

The Propeller System Module (PSM)

User Manual

Raymond Allen, Ph.D., E.E

Copyright © 2009 www.rayslogic.com

Rev. 1a
11Jul09

Forward

Thank you for purchasing the Propeller System Module (PSM). I've enjoying working with the Propeller Microcontroller for the past few years and am delighted to bring you this Propeller based display and control module. I think you'll find the PSM to be a very versatile instrument, very good at playing roles from simple serial display to complete system on a board. Parallax has made the microcontroller easy to use with the Propeller, and the PSM makes it even easier in a fully integrated package. The PSM is ready to go with just the free Prop Tool software download from Parallax.com and the PSM drivers posted on Rayslogic.com. With just a computer and USB cable, you're ready to program the PSM and get you project off to a running start!

Table of Contents

User Manual.....	i
<i>Forward</i>	ii
<i>Table of Contents</i>	iii
<i>Table of Figures</i>	iv
<i>Main Features</i>	1
<i>Component Details</i>	3
The Parallax Propeller Microcontroller	3
QVGA TFT LCD	3
USB Interface and Power	4
PropPlug Compatible Serial Header	5
MEMS 3-Axis Accelerometer and ADC Input Header	5
I2C Interface Bus Header	6
I2C DAC Output Header and LCD Backlight Control	6
Header with 8 Free Propeller Pins	7
Amplified Stereo Output	7
Full-Size SD Card Mass Storage Interface	8
Front Panel Pushbutton	8
Small Prototyping Areas	8
<i>LCD Interface, Drivers and Software</i>	10
LCD Interface	10
LCD Drivers and Software	10
• PSM Graphics Demo	10
• PSM TV Text Demo.....	11
• PSM Serial Control.....	11
• PSM Slideshow.....	12

Table of Figures

Figure 1. Main Features of the Propeller System Module	1
Figure 2. Circuit diagram and photo of the USB and External Power interfaces.	4
Figure 3. Circuit diagram and photo of ADC and MEMS accelerometer.	5
Figure 4. Circuit diagram and photo of I2C DAC output and LCD backlight control.	6
Figure 5. Circuit diagram and photo of amplified stereo output.....	7
Figure 6. Circuit diagram and photo of SD card interface.....	7
Figure 7. Photo and circuit diagram of front panel pushbutton.	8
Figure 8. Photo showing locations of small prototyping areas.	8
Figure 9. Photo of the PSM running the “Graphics Demo”.....	11
Figure 10. Photo of the PSM running the “TV Text Demo”	11
Figure 11. Screenshot of the included Windows application for testing the PSM in serial control mode.	12
Figure 12. Photo of example slideshow photos.	12

Main Features

Microcontroller and Display:

The Propeller System Module (PSM) features the Parallax Propeller Microcontroller (MCU) and a 2.4" QVGA (320xRGBx240) TFT LCD display. The Propeller MCU has eight independent cores all running with 80-MHz clock speed. Each of the eight cores executes 20 million instructions per second (20 MIPS) for a total processing power of 160 MIPS. Generally, one core is utilized to drive the TFT display and another core is dedicated to the main program. This leaves six completely free cores, more than enough to utilize all the other devices on the PSM and for many more tasks.

Integrated Design:

As shown in Fig. 1, the PSM comprises a 3.8" by 2.5" circuit board with just the TFT display and pushbutton on one side and all other components on the other. The LCD is glued to the board and cannot be removed. There are four mounting holes for #4-40 hardware spaced 3"x2" apart for easy mounting.

