

**AWC**

**GP7 Two Wire Serial  
I/O Kit**

© 2004 by AWC

AWC  
1279 FM 518 #2  
Kemah, TX 77565  
(281) 334-4341

<http://www.al-williams.com/awce.htm>

V1.0 29 June 2004



# Table of Contents

Overview .....	1
If You Need Help .....	1
Building .....	1
Operation .....	4
Reference .....	5
ActiveX Commands .....	5
ASCII Commands .....	7
Example Visual Basic Code.....	8
Notes.....	<b>Error! Bookmark not defined.</b>
Notes.....	10



## Overview

The GP7 provides a simple RS232 interface to industry-standard two-wire serial integrated circuits. In addition, the GP7 provides additional digital and analog I/O plus an ActiveX control to make programming simple. Features include:

- Connects to common two-wire serial EEPROMs, A/Ds, DACs, etc.
- Slave addressing allows multiple chips to connect to the same bus
- PC acts as master
- Compatible with I2C, IIC, SMBUS, and similar protocols
- 8 digital I/O ports available
- 4 10-bit analog inputs available
- Control via serial protocol (57600 baud) or use included ActiveX control

## 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](mailto:stamp@al-williams.com)). However, you may also call between 9AM - 4PM Central Time at (281) 334-4341. You can also fax to (281) 754-4462. Be sure to check out our Web page for updates at [www.al-williams.com/awce](http://www.al-williams.com/awce).

## Building

Please follow the directions included for building the GPMPU40 kit. The board requires no modifications to work with the GP7. 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 GP7 as appropriate for your situation.

If you are prototyping, you may want to install the headers at JP1 and JP2 and use them to connect to a solderless breadboard. You can also directly connect these wires, if desired.

The GPMPU40 requires several jumpers to configure it for use with the GP7 IC. Please make your connections according to this chart:

Connection 1	Connection 2	Note
JP5-T	JP5-2	RS232 connection
JP5-R	JP5-3	RS232 connection
RS-T	IC1-3	TX
RS-R	IC1-2	RX
RST (either pin)	IC1-1	Reset
Vcc	IC1-7, IC1-32 (analog Vcc)	+5V
Ground	IC1-8, IC1-34 (analog ground)	Ground
CLK (either pin)	IC1-9	Clock
CLK (either pin)	IC1-10	Clock

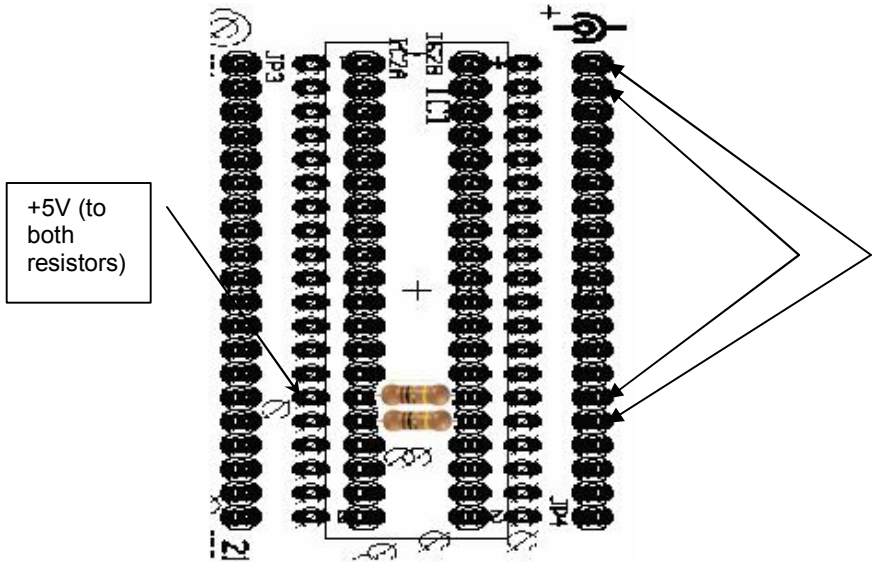
You may wish to connect external circuitry through JP1 and JP2 or you can make a customized 6-pin connector by installing jumpers to JP8 and using JP7 to connect your circuit. You can also install an 8 pin IC of your choice at the bottom of IC1 and wire it according to the 8 pin IC's pinout, if desired.

