

E Ink Display Controller BoosterPack

DLP-ECON-BP BOOSTERPACK FEATURES:

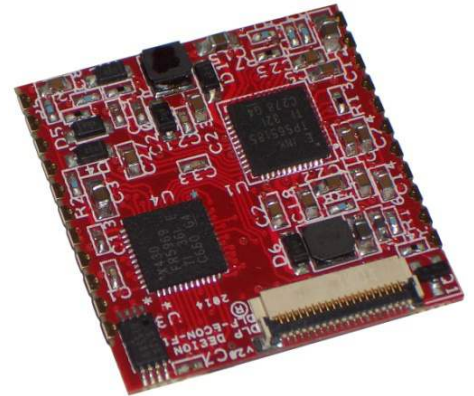
- Compatible with Texas Instruments' G2 LaunchPad (Required)
- Temperature/Humidity Sensor
- DLP-ECON-F1 Control Module Incorporated
- E Ink 1.73" Mobius Display Included
- Selectable Host: GUI/USB or LaunchPad
- Host PC GUI for Image/Icon Creation
- Four User-Definable Buttons
- All Power Taken from the TI LaunchPad



Shown here with the TI G2 LaunchPad
(Sold Separately)

DLP-ECON-F1 E INK CONTROLLER FEATURES:

- Compatible with the E Ink 1.73" Mobius Display
- Simple UART Interface at 115.2K Baud
- Single 3.3-3.6V Supply
- On-Board, 5-Rail PMIC
- Programmable VCOM (Non-Volatile)
- Surface Mount Module Format
- 1.2" x 1.2" (30.5 x 30.5mm) Footprint
- Built-in FPC Display Connector
- On-Board, Non-Volatile Storage for Three Full Display Images
- Up to 90 32x32-Pixel, User-Defined Icons
- Windows Demonstration GUI with Barcode Creation Support Provided



1.0 INTRODUCTION

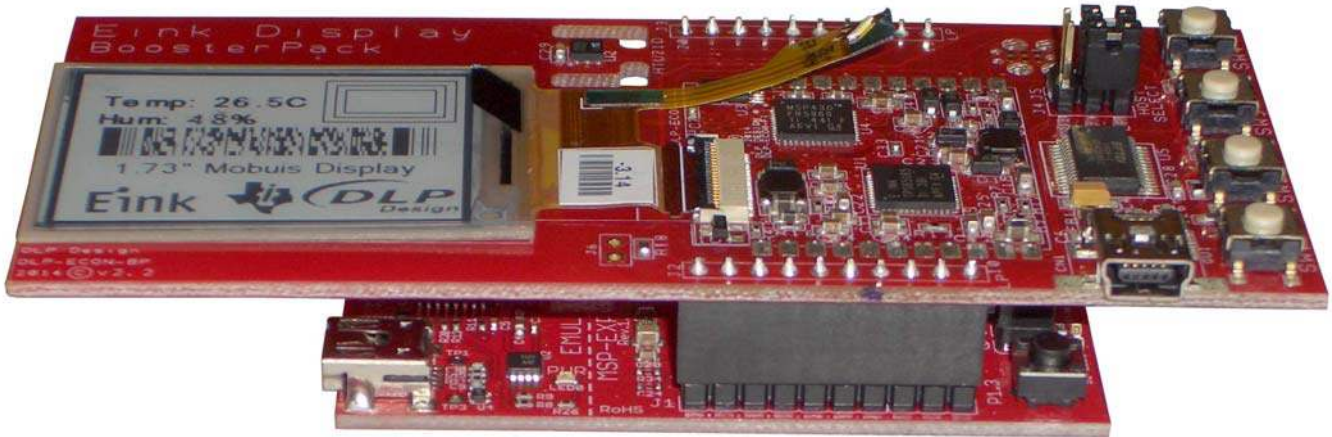
The DLP-ECON-F1 E Ink Display Controller is designed to make easy work of driving E Ink's 1.73-inch Mobius display. This module consists of the electronic components necessary to correctly update the image on the e-paper display via a simple TTL serial interface typically found in low-cost microcontrollers. The DLP-ECON-BP BoosterPack can be controlled by two types of host: PC GUI via USB (Virtual COM Port), or the Texas Instruments LaunchPad.