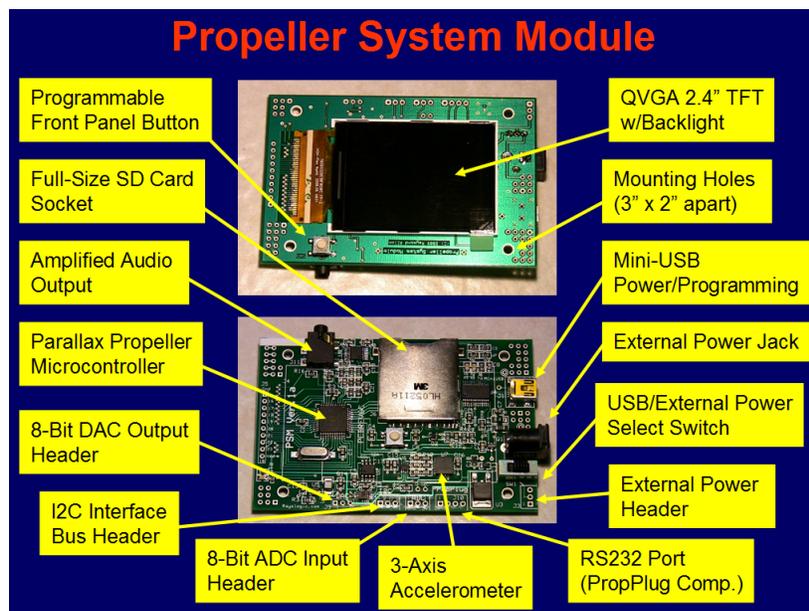


Figure 1. Main Features of the Propeller System Module

USB or External Power Supply:

The PSM can be powered by the mini-USB port or “external” power with the standard size barrel jack (5.5-mm OD/ 2.1-mm ID). A slide switch selects between USB and external power. This switch also serves as the On/Off switch (when only one source of power is connected). A 3-pin header is also available to provide input or output power.

USB or PropPlug Programming/Serial Interface:

The mini-USB port provides an easy way to program the Propeller MCU, in addition to providing power. All required Windows drivers are provided with the free “Prop Tool” programming environment from Parallax. A 32-kB EEPROM provides non-volatile storage for programs. A Parallax PropPlug compatible, 4-pin header is also provided as a second means of programming. In addition to programming, these interfaces can also be easily used for serial communications. For example, the PSM can be used as a serial controlled display using the provided driver. A Windows application is provided to demonstrate all the serial commands.

SD Card Mass Storage:

The PSM features a full-size SD (secure digital) card socket. This allows for mass storage of data, currently up to 2 GB. The FAT file system allows easy file transfer between PC and PSM.

3-Axis Accelerometer and ADC Input:

A 3-axis MEMS accelerometer provides an easy way to detect device orientation and as a means of input. A four channel ADC chip reads the accelerometer’s analog output and provided the data on the I2C bus. The spare ADC channel is provided on a 3-pin header.

DAC and LCD Backlight Control:

A two channel DAC chip is also provided on the I2C bus to control the LCD backlight intensity. The backlight is the single biggest power consumer on the PSM, so reducing the backlight when not in use is generally advisable for battery powered applications. One of the DAC channels controls the LCD backlight and the spare channel is provided on a 3-pin header.

Stereo Headphone Output:

A stereo audio amplifier is provided to directly drive headphones with the standard 3.5 mm jack. It can also drive the audio input of audio equipment.

Eight Free I/O Pins:

A 10-pin header provides power and 8 free Propeller pins. These pins connect directly to the Propeller with short traces. These can be used for a very wide variety of applications using available drivers, such as servo motor control, TV output, VGA output, PS2 keyboard, PS2 mouse, and much, much more...

I2C Bus Header:

For additional expansion, the I2C bus is also provided on a 3-pin header.

Small Prototyping Areas:

A few through holes are provided on the free space on the periphery of the board for adding components or connecting wires. Holes with a white circuit around them (on the labeled side) are connected to power and not for general use. All holes with square pads are grounded. Holes with a white line between them (only in top-left corner of bottom photo in Fig. 1) are connected together to make it easy to connect wires to the PSM.

Component Details

The Parallax Propeller Microcontroller

Parallax is known for making microcontrollers easy to use, especially with their Basic Stamp series of microcontrollers. The Propeller gives users a much more powerful microcontroller that is also very easy to use. Instead of managing interrupts, the multi-core RISC architecture of the Propeller allows one to dedicate a core to one or more specific tasks. This makes writing code for timing critical applications very simple.

The Propeller is a very robust microcontroller that can source or sink 60 mA on each pin and can directly drive LEDs. Although it operates at 3.3 V logic, it can be made 5 V tolerant by adding a series resistor in the 1 to 10 k Ω range.