Here are the important external connections:

<i>Pin (counting 40 pins of IC1/2 footprint)</i>	<i>Description</i>
6	P4 (digital I/O bit 4)
7	+5V
8	Ground
11	P5 (digital I/O bit 5)
12	P6 (digital I/O bit 6)
13	P7 (digital I/O bit 7)
14	P0 (digital I/O bit 0)
27	P1 (digital I/O bit 1; IC1 pin 15)
28	P2 (digital I/O bit 2; IC1 pin 16)
29	P3 (digital I/O bit 3; IC1 pin 17)
32	+5V (Analog Vcc)
33	Analog reference (decouple to Analog ground if using internal reference, or connect to external reference). Reference can accept 2V to Analog Vcc and presents about 32K ohms.
34	Analog ground
35	Analog input 0
36	Analog input 1
37	Analog input 2
38	Analog input 3
39	Serial data (SDA)
40	Serial clock (SCL)

Some serial busses require pull up resistors on the bus lines. The GP7 includes two 4.7K resistors, although you should verify that these values are suitable to use with your application.

Installing these pull ups will depend on your particular application. We recommend using the row of pins at IC1/IC2 pins 15 and 16. Place one resistors across the inside pins from pin 15 and 21. Then connect the +5V supply to one side of the resistors (using the outer pins) and connect the other sides to pin 39 and 40 (SDA and SCL):



## Operation

The GP7 connects to a PC or other host device via the RS-232 port. The GP7 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 the ActiveX control supplied.

Analog inputs read raw counts from 0 to 1023 (0 to 5V or the reference voltage). Each analog input presents 55 to 100 megaohms of input impedance.



## Reference

The GP7 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 port parameter to match the port the GP7 is using (e.g., set to 1 for COM1). In addition, you must set portopen to TRUE to connect to the hardware.

By default, all digital I/O pins are inputs until you change them explicitly to outputs. If you set the output bit high on a pin configured as an input, you will enable a weak pull up resistor on that pin.

### **ActiveX 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.

**Port** – Selects the COM port used for the GP7. For example, 1 is COM1.

**PortOpen** – Set to **TRUE** to open the GP7's communication channel. Set to **FALSE** to release the communications channel. Note that this parameter must be set **TRUE** before accessing any of the device functions.

**Address** – This property stores the target slave address for the Send and Receive routines.

**Data** – This array property holds data for the Send and Receive routines. The maximum size for received data is 255 bytes.

**Send** – Send the data in the **Data** array to the slave at **Address**.

**Receive(n)** – Receive data from the bus at the given slave **Address**. First, the GP7 sends the data from the **Data** array (if any) and then fills the **Data** array with **n** bytes of data.

**BusStop** – Sets a stop condition on the serial bus.

**A2D(n)** – Reads from analog port **n**. The return number is from 0 to 1023 where 0 is 0V and 1023 is the analog reference value.

**ARef** – Set the voltage reference to 0 (external; default), 1 (internal Analog Vcc; nominally 5V), or 3 (internal 2.56V). Note that to use either internal reference, you must not feed a voltage to the analog reference pin (you may, however, use a .1uF bypass capacitor from analog reference to analog ground). The 2.56V reference can vary from 2.3 to 2.7V.

**Dir** – This property sets which digital I/O pins are inputs and which pins are outputs. Each bit in the byte-sized property represents one I/O pin. A 1 makes the corresponding pin an output. For example, setting **Dir** to &H81 will make pin 7 (&H80) and pin 0 (1) outputs, while setting other pins to inputs.

**IO** – Reading this property reads the input state of the digital I/O pins. Writing to this property sets the output state of the digital I/O pins. Note that writing a 1 to a bit that is set as an input will enable a weak pull up resistor on that pin.

**Reset** – This method attempt to reset the GP7 and sets **Dir** and **IO** to zero. It also sets **ARef** to 0 and **BitRate** to 100,000 Hz.

**BitRate** – This property sets the nominal frequency of the serial bus in Hertz. Notice that although you can set any frequency you want, the hardware only changes the speed in units of 10,000 Hz, so setting the speed below 10,000 Hz has no effect. In addition, setting the speed to, say, 30,000 Hz, and then changing to 35,000Hz will not have any real effect. The normal (default) rate is

100,000 Hz. Most devices will not work with speed above 400,000 Hz. Note that higher speeds may require reduced bus resistors and other special design considerations.

### ***ASCII Commands***

Each ASCII command ends with a carriage return. All hex numbers are encoded as ASCII bytes, so FF is two characters, not one byte. Characters are not case sensitive. All error codes are two characters starting with X and ending in a carriage return (XS, for example, is an error while sending).

? – Returns a check code, 2C, followed by a carriage return.

B – Sets bit rate. Requires a single hex byte that represents the speed in 10kHz units. So to set the speed to 100kHz, you'd send "B0A". Note that the GP7's RS232 rate is always 57600 baud.

