of 4 uses analog input 2 as the negative reference and analog input 3 as the positive reference. This is useful for expanding the sensitivity of the A/D converter. So using 0V and 3.3V as references would mean that a count of 1023 would be equal to 3.3V instead of 5V. Note that the minimum reference voltage allowable is 2.7V. Also note that the clkx2 argument is no longer used, but remains for historical reasons. *Raw command: 0 0 1 1 0 0 0 1   1 J 0 0 0 R R R where J is the justification bit and RRR is the cselect argument.*

eeprom – This property allows you to access the 128 bytes of EEPROM available in the GP3. Keep in mind that EEPROM eventually wears out (about one million write cycles) so EEPROM is best for storing values that don't change frequently such as serial numbers or configuration parameters. For the library, simply provide the address as a parameter: eeprom(0)=41. When using raw commands, the address comes from ARG and the data byte to store is sent after the command (or read, if reading). *Raw command: to set – 0 1 0 1 0 0 0 L; to read – 0 1 0 1 0 0 1 0 where L is the LSB of the data byte.*

pwm – This function has two different capabilities. If you supply a pin number, a duty cycle, and a duration, the GP3 will produce a PWM wave form with the indicated duty cycle on any of the 8 general I/O pins. This will force the pin to output status and also tie up the GP3 for the indicated duration. If you supply a pin number of -1, the GP3 will set its hardware PWM output to the indicated duty cycle and use the duration as a frequency. The hardware PWM output runs continuously without stopping the GP3. The frequency can be as low as 1225 Hz or as high as 32767Hz. In either case, the duty cycle runs from 0 to 255 (0 to 100%). *Raw command: software PWM – 0 1 1 0 P P P L; hardware PWM – 0 1 0 1 0 1 0 L where P P P is the pin number and L is the* *least significant bit of the duty cycle. Send the duty cycle as the second byte. The duration/frequency is in ARG.*

led – This property allows you to turn the onboard LED on or off. *Raw command: 0 0 0 0 1 1 0 X where X=0 to turn LED off and X=1 to turn LED on.*

count - The count command allows the GP3 to count the number of low to high pulses that occur on an input pin over a particular period of time. This is not the hardware counter, but instead requires the GP3 to watch the input pin (the pin is forced to be an input). The duration (the argument when using raw commands) is specified in milliseconds. The GP3 samples the pin every 4uS for that period and returns the count as a 16-bit integer (most significant byte first). *Raw command: 0 1 0 0 P P P 0 where P P P is the pin number.*

rctime – This command allows the GP3 to measure the time a pin (forced to be an input) stays in a particular state. This is useful for measuring the charge or discharge time of an RC network (which can be used to measure a potentiometer or a thermistor, for example). The time returned (a 16-bit integer with most significant byte first) indicates the time in 2uS units. *Raw command: 0 0 0 0 1 1 1 0, 1 0 1 0 P P P S where P P P is the pin number and S is the desired state.*

pulsein – Use this function to read a single pulse on one of the 8 digital I/O pins (the pin is made an input). The return value (16-bits, most significant byte first) is in 2uS units (e.g., a return value of 100 indicates 200uS). The state argument indicates if you want to measure a low pulse (0) or a high pulse (1). If there is no pulse within 131.07mS, the command times out and returns 0. *Raw command: 0 1 1 1 P P P S where P P P is the pin number and S is the pulse state.*

pulseout – Generate an output pulse on one of the 8 digital I/O pins (the pin is made an output). The duration (in 2uS units) uses ARG in raw command mode. The pulse is formed by inverting the current state of the output pin for the time specified. *Raw command: 0 1 0 0 P P P 1 where P P P is the pin number.*

freq – This command generates a simulated sine wave (using PWM) on one of the 8 digital I/O pins. The duration is specified in milliseconds and the frequency in Hertz. For raw commands, ARG indicates the duration in milliseconds. The frequency in Hertz (up to 32767) is sent as two bytes. The first byte is the most significant byte with bit 7 set (the GP3 ignores this bit). The second byte is the least significant byte shifted right one place and also with bit 7 set. The L bit is the least significant bit of the least significant byte. *Raw command: 0 0 0 0 1 1 1 0, 1 0 0 0 P P P L where P P P is the pin number and L is the least significant bit of the frequency's least significant byte.*