In a typical usage scenario when the host is a PC, the GUI is utilized to create and send over full-screen displays including barcodes and icons via the USB interface. The GUI can also import *.bmp bitmaps which can be used in full-screen displays. When the host is a TI LaunchPad (or small microcontroller), non-full-screen control features are used to update the contents of the display: Clear Display, Display Text, Icon On/Off and Global Update via non-volatile memory.

2.0 QUICK-START GUIDE

The DLP-ECON-BP BoosterPack can be controlled via two types of hosts: USB connection to a PC or by a TI LaunchPad. Using either method, power to operate this BoosterPack is sourced via the USB interface on the TI LaunchPad; therefore, a TI LaunchPad is required for normal operation.

WARNING - Do not mount the DLP-ECON-BP BoosterPack to the LaunchPad backwards. Doing so may cause permanent damage to the BoosterPack. Only mount the BoosterPack to the LaunchPad as shown here:



2.1 USING THE LAUNCHPAD AS HOST CONTROLLER

When the TI LaunchPad is used as the host controller, the typical application scenario is one whereby the intent is to only send small control packets for local updates to the screen such as placement of text, loading full-screen images to the display from non-volatile memory, icon display/clear, etc. More to the point, full-screen loading of images to the frame buffer is typically avoided because small micro-controllers serving as a host typically do not have the storage capability to hold full-screen (20K byte) images. That said, even the smallest microcontroller can easily send over full-screen images if it can send serial data at 115,200 baud.

2.2 USING THE GUI AS HOST CONTROLLER VIA THE USB INTERFACE

When a PC is used as the host, the DLP-ECON-F1 development GUI (or customer-provided host software) is utilized to create and send images.

2.3 QUICK-START GUIDE

2.3.1 Connect the DLP-ECON-BP to a TI G2 LaunchPad paying careful attention to the orientation between the two boards.

WARNING - Do not mount the DLP-ECON-BP BoosterPack to the LaunchPad backwards as this may cause permanent damage to the BoosterPack. Only mount the BoosterPack to the LaunchPad as shown in Section 2.0 (above).

2.3.2 Select the host by setting Jumpers J4 and J5 on the BoosterPack to either "LP" for LaunchPad or "GUI" for PC as host via USB.

2.3.3 Apply power to the board set by connecting the LaunchPad to a host PC using the USB cable provided with the LaunchPad.

2.3.4 If using a PC as the host, connect the BoosterPack to the host PC using the provided USB cable and the USB connector on the BoosterPack PCB.

2.3.5 Verify operation of the system by clearing the display. If a LaunchPad is the host, press SW1 on the BoosterPack. If a PC is the host, verify that the USB drivers are correctly loaded by looking in Device Manager and verifying that there are no error messages. If a COM port is present for the BoosterPack, then this must be disabled in order for the demonstration GUI to work correctly. Disable the VCP (Virtual COM Port) drivers via Properties for the USB Serial Converter listed under USB Controllers within Device Manager. After disabling the VCP drivers, you must unplug and replug the USB cable. Finally, verify correct operation of the system by clicking on the Clear Display button in the GUI.

At this point the DLP-ECON-BP BoosterPack is ready for use.

3.0 DLP-ECON-BP BOOSTERPACK

DLP-ECON-BP BoosterPack contains an E Ink display, display controller, temperature and humidity sensor, four push-button switches and connections for both a TI LaunchPad and USB connection to a host PC.

The DLP-ECON-BP BoosterPack can be controlled via two methods: USB host to a Windows PC or by a TI LaunchPad. Using either method, power to operate the BoosterPack is sourced via the USB interface on the TI LaunchPad; therefore, a G2 LaunchPad is required in order to use this product.

3.1 BOOSTERPACK SPECIFICATIONS

3.1.1 POWER SUPPLY

Power for the BoosterPack is only supplied via the LaunchPad. The USB cable connection to the host PC supplies all required power as well as providing the program and debug interface between the Code Composer Studio compiler and G2 microcontroller on the TI LaunchPad.