Parallax provides a complete set free tools to program the Propeller, either its unique high-level language, SPIN, or in much faster assembly language. For most applications, SPIN is fast enough to be used for the main program. But, some drivers that require maximum speed, such as the PSM LCD driver, are written in assembly. In addition to the large set of device drivers provided, Parallax also hosts the Object Exchange (OBEX) where users have posted free (MIT Licensed) drivers for an even wider variety of devices.

It is recommended to utilize SPIN for most applications. However, for increased portability and speed, a C compiler is available from the Parallax website from Imagecraft. For very high-speed apps, you may need to program in assembly. Assembly programming for the Propeller is relatively simple, but very much different from 80X86 assembly due to the RISC architecture.

The Propeller User Forum, hosted by Parallax, is a much recommended, valuable resource for code and discussions about various aspects of the Propeller Microcontroller. In particular, compilers for different platforms and/or languages can be found there.

QVGA TFT LCD

The integrated 320xRGBx240 pixel LCD is what makes the PSM different from all other Propeller based boards. The convenience of having an integrated display cannot be underestimated. Similar board for other microcontrollers exist, but none can match the low cost (free programming software!) and ease of use of the Propeller.

The display is a Tianma Microelectronics Co., LTD model TM240320C1NFWGWC 2.4" TFT LCD display module. This is similar to displays typically found in PDAs and cell phones. The interface supports both 8 and 16-bit 8080-series modes, but the PSM uses the 8-bit mode exclusively in order to provide 8 completely free Propeller I/O pins.

Pixel color can be specified in either 16-bit or 24-bit modes. The 16-bit mode takes 2 transfers and the 24-bit mode takes 3 transfers per pixel. In general, when mimicking the

External Power Jack and Header

Power can also be provided using the standard power jack (5.5-mm OD, 2.1-mm ID), J8 also shown in Fig. 2. This jack connects directly to the 500-mA rated 3.3 V LDO voltage regulator with no fuse. So, you must be careful not to add any circuitry to the PSM that will increase the current draw beyond that level. The LDO regulator will work with applied voltages from 4 to 9 VDC. It may work for slightly lower voltages as long as the current draw is small. It may work up to 10 VDC as long as the regulator is adequately cooled. See the LP8345 LDO regulator datasheet for further details.

A 3-pin header, J3, is provided as another means of providing or extracting power. Pin 3 of J3 connects directly to the regulator. J3 can also be used to provide power to another circuit. Pin 2 of J3 is connected to 5 V USB power and could be used to power things like PS2 keyboard or mice.

PropPlug Compatible Serial Header

If for some reason you would rather program the PSM using a Parallax “PropPlug” USB to serial device, you can do so using J10 (see Fig. 3). The RX and TX lines could also be used for serial communications, besides programming. Additionally, those pins could be used as general purpose I/O when not programming. They could also provide a means for some other device to exchange data with the PC host over the USB port.

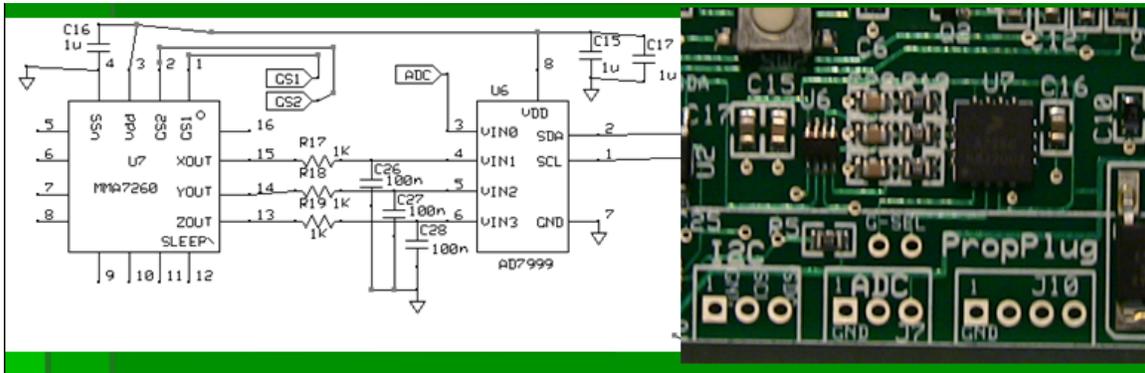


Figure 3. Circuit diagram and photo of ADC and MEMS accelerometer.