setcounter – The GP3 has a hardware counter. This command sets the mode for the counter. In particular, you can select a prescale factor of 1, 2, 4, or 8. You can also select the internal 5MHz clock as a counter source (set the mode=0 for internal clock). The counter is 16-bits, so at 5MHz and a prescale of 8, for example, it rolls over at roughly 105mS. When counting external pulses, the hardware counter counts high-going pulses. When using the library, simply specify the prescale number (e.g., 8). Setting the counter resets the count to zero. *Raw command: 0 0 0 0 1 1 1 0, 1 0 0 1 0 V V X where V V is the prescale factor (0 0 = 1:1, 0 1 = 1:2, 1 0 = 1:4, 1 1 = 1:8).*

counter – This command reads the current counter value and resets the counter to zero. The 16-bit result is sent most significant byte first. *Raw command: 0 1 0 1 1 0 1 0.*

resetall – Use this command to return the GP3 to its initial state (all digital I/O as inputs, counter off and reset, PWM off). *Raw command: 0 1 0 1 1 1 0 0.*

repeat – You may wish to repeat certain input operations as rapidly as possible. If this is the case, set ARG to the number of times you want to execute the command (minus 1 if using raw commands) and call the repeat function. Then the next command will return that number of results. The repeat command works with a2d, inp, rctime, and pulsein commands. It also works when reading pins or tris. When using the library, use readword or readbyte to retrieve the additional results. The command does not change ARG. Once a repeating command executes the repeat count, which is independent of ARG, is reset to 0 (single operation). *Raw command: 0 1 0 1 1 0 1 1.*

Library-only Commands

Several of the commands available in the library don't correspond to GP3 commands.

commport – Use this property to set the correct COM port you are using. This is a numeric value, so to use COM4, set this property to 4.

portopen – This property must be set to TRUE to connect to the hardware. If set to FALSE, the library will disconnect and no I/O commands will work.

ready – This function returns TRUE if the GP3 is ready to accept more commands.

ok – This function returns TRUE if the GP3 is responding.

errflag – If an error occurs, this flag will be set to true.

readbyte – Reads a byte from the GP3. This is useful when using a repeating command that returns bytes.

readword – Reads a word from the GP3. This is useful when using a repeating command that returns words.

Direct-only Commands

There are several commands the GP3 recognizes that are not exposed by the libraries directly.

setarg – To set ARG, there are three different commands used:

> 0 0 0 0 0 1 1 L: Set a byte into ARG (top word is set to 0). The byte has the usual format (LSB in bit L).
>
> 0 0 0 0 1 0 0 L: Set a word into ARG where the least significant bit is 0. The L bit represents bit 8. So to send &H306 you'd send 00001001 10000001 10000011.
>
> 0 0 0 0 1 0 1 L: Set a word into ARG where the least significant bit is 1. The L bit represents bit 8. So to send &H307 you'd send 00001001 10000001 10000011.

increment/decrement – You can use the 0 1 0 1 1 0 0 L command to increment (L=1) or decrement (L=0) ARG.

reset – The 0 0 0 0 0 0 0 0 command performs a quick reset that does not change any state, but resets the GP3's communication state.

check – Send a 0 0 0 0 0 0 0 1 command and the GP3 will return &H33.

Quick Reference

Methods/Functions

Name	*Parameters*	*Return*	*Description and Notes*
high	pin	none	Set output pin to high Automatically sets pin to output
low	pin	none	Set output pin to low Automatically sets pin to output
toggle	pin	none	Reverse state of output Automatically sets pin to output
inp	pin	Byte (ASCII 0 or 1)	Read input pin Automatically sets pin to input
a2d	channel	word	Read analog channel Returns 0 to 1023
a2dconf	justify, clk2x, reference	None	Set 1 to left justify result Set reference = 0 for supply rails, 1 for analog 3 used as + reference, or 4 for analog 2 as – reference and analog 3 as + reference (note: clk2 parameter not used)
pwm	channel, duty_cycle, duration/freq	none	Output pulse width modulation If channel = -1, use hardware PWM with specified frequency; otherwise use software PWM for specified duration Automatically makes pin an output
count	pin, duration	word	Counts pulses on specified pin for duration (in mS or 2mS for Pi) Automatically makes pin an input
rctime	pin, state	word	Measures time pin stays in state (2uS resolution; 4uS for Pi) Automatically makes pin an input
pulsein	pin, state	word	Measures pulse (2uS resolution; 4uS for Pi) Automatically makes pin an input
pulseout	pin, duration	none	Creates pulse on output pin for specified duration (2uS resolution; 4uS for Pi) Automatically makes pin an output
freq	pin, frequency, duration	none	Make pin an output and synthesize sine wave Frequency in Hz (2 Hz for Pi), duration in milliseconds (2 mS units for Pi)
setcounter	prescale, mode	none	Set prescaler (1, 2, 4 or 8) and timer/counter mode
resetall	none	none	Reset GP3
repeat	count	none	Repeat next command
readbyte	none	byte	Read bytes from repeated command
readword	none	word	Read word from repeated command