3.1.2 LAUNCHPAD PINOUT

DLP-ECON-BP BOOSTERPACK PIN CONNECTIONS TO THE G2 LAUNCHPAD	
1	VCC - 3.6V power from the LaunchPad
2	P1.0 - Not used
3	UART RXD - Data sent from DLP-ECON-F1 to the host LaunchPad
4	UART TXD - Data sent from the LaunchPad host to the DLP-ECON-F1

5	P1.3 - Not used
6	P1.4 - Not used
7	P1.5 - Not used
8	P2.0 - SW1 Switch input to G2 microcontroller on the LaunchPad
9	P2.1 - SW2 Switch input to G2 microcontroller on the LaunchPad
10	P2.2 - SW3 Switch input to G2 microcontroller on the LaunchPad
11	P2.3 - SW4 Switch input to G2 microcontroller on the LaunchPad
12	P2.4 - I/O used to communicate with the humidity sensor (data)
13	P2.5 - I/O used to communicate with the humidity sensor (clock)
14	P1.6 - Not used
15	P1.7 - Not used
16	RESET - Not used
17	TEST - Not used
18	XOUT - Not used
19	XIN - Not used
20	GROUND

3.1.3 COMMUNICATIONS PARAMETERS

TTL serial communications parameters are as follows:

Baud: 115,200
Parity: None
Data Bits: 8
Stop Bits: 1
Voltage: 0-3V; Idle When High

3.1.4 CURRENT-SENSING RESISTOR

A current-sensing resistor is installed on the BoosterPack in series with the power being supplied to the DLP-ECON-F1 controller module. The resistor (R18) is a 0.1-Ohm, 1%-tolerance resistor. To monitor the current being drawn by the module, a millivoltmeter can be used to measure the drop across this resistor, and the current draw can be calculated. Jumper J6 can be used to either short out resistor R18 or provide an easier connection to a meter.

3.2 HOST-SELECTION JUMPERS

Select the host by setting Jumpers J4 and J5 on the BoosterPack to either "LP" for LaunchPad or "GUI" for PC as host via USB.

3.3 HUMIDITY SENSOR

The humidity sensor provided on the BoosterPack is the HTU21D from Servoflow Corporation (www.servoflo.com). The data from this sensor is read via a two-wire (clock/data) digital interface by the G2 microcontroller on the LaunchPad.

This sensor will measure temperatures from -40 to 125°C, and humidity from 0 to 100%. Refer to the manufacturer's datasheet for more details:

<http://www.servoflo.com/environmental-sensors/humidity-sensors/digital-humidity-sensors/1025-htu21d.html>.)

The G2 source code for reading data from this sensor is included in the source code for the BoosterPack and is provided as a free download upon purchase.

3.4 PUSH-BUTTON SWITCHES

There are four push buttons on the BoosterPack that can be used when the LaunchPad is the host controller (J4 and J5 set to "LP"). These buttons are read by the G2 microcontroller on the LaunchPad. When the G2 detects a button press, it responds by sending a small control packet to the DLP-ECON-F1 controller to execute one of the four features listed below: (Demonstration firmware running in the G2 microcontroller can be replaced by the user's custom code.)

3.4.1 BUTTON 1 Clear and refresh the display. This function will cycle the entire screen from black to white then black to white again. Doing so serves to both clear the screen and rejuvenate any locations in the display that may have been overdriven by other hardware.

3.4.2 BUTTON 2 This button starts a continuous loop in which the G2 reads temperature and humidity data from the sensor on the BoosterPack, converts the data into two short strings and sends the strings via command packet to the DLP-ECON-F1 controller for local update. The characters within the strings are converted to their binary equivalents in the DLP-ECON-F1 and loaded to the display at the location specified by the packet sent. (With the local update, only the pixels that must change for the required presentation of data are changed. The rest of the display remains unchanged.)

NOTE - While in this "Run" mode, the other buttons can be pressed, and their functions will occur as expected, but you may have to hold the button a little longer for it to be detected by the G2.

If humidity at the sensor rises above 50%, a "rain" indicator icon in the upper left-hand corner will turn on and stay on until the level drops below 50%. This icon resides in the DLP-ECON-F1 controller's non-volatile memory. This memory will hold up to 90 user icons.

3.4.3 BUTTON 3 When pressed, the next of three stored full-screen images is transferred from non-volatile memory in the DLP-ECON-F1 to the display. This process will completely change the contents of the screen; so if the loop is running (see Button 2), then the temperature and humidity data will reappear shortly thereafter. (These three saved full-screen images can be easily replaced by images created by the user.)