MEMS 3-Axis Accelerometer and ADC Input Header

The MMA7260 3-axis MEMS accelerometer has four different G ranges, but defaults to the most sensitive range of +/- 1.5 G. This range is useful for detecting the orientation of the device. If a larger range is required, two pins are provided for setting the range. They are under the “G-Sel” label on the board.

The accelerometer has an analog output, so an AD7999 4-channel ADC is used to digitize the data. RC filters are used on all 3 outputs as recommended in the MMA7260 datasheet. The ADC is connected to the I2C bus to transmit data to the Propeller MCU. Since the ADC is 8 bit the X and Y axis read around 128 when the board is flat. Note

that there is some device-to-device variation in this 0-G reading as specified in the datasheet.

Since the AD7999 has 4 channels and only 3 are required for the accelerometer, the spare input is provided on header J7. You may want to add a series resistor and shunt capacitor to this input for best results.

I2C Interface Bus Header

The PSM can be extended using the I2C bus header, J2, shown in the lower left corner in the photo of Fig. 3. There are several devices that can be added such as I/O expanders, RTC (real-time clocks), in addition to the I2C ADC and DAC used by the PSM.

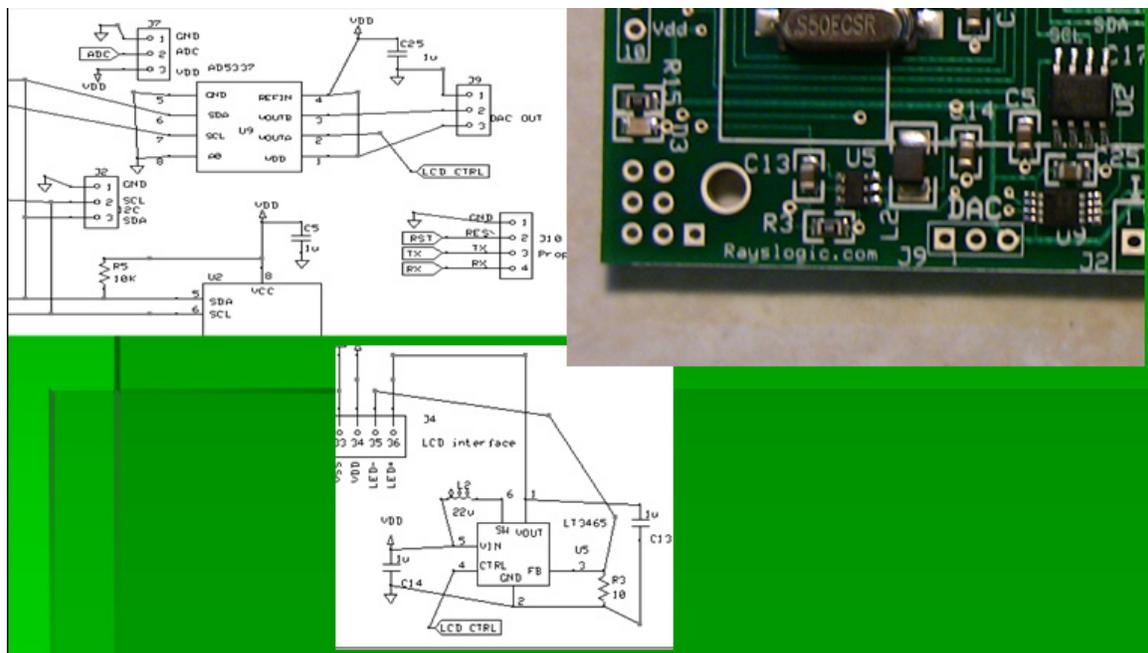


Figure 4. Circuit diagram and photo of I2C DAC output and LCD backlight control.