Notes:

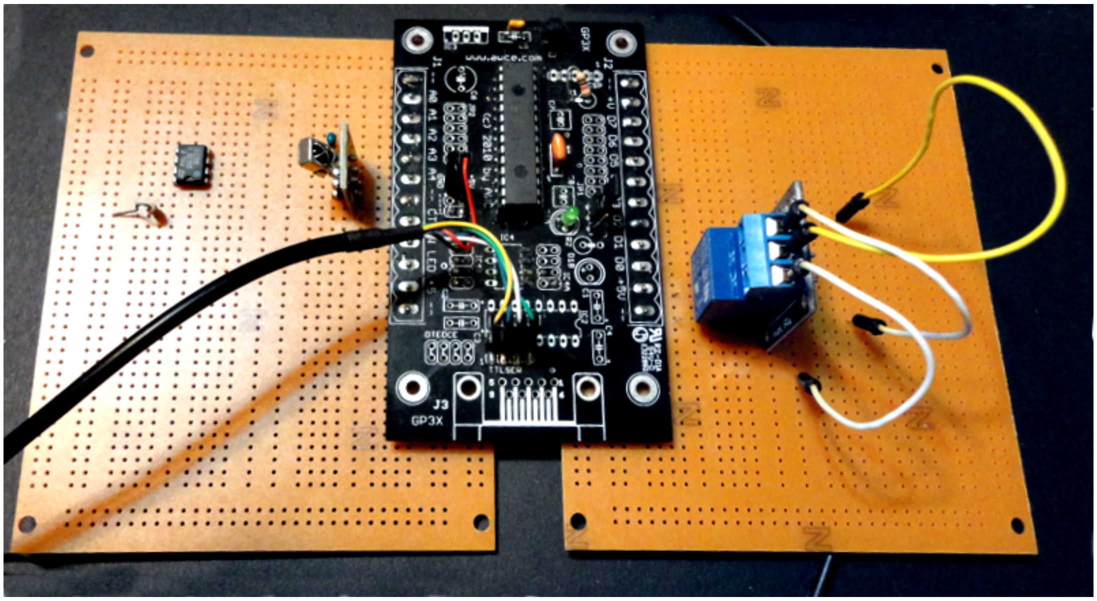
Pin numbers are from 0 to 7
A/D channels are from 0-4

Additional Prototype Space ("Wings")

AWC offers additional "wings" you can mount on either (or both) sides of the GP3-X to gain additional space to prototype. The boards are universal PCBs and, of course, you can use other boards or use the wings with other GP3 variants.

The wings have connector patterns around the edges that have two holes connected together. They also have 8 individual bus bars that have many holes connected together. These are perfect for grounds, power, or other widely distributed signals. There are also 4 groups of "breadboard" holes each with 4 holes connected together. These may be used in pairs to hold DIP ICs or hold other components.

Generally, if you want to use wings, you will order the GP3-X with pins instead of screw terminals. The pins will mount on the bottom of the PCB (so you will solder them from the top). Then you can solder the wings to pins on the sides you want to add prototyping space. While you can put them anywhere, it is best to put the pins in the edge of the first group of 4 holes in a row (the "breadboard" holes). The pins will consume one column of of the holes and the edge of the board will cover up another column. This leaves two columns free for wiring or probing. As an alternative, you place the pins right on the edge pads of the wing, however, the top of the adjacent hole will become inaccessible. The figure below shows a GP3-X with two wings (and some user-supplied components):