3.4.4 BUTTON 4 This button toggles the E Ink display's backlight on and off.

4.0 DLP-ECON-F1 CONTROLLER

The DLP-ECON-F1 controller generates all driving voltages required by the E Ink Mobius 1.73" display from its single 3.3-3.6 volt power supply. It can store both full-screen images and 90 32x32-bit custom icons that can be displayed and cleared from 20 preset locations along the top and bottom rows of the E Ink display. The production version of this module is designed to be surface mounted to customer PCB's and is RoHS, RoHS2 and REACH compliant.

4.1 PINOUT

DLP-ECON-F1 PIN CONNECTIONS	
1	PJ.2 - GPIO
2	PJ.3 - GPIO
3	P4.0 - GPIO
4	GROUND
5	BKLT - Backlight Input - Connected to Pin 6 (P4.1) on the DLP-ECON-F1
6	P4.1 - GPIO
7	3V Power In: 3.3-3.6V power source
8	US_SEL - Not used
9	GROUND
10	FRMHST - Data sent from the host to the DLP-ECON-F1
11	TOHST - Data sent from the DLP-ECON-F1 to the host
12	SPWTCH - Not used
13	RESET - System reset
14	P2.3 - GPIO
15	P2.4 - GPIO
16	PJ.6 - GPIO
17	PJ.7 - GPIO
18	GROUND
19	PJ.4 - GPIO
20	PJ.5 - GPIO
21	P4.7 - GPIO
22	P1.3 - GPIO

4.2 BACKLIGHT

Pin 5 on the DLP-ECON-F1 controller is used to control the backlight on the E Ink display. Taking this pin high will activate the LED backlight on the E Ink display. This pin is connected to one of the GPIO lines on the DLP-ECON-F1 (P4.1) so that the backlight can be controlled via host software.

Pin 5 must be tied high, low, to a GPIO or to another controlling circuit--it must not be left floating.

4.3 COMMAND SET

The TTL serial interface to the DLP-ECON-F1 controller operates at 115,200 baud. The host can be either a PC via USB or the TI G2 LaunchPad. Commands are issued to the controller in packets as described in this section. Every packet sent has the following components:

SOF (Start of Frame) - 0xF5: first byte of every packet

SZ - Packet size from SOF up through the CHKSM, not including CHKSM (single byte)

CMD - Command being issued (single byte)

DATA - Payload, if any (zero or more bytes). If there are zero data bytes, substitute the CHKSM byte instead.

CHKSM - Checksum; 0x55 XOR'd with all bytes in the packet except the CHKSM (single byte)

This packet structure (SOF, SZ, CMD, DATA, CHKSM) must be used to control the DLP-ECON-F1 module. For example, to issue the Clear Screen command, the following bytes must be sent: 0xF5, 0x04, 0x40, 0xB1. (In this example, no data is required for the command byte, so 0x40 is the last byte issued before the checksum byte.)

Every command packet received by the controller results in a reply being sent back to the host indicating which command was received (command sent +1). This assumes that the incoming checksum byte was correct. If the reply contains 0x99, an error has occurred.

If an incorrect packet is received by the controller, then either no byte or an error byte (0x99) is returned to the host by the DLP-ECON-F1 controller. If an invalid packet is issued and no byte is returned to the host; then the packet processor in the DLP-ECON-F1 is in an unknown state, and it is recommended that the circular buffer be cleared by writing 200 zeros to the controller or by cycling power.

Every byte received by the controller is stored in a 200-byte circular buffer. Care must be taken to never send more than 200 bytes without waiting for the controller to reply indicating the presence of additional capacity in the buffer.

COMMANDS:

0x30 → Request image from the target frame buffer

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x30	CMD	Command
0x94	CHKSM	XOR of preceding bytes, seeded with 0x55

(No payload, just CHKSM)

0x31 ← Reply to command 0x30

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x31	CMD	Reply (0x31)
0x95	CHKSM	XOR of preceding bytes, seeded with 0x55
0-x4AFF		19,200 bytes of image data

(No payload, just CHKSM)

0x40 → Clear/refresh the E Ink display

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x40	CMD	Command
0xE4	CHKSM	XOR of preceding bytes, seeded with 0x55

(No payload, just CHKSM)

0x41 ← Reply back from target for command 0x40

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x41	CMD	Reply (0x41)
0xE5	CHKSM	XOR of preceding bytes, seeded with 0x55

(No payload, just CHKSM)

0x42 - Perform Global Display Update – (writes an inverse image first to refresh the display)

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x42	CMD	Command
0xE6	CHKSM	XOR of preceding bytes, seeded with 0x55