I2C DAC Output Header and LCD Backlight Control

A small IC chip controls the backlight intensity with an analog signal provided by one channel of a 2 channel DAC on the I2C bus. The extra DAC channel is provided on the header, J9, shown in Fig. 4. The DAC is 8-bit and converts an input level from 0-255 into a voltage between 0 and 3.3 V. The useful range of the backlight driver is less than that however. Anything above about 1.25 V (setting of ~100) translates into full intensity. To save power (especially in battery powered setups) it is a good idea to dim the backlight when not in use.

Header with 8 Free Propeller Pins

Having 8 free Propeller pins was a very important design consideration. These can be used for a wide variety of applications. With the PSM Accessory Board (currently in development, showcased at the Expo), these pins can be used to provide NTSC video, keyboard input, mouse input, and control a servo. The Propeller pin numbers (16..23) are labeled on the header, J5. Ground and Vdd are also provided on J5.

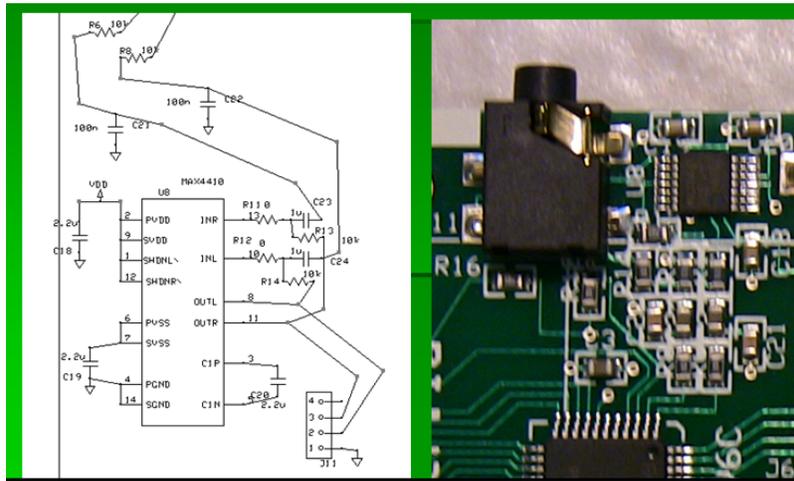


Figure 5. Circuit diagram and photo of amplified stereo output

Amplified Stereo Output

Two Propeller pins, P4 and P5, are used to generate stereo sound. This is fed to a MAX4410 headphone amplifier that can deliver fairly high quality sound to regular headphones via the standard 3.5 mm jack. The circuit diagram and photo of the jack, J11, are shown in Fig. 5.

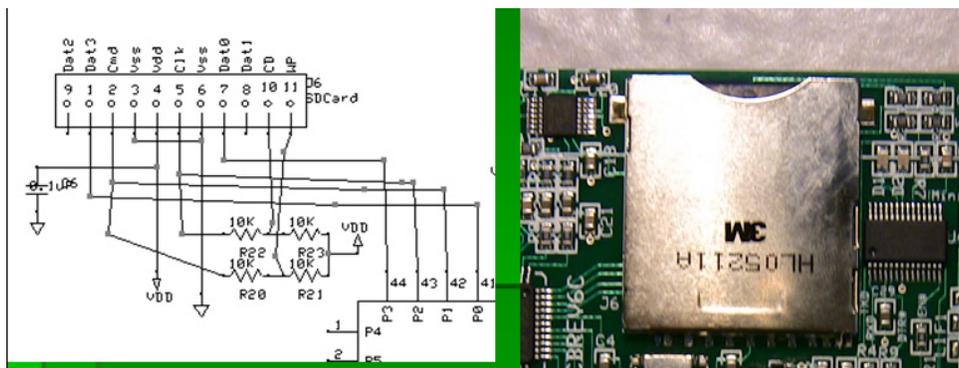


Figure 6. Circuit diagram and photo of SD card interface

Full-Size SD Card Mass Storage Interface

The PSM provides mass storage via the SD card interface using Pins 0..3, as shown in Fig. 6. The OBEX provided free (MIT licensed) code for the Propeller that can read and write files from a FAT16 formatted SD card (up to 2 GB). The driver does not currently support directories so all files to be used must be in the root directory.

With the serial control app loaded, you can transfer files back and forth from the PC using the example Windows App provided. The serial control application also shows how you can use the novel write-protect and card-detect circuits to detect if an SD card is inserted and to detect the position of the write-protect tab.