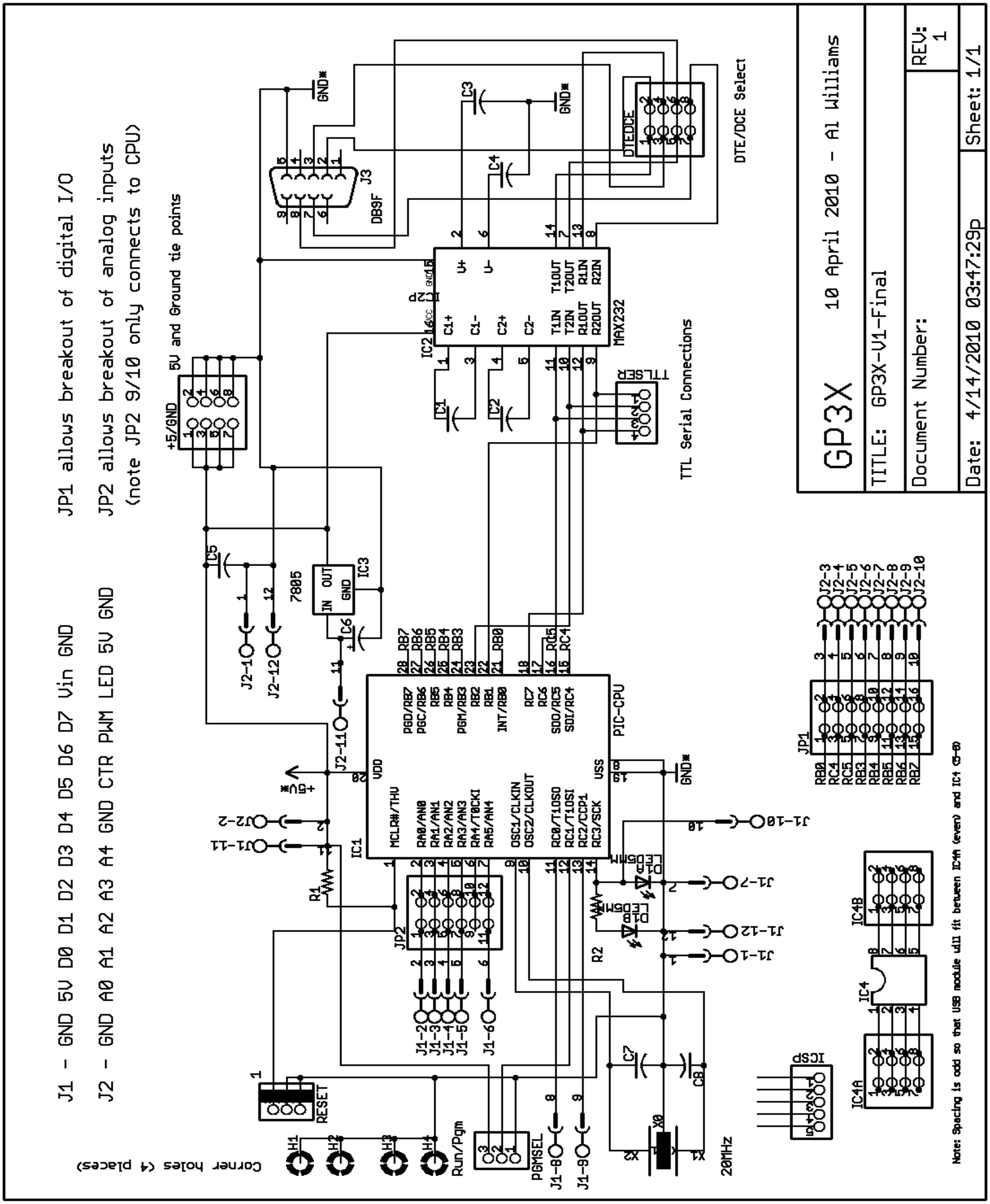


If you need additional mechanical stability, the wings can be drilled to have mounting holes that align with the ones on the GP3-X board.