S – Send data. You must include a slave address, and as many bytes as you wish to send (up to 127). So to send a zero to slave address A0, you would send "SA000". The return is either OK, or an error code beginning with X.

R – Receive data. You include a slave address, a receive count, and any bytes you wish to send to the slave first. You do not need to set the read bit in the slave address. So to read two bytes from a device at address A0, you might send "RA002". Returns a string of hex digits or an error code starting with X.

X – Stop. Sets a stop condition on the bus. You should follow each S or R transaction with this command unless you want to issue multiple commands as a single transaction. Do not send a stop unless you've already issued an S or R. Returns OK.

D – Set direction register. A one in the direction register makes the corresponding P pin an output. So "D81" would make P7 and P0 outputs, leaving the other 6 P pins as inputs. Returns OK.

I – Read digital inputs. Reads a byte of input.

O – Write digital outputs. Writes a byte to the output pins (note that pins that are inputs will use the output to turn on a weak pull up resistor). So “OFF” would turn on all outputs or pull up resistors (depending on the state of the direction register).

V – Select voltage reference. By default, the GP7 uses an external reference on the reference input pin (condition 0). However, if you do not connect to this pin, you can select an internal 2.56V reference (condition 1) or the internal 5V supply (condition 3). So to select a 5V internal reference, you’d send “V3”. Note this command takes a single digit from 0, 1, or 3, and returns OK.

A – Read an analog voltage. This command returns a number between 000 to 3FF where 000 is zero volts, and 3FF is the reference voltage. So to read channel 3, you’d issue “A3”.

## Example Visual Basic Code

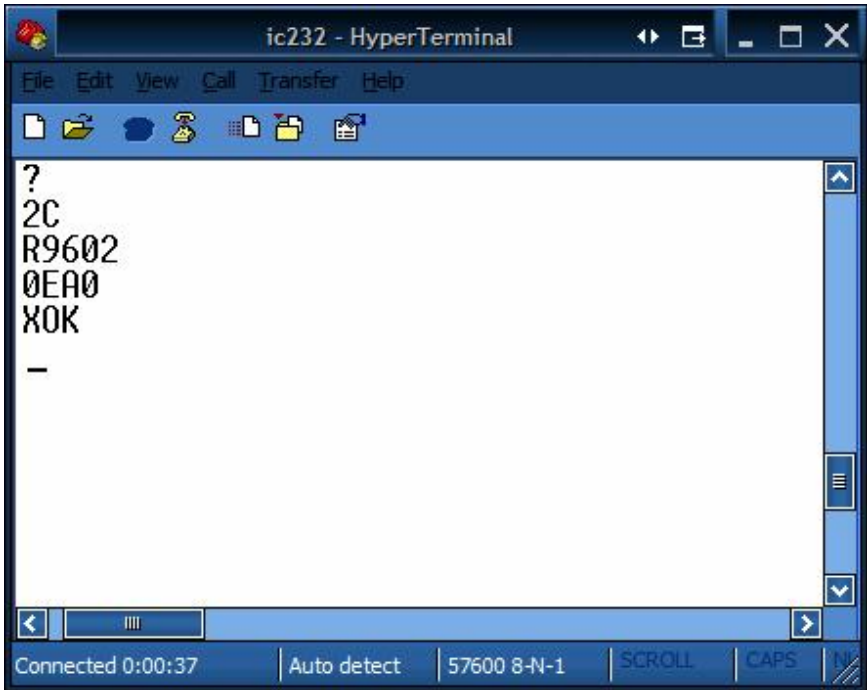
The available libraries include an ActiveX control. Here is a simple example of using a control named IO that references a GP7 ActiveX control.

```
IC232A.Port = 1
IC232A.PortOpen = True
IC232A.Dir = &H81
IC232A.IO = 1 ' Turn on P0
IC232A.Address = &HA0
IC232A.Data = Array(0,0,&H55)
IC232A.Send
IC232A.BusStop
```

Of course, you have to add the GP7’s ActiveX control (IC232 to your project) and create a control named IC232A in your program. For more examples, please visit our Web site.

## Example ASCII Session

Here's an example session produced with Hyperterminal connected to the GP7. The GP7's bus connects to an LM76 temperature sensor:



? *Request communications byte*

2C *Response from GP7 indicates it is ready*

R9602 *Read 02 bytes from device 96*

0EA0 *Device responds (29.25C or 84.65F)*

XOK *Stop condition and GP7 acknowledgement*

# Notes

# Notes

# Notes