

## Manual for Large Display Unit

The Micro-VGA is the 640\*480 definition VGA display interface from Dontronics. There are two parts to the device – the VGA unit itself (Core) and the carrier board (Base board) into which it plugs. Both sections are needed. Total cost is about \$96.

In addition an additional circuit is needed between the shack unit and the VGA system. Although it would be possible to run the VGA system directly from the Shack Unit there is a speed problem. The shack unit has the job of calculating the position of the moon/sun and communicating with the motor and the antenna unit. In my early experiments it was clear that there are delays in the VGA system that would have added unwanted delays in control functions. For this reason it was decided to send the VGA data out as fast as possible from the Shack Unit without waiting for a response and doing the VGA communication elsewhere.

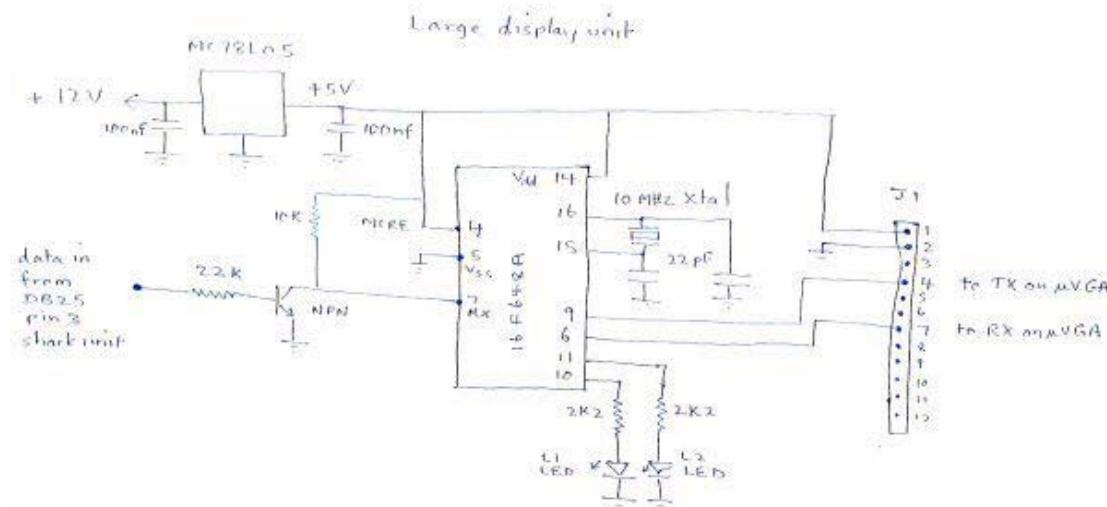
The VGA unit requires each character to be sent one at a time in a complex string. I use the following byte sequence to send each numeral/letter:

**Syntax : cmd, char, x(msb), x(lsb), y(msb), y(lsb), color, x\_size, y\_size**

The system must then wait for an <ack> from the u-VGA unit.

To write all the information to the VGA screen at once can take a significant part of a second. For this reason I only send the information that has changed. This is very effective in speeding things up but it does add to the complexity of the software. Updates are now very satisfactory.

The circuit for the interface:



Note the uVGA draws 175 mA so cannot be run from the MC78L05  
Replacement with a 7805 would enable uVGA to be run from this supply.

To implement the circuit I chose to modify an older version of the AZ/EL unit. By cutting a couple of traces I was able to solder an NPN transistor in the holes intended for the LEDs 3 and 4. It was necessary to cut a couple of traces to achieve this.

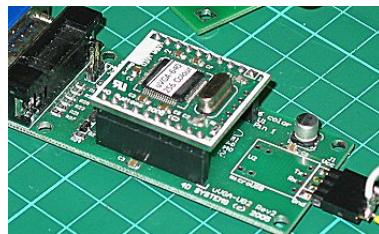
The hex code is burned into a 16F648A. The code is too long to fit into a 16F628. The pinouts for a 16F628 and 16F648 are identical.