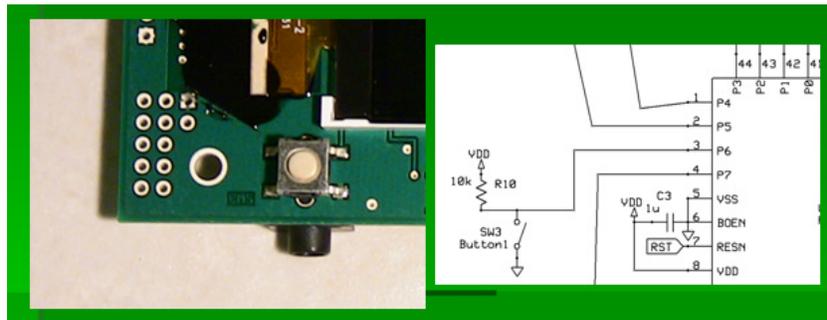


Figure 7. Photo and circuit diagram of front panel pushbutton.

Front Panel Pushbutton

The front panel (LCD side) of the PSM provides a pushbutton, SW3, for general purpose use as shown in Fig. 7. The front panel pushbutton is connected to Pin 6 of the Propeller. It is pulled up to be normally high and goes low when pushed.

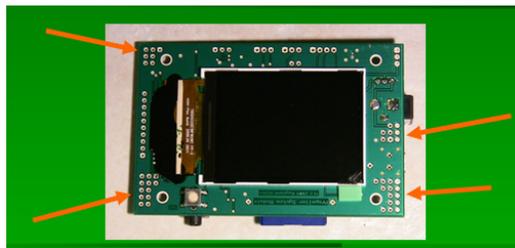


Figure 8. Photo showing locations of small prototyping areas.

Small Prototyping Areas

Any free space around the board was turned into a prototyping area as indicated in Fig. 8. All holes with square pads are ground. Holes with a white circle on the stenciled side of the board are special. One white circle hole is 3.3 V from the FTDI chip, the other is directly connected to the 5.1 mm barrel power supply connector. Holes in the lower left

corner (by the pushbutton) with a white line between them on the stenciled side are connected together.

LCD Interface, Drivers and Software

LCD Interface

The main feature of the PSM is the integrated full color display. The display is 320x240 pixels and has 18-bits of internal memory per pixel. The pixel color information can be sent to the display either in 16 or 24 bits. Pixel data can also be read from the display (rather slowly) as 16 bits. The LCD is interfaced in 8-bit 8086 mode so as to minimize pin usage with only a small sacrifice in speed.

Windows .bmp images can be displayed using both the serial control and slideshow apps provided. The drivers are currently optimized for the normal, 24 bit-per-pixel, image format, but most BMP formats can be displayed.

LCD Drivers and Software

The PSM LCD drivers can be grouped into buffered or unbuffered modes. In buffered mode, a representation of the entire screen is stored in HUB RAM. In this mode, pixel color information is sent to the display in 16-bit format. Examples of buffered mode are the “PSM Graphics Demo” which faithfully reproduces the well known Propeller “Graphics Demo”, and “PSM TV Text Demo”, which works almost identically as the regular “TV Text Demo”. In unbuffered mode, routines write directly to LCD. In this mode the pixel data is sent as 24 bits for best photo rendering. Examples of unbuffered mode are the “PSM Serial Control”, which turns the display into a serial controlled display, and “PSM Slideshow”, which shows a series of Windows BMP pictures on the LCD.

- PSM Graphics Demo

The PSM can run the amazing “Graphics Demo”, as shown in Fig. 9, with just a few, slight modifications. The current driver can run the Graphics Demo in either the original 256x192 resolution, or can be increased to 320x240. Because the LCD contains it’s own GRAM buffer, double-buffering in HUB RAM is not required. This is a huge savings in memory and can allow for much larger programs even with the increased resolution.