(No payload, just CHKSM)

0x43 - Reply to command 0x42

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x43	CMD	Reply (0x43)
0xE7	CHKSM	XOR of preceding bytes, seeded with 0x55

(No payload, just CHKSM)

0x44 - Perform Local Display Update - only writes pixels that change

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x44	CMD	Command
0xE0	CHKSM	XOR of preceding bytes, seeded with 0x55

(No payload, just CHKSM)

0x45 - Reply to command 0x44

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x45	CMD	Reply (0x45)
0xE1	CHKSM	XOR of preceding bytes, seeded with 0x55

(No payload, just CHKSM)

0x49 -> Turn Backlight ON

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x49	CMD	Command
0xED	CHKSM	XOR of preceding bytes, seeded with 0x55

(No payload, just CHKSM)

0x4A <- Reply to Turn Backlight ON

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x4A	CMD	Reply (0x4A)
0xEE	CHKSM	XOR of preceding bytes, seeded with 0x55

(No payload, just CHKSM)

0x4B -> Turn Backlight OFF

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x4B	CMD	Command
0xEF	CHKSM	XOR of preceding bytes, seeded with 0x55

(No payload, just CHKSM)

0x4C <- Reply to Turn Backlight OFF

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x4C	CMD	Reply (0x4C)
0xE8	CHKSM	XOR of preceding bytes, seeded with 0x55

(No payload, just CHKSM)

0x50 -> Ping controller

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x50	CMD	Command
0xF4	CHKSM	XOR of preceding bytes, seeded with 0x55

(No payload, just CHKSM)

0x51 <- Reply to ping from target

0xF5	SOF	Start of frame
0x09	SZ	Packet size (0x09)
0x50	CMD	Reply (0x51)
5 Bytes	ASCII data	for FW version (MM.mm)
chksm	CHKSM	XOR of preceding bytes, seeded with 0x55

0x52 -> Send full display image from host to frame buffer in controller

0xF5	SOF	Start of frame
0x05	SZ	Packet size (0x05)
0x52	CMD	Reply (0x51)
CRC	CRC	of the image data
chksm	CHKSM	XOR of preceding bytes, seeded with 0x55
image	19,200 bytes of image data;	200 bytes at a time

0x53 <- Reply to command 0x52 indicating display image received and CHKSM is correct

0x53	Reply (0x53)
------	--------------

Or if error: 0x99 <- Error

0x99	Reply (0x99)
------	--------------

0x55 -> Request a CRC of the frame buffer

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x55	CMD	Command
0xF1	CHKSM	XOR of preceding bytes, seeded with 0x55

(No payload, just CHKSM)

0x56 <- Reply with CRC value

0xF5	SOF	Start of frame
0x05	SZ	Packet size (0x05)
0x56	CMD	Reply (0x56)

crc	CRC of the image buffer data
chksm	CHKSM XOR of preceding bytes, seeded with 0x55

0x57 -> Request PMIC register values

0xF5	SOF	Start of frame
0x05	SZ	Packet size (0x05)
0x57	CMD	Reply (0x57)
reg adr	PAYLOAD	PMIC register address
chksm	CHKSM XOR of preceding bytes, seeded with 0x55	

0x58 - Reply to 0x57 command with values read

0xF5	SOF	Start of frame
0x05	SZ	Packet size (0x05)
0x58	CMD	Reply (0x58)
data	PAYLOAD	PMIC register data
chksm	CHKSM XOR of preceding bytes, seeded with 0x55	

0x59 - Write single PMIC register @ address

0xF5	SOF	Start of frame
0x06	SZ	Packet size (0x06)
0x57	CMD	Reply (0x57)
reg adr	PAYLOAD	PMIC register address
reg data	PAYLOAD	PMIC register data
chksm	CHKSM XOR of preceding bytes, seeded with 0x55	

0x5A - Reply to 0x59 command with pass/fail indication

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x5A	CMD	Command
0xFE	CHKSM XOR of preceding bytes, seeded with 0x55	

(No payload, just CHKSM)

0x5B - Write VCOM voltage to PMIC

0xF5	SOF	Start of frame
0x06	SZ	Packet size (0x06)
0x5B	CMD	Command
reg adr	PAYLOAD	VCOM voltage MSByte
reg data	PAYLOAD	VCOM voltage LSByte
chksm	CHKSM