### Memory map in the EEDATA of the PIC16F648A

Memory location	Value	Function
00 hex	00 hex	Foreground colour (00 = black)
01	C0 hex	Moon colour (\$C0 = blue)
02	07 hex	Sun colour (\$07 = red)
03	06 hex	Font multiplier (6 times base font size)
04	00 hex	Cursor X start position (* by 10 pixels)
05	00 hex	Cursor Y start position (* by 10 pixels)
06	4D hex	Start of "Moon" string
0B	53 hex	Start of "Sun" string
0F	20 hex	Start of blanks string
14	41 hex	Start of "Antenna" string
1C	20 hex	Start of " Park ON " string

The colour information can be changed at the time of burning the PIC. See the u-VGA manual for the colour information. Similarly the size multiplier could also be changed if you wish but I have set things for maximum character size on a 640\*480 screen.

### The Micro-VGA Unit



The u-VGA unit and its base board

The u-VGA draws nearly 200mA so it is not possible to access the +5V from the AZ/EL board as the 78L05 is limited to 100mA. One alternative is to replace the 78L05 with a 7805. Even so it would be a good idea to put a small heatsink on the 7805 if you use 12V input.

The u-VGA does take a little while to get itself sorted out. The boot sequence is something like this:

1. wait 1000ms for the VGA unit to boot
2. send a character \$55 at 19200 baud to enable the autosync baud rate feature.
3. wait for an 'Ack'.
4. If there is no <ack> after a suitable wait, resend (2) and wait for <ack>
5. send the background colour as read from the eedata (see memory map)
6. clear the screen
7. read the data coming in from the Shack Unit.
8. send all the data to the screen and setup the buttons (Wait for <ack> for each character)
9. save the data in RAM.

The VGA sequence then becomes:

