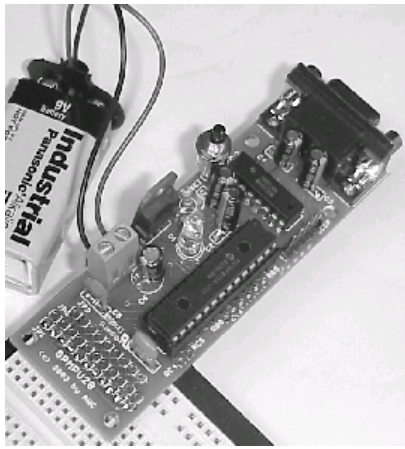
AWC

GP3a PC Analog/Digital I/O Kit



AWC
310 Ivy Glen
League City, TX 77573
(281) 334-4341
http://www.al-williams.com/awce.htm
V1.6 31 Dec 2007

Table of Contents

Overview

The GP3 is the easiest way to interface a PC to the real world. 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 or 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
- ActiveX control (OCX) and DLL supplied; or control via serial protocol (57600 baud)

If You Need Help

If you require assistance, please feel free to contact us. The best way to get support is via e-mail (stamp@al-williams.com). However, you may also call between 9AM - 4PM Central Time at (281) 334-4341. Be sure to check out our Web page for updates at www.awce.com.

Building

Please follow the directions included for building the GPMPU kit. 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. You can also solder wires to the holes along the long edge of the board near IC1 to connect to external circuitry.

For a permanent installation, you'll want to omit the header at JP1 and simply make the connections to fit your installation.

In any event, make your connections according to this chart:

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 GP3As 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
JP1-9	Analog input #1
JP1-10	Analog input #0
JP1-11	+5V
RB0	Digital input/output #0

Pin	*Description*
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

Operation

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.

Analog inputs read raw counts from 0 to 1023 (0 to 5V) 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 tying 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, this only affects pins that are 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*.

eeprom – This property allows you to access the 128 to 256 bytes of EEPROM available in the GP3 (the GP3A1 and A2 have 256 bytes). 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 pint 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) and X=0 for internal or 1 for external).*

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.* "A" firmware can use raw command: 0 1 0 1 1 1 1 0 to read the counter with no reset.

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.

- 0 1 0 1 1 1 1 0: Read counter without reset

- 0 1 0 1 1 1 1 0: Read extended version code (&H34 for GP3A1; &H35 for GP3A2; times out for GP3).

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

Properties

Property	Notes
tris	Determines the status of each digital I/O pin; 1=input, 0=output
pins	Read or write all digital I/O pins at once
led	Controls the onboard LED (write only)
eeprom	Reads or writes from the 128 (or 256 for GP3A2) EEPROM cells (numbered 0 to 127) – example: eeprom(2)=41.
counter	The hardware timer/counter (read only)
commport	Serial port to use
portopen	Set to TRUE to connect to the GP3 board
ready	Returns TRUE if GP3 is not busy (read only)
ok	Returns TRUE if GP3 is present and responding (read only)
errflag	Set to TRUE if error occurs

Note: EEPROM addresses range from 0 to 127 for original GP3 and GP3A1. GP3A2 has a range of 0-255.

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
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) Automatically makes pin an input
rctime	pin, state	word	Measures time pin stays in state (2uS resolution) Automatically makes pin an input
pulsein	pin, state	word	Measures pulse (2uS resolution) Automatically makes pin an input
pulseout	pin, duration	none	Creates pulse on output pin for specified duration (2uS resolution) Automatically makes pin an output
freq	pin, frequency, duration	none	Make pin an output and synthisize sine wave Frequency in Hz, duration in milliseconds
setcounter	prescale, mode	none	Set prescaler (1, 2, 4 or 8) and timer/counter mode (1=ext, 0=int)
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

Example Visual Basic Code

The available libraries include an ActiveX control and an ActiveX DLL. The control is easy to use when you have a form-based program. The DLL is easier to use in cases where you don't have a form. Here is a simple example of using a control named IO that references a GP3 ActiveX control.

```
Private Sub Button1_Click()
IO.commport=7      ' set comm port
IO.portopen=true   ' connect
IO.led=True        ' turn on LED
tmp=IO.a2d(0)      ' read analog channel #0
IO.portopen=false  ' disconnect
End Sub
```

To use the ActiveX DLL, use AWCGP3DLL.GP3DLL as the PROGID (or add a reference to the DLL in your project).

Check out http://www.awce.com for more free software to use with your GP3 including the GP3DAQ plotting application and GP3EZ, the control software that lets you create GP3 applications with no programming.