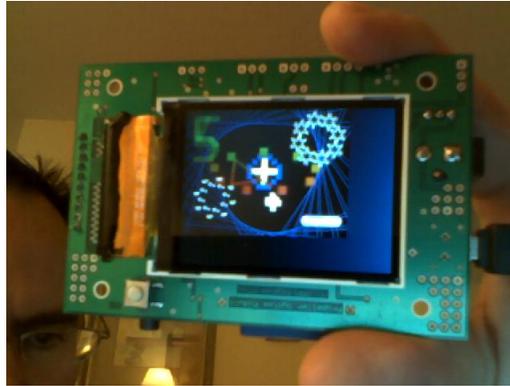


Figure 9. Photo of the PSM running the “Graphics Demo”

- PSM TV Text Demo

The PSM can replicate the “TV Text Demo” as shown in Fig. 10. It is possible to have concurrent display on both the LCD and NTSC TV monitor using 3 of the free pins and a few resistors. This driver continuously updates the screen, making it the simplest driver to use. The number of rows can be increased from 13 to 20, while the number of columns is the same as the TV version, 40.



Figure 10. Photo of the PSM running the “TV Text Demo”

- PSM Serial Control

With the included Windows application, shown in Fig. 11, and the serial control driver, you can try out many functions of the PSM using the serial connection provided by a USB cable.

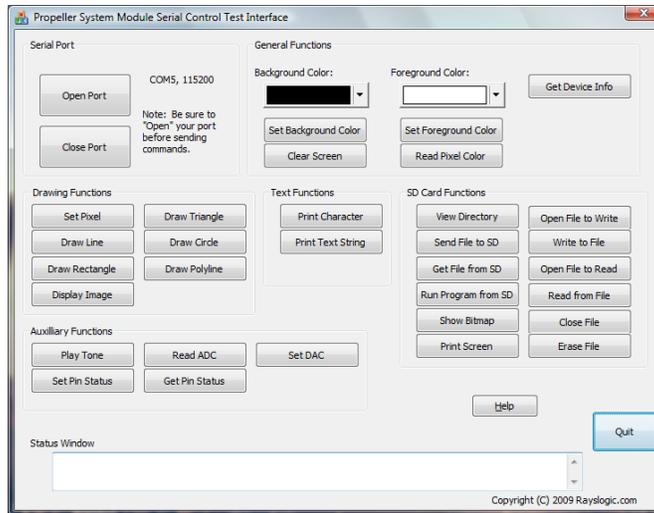


Figure 11. Screenshot of the included Windows application for testing the PSM in serial control mode.

- PSM Slideshow
 With this application you can display Windows BMP files that are loaded on the SD card. Just change to code to reflect your filenames. The driver comes configured for the two example photos shown in Fig. 12.

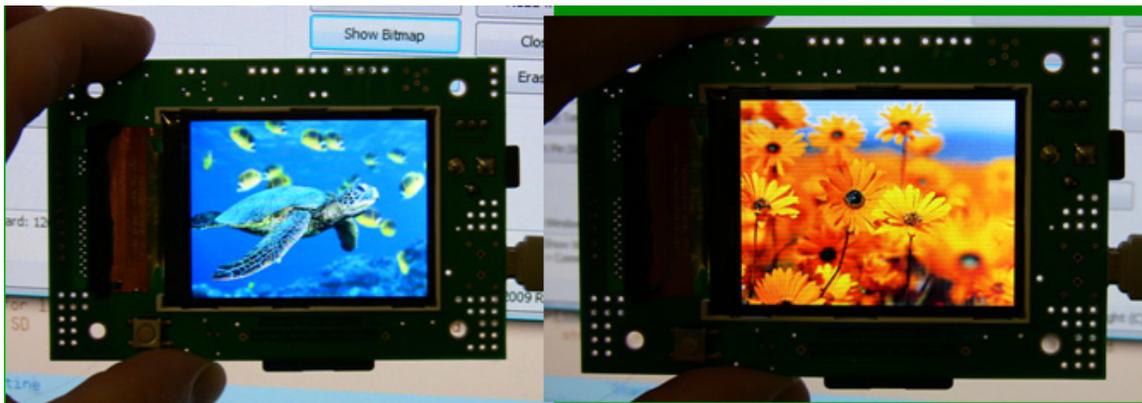


Figure 12. Photo of example slideshow photos.