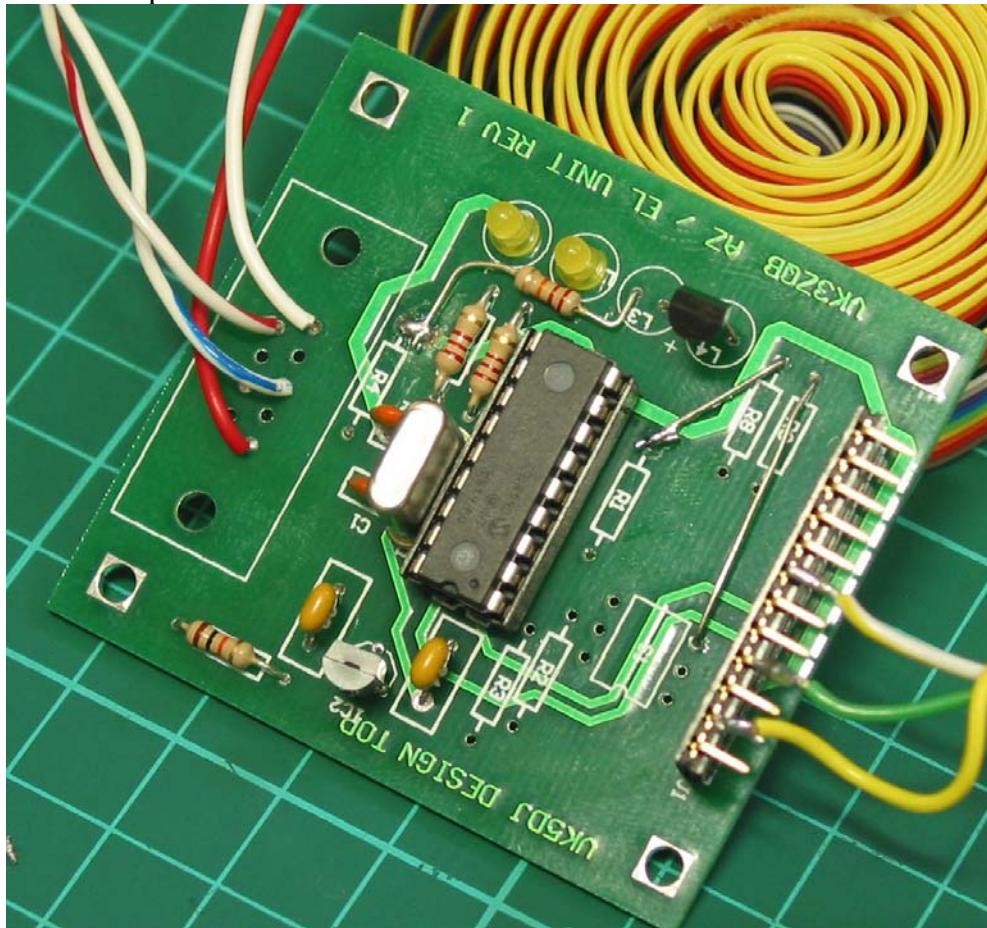
1. Bring data in from shack unit,

2. Compare it with the saved data from last time and
3. Send any data that is different.
4. Wait for <acks>
5. Update the saved data. and
6. Go to (1) of the VGA sequence

**Connections:**

DB25 pin 3 is the 9600 baud output of the Shack Unit and goes to the inverter transistor. DB25 pin 7 is the earth pin shared with the interface unit.

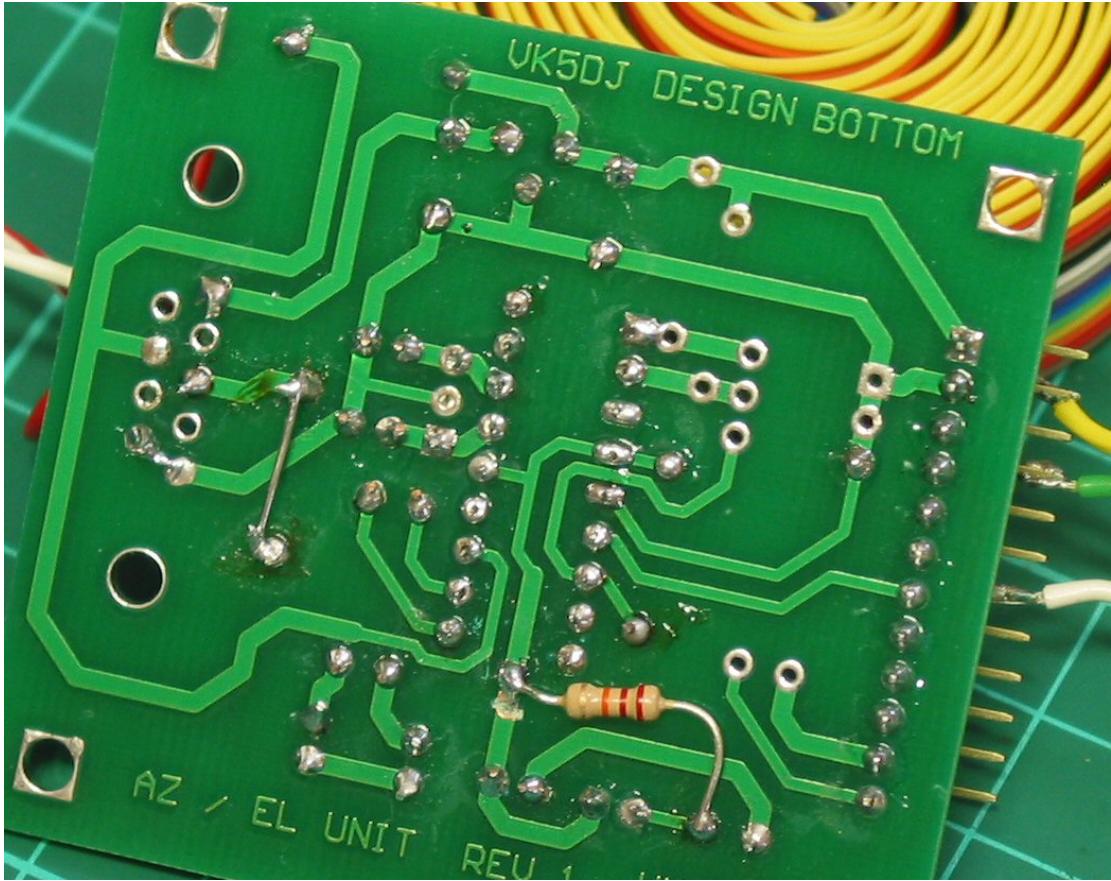
Photo of top of modified obsolete version of AZ/EL board