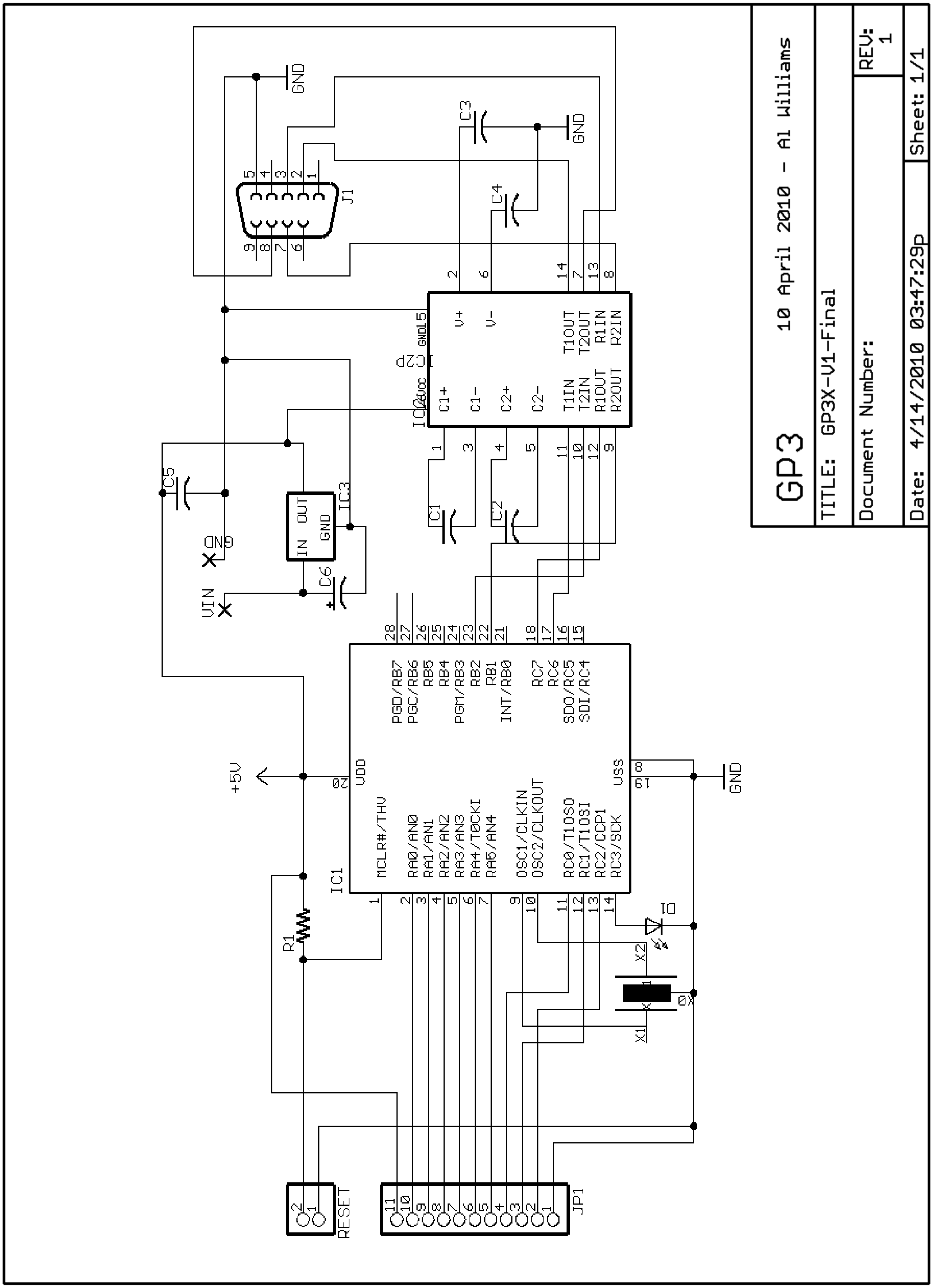
Specifications

Maximum total current sunk or sourced by all I/O pins	90mA
Supply voltage	4.5 to 5.5V (3V to 5.5V for Pi)
Supply current (with regulator, no output)	12mA nominal (RS232 installed, no IC4 installed)
Input low voltage (maximum)	0.8V
Input high voltage (minimum) 2.0V	
Input leakage current (maximum)	+/- 1uA
Output low voltage (maximum)	0.6V
Output High voltage (minimum)	Vdd-0.7V
EEPROM endurance	(minimum) 100,000 write cycles
A/D resolution	10 bits
A/D Integral error	+/- 1LSB
A/D Differential error	+/- 1LSB
A/D Offset error	(maximum)
A/D Vref minimum	2.7V
Recommended analog input impedance	10k ohm
Vref input current (maximum)	1000uA

Schematic (GP3X)



Schematic (GPMPU28)


GP3
10 April 2010 - Al Williams
TITLE: GP3X-V1-Final
Document Number:
REV: 1
Date: 4/14/2010 03:47:29p
Sheet: 1/1
J1
GND
C3
C4
IC2P
IC2
V+
V-
C1+
C1-
C2+
C2-
T1OUT
T2OUT
R1IN
R2IN
T1IN
T2IN
R1OUT
R2OUT
GND
VCC
C1
C2
C5
C6
IC3
IN
OUT
GND
VIN
+5V
IC1
VDD
VSS
MCLR#/THV
RA0/AN0
RA1/AN1
RA2/AN2
RA3/AN3
RA4/T0CKI
RA5/AN4
OSC1/CLKIN
OSC2/CLKOUT
RC0/T1OSO
RC1/T1OSI
RC2/CCP1
RC3/SCK
PGD/RB7
PGC/RB6
RB5
RB4
PGM/RB3
RB2
RB1
INT/RB0
RC7
RC6
SDO/RC5
SDI/RC4
R1
D1
X1
X2
X0
RESET
JP1