Tips

- You can wire a Molex or other similar connector to the I/O holes to make a neat, removable assembly.

- If you don't need all the analog inputs, it is easy to use them to monitor slowly changing digital inputs like push buttons switches. Simply treat anything below, say, 511 as a logic 0 and anything 512 or above as a 1.

- The hardware PWM output is excellent for use as a digital to analog converter. Simply filter the output with an RC filter. Since the output runs continuously, the output will remain stable.

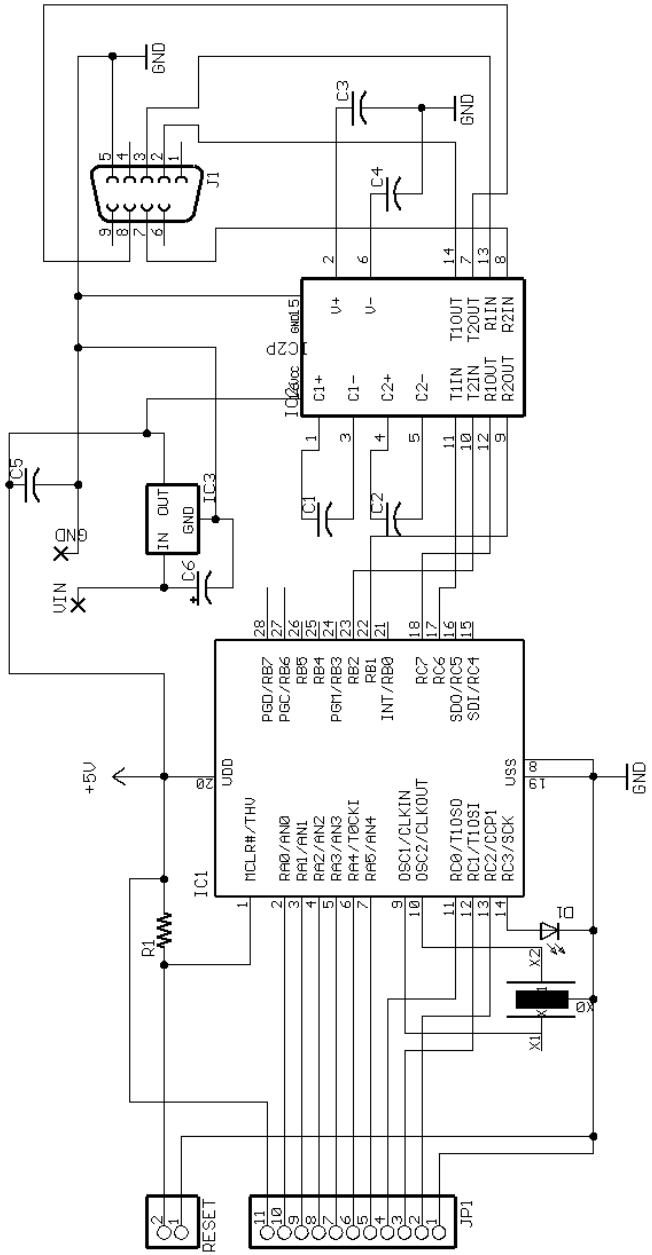
- In addition to the ActiveX library, there is also a standard DLL that you can call from many languages. The entry points are similar to the ActiveX calls (see the README.TXT file included with the DLL for details).

- It is possible to use the GP3 IC in a GPMPU40 board to provide additional area for op amps, relays, and improved connectivity through the board's edge connector.

- Check our Web site for sample projects and the latest library files. See http://www.awce.com/gp3.htm.

Notes

Schematic


GND
C3
GND
C4
J1
IC2
C1+
C1-
C2+
C2-
V+
V-
T1IN
T2IN
R1OUT
R2OUT
T1OUT
T2OUT
R1IN
R2IN
C1
C2
C5
C6
IC3
IN
OUT
GND
VIN
VOUT
IC1
MCLR#/THV
VDD
VSS
RA0/AN0
RA1/AN1
RA2/AN2
RA3/AN3
RA4/T0CKI
RA5/AN4
OSC1/CLKIN
OSC2/CLKOUT
RC0/T1OSO
RC1/T1OSI
RC2/CCP1
RC3/SCK
RB0
RB1
RB2
RB3
RB4
RB5
RB6
RB7
RC7
RC6
RC5
RC4
SDO
SDI
PGM
PGC
PGD
+5V
R1
LD
X2
X1
RESET
JP1