Notes to top of board:

- the trace at the LED end of R4 has been cut to isolate this point from pin 8 of the PIC
- a 22k resistor wired from this end of the R4 spot to one side of the L3 LED (see photo)
- an NPN transistor soldered into the holes for L3 and L4 (Emitter into L3 hole)
- jumper from holes for R8 to R1 (see photo)
- jumper from end of R9 to SW1 (see photo)
- the PCB overlay is wrong for the 78L05, the flat should face into the board (see photo)
- connections to the VGA board are pin 2 = earth, pin 4 = to TX pin on u-VGA, pin 7 = to RX on u-VGA

Photo of bottom of modified obsolete version of AZ/EL board



Changes to bottom of board –

- note trace cut to separate LED connections to +5V
- 22k resistor on the colour codes but use a 10K as per circuit added as shown
- Collector of transistor to the left of the resistor, the base just above the “1” of “Rev 1”, emitter above the “V” of “Rev 1”
- Jumper across R4 (due to an error when building it would have been neater to just take the 22k input resistor to the trace going to pin 3 of the DB9 and omitting this jumper)
- Jumper across pins 5,9 of the DB9 to produce an extra earth connection so I could have an earth power supply connection and an earth to the Shack Unit. This does not show clearly on the photo.
- Connections to the DB9 are: pins 5,9 earth, pin 3 data in from Shack Unit, pin 1 is +12V

#### How the serial comms work:

The 16F648A communicates with the Shack Unit via its UART running at 9600 baud. Because the computer port on the Shack Unit has a MAX232 fitted the data is inverted and at RS232 levels, therefore it is necessary to both invert the incoming data and protect the input port of the PIC against the negative transitions of the RS232 data with the NPN transistor.

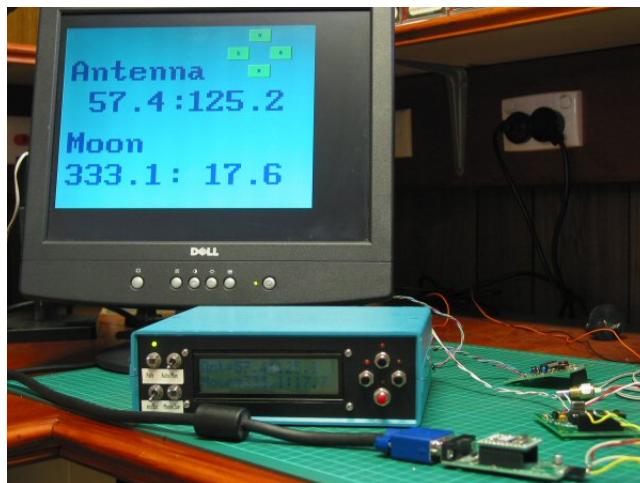
The 16F648A communicates with the u-VGA unit with bit bashed serial comms at 19200 baud. The command set for the u-VGA is available from the Dontronics site.

Neither of the two serial data streams use hardware handshaking although both wait for the correct start to each packet of information.

### **Important**

**To initiate communications with the VGA interface it is necessary to activate the VGA output on the Shack Unit's menu item number 30.**

**Version 5.62 or later of the Shack Unit's 18F4620 must be used as some changes were made to the information sent to the VGA adapter (antenna movement and Park Switch awareness added).**



The large display unit in action  
Note direction buttons top/right. These change colour when active.



The display in Park mode  
Buttons indicate left/down active

Enjoy the project

John Drew

VK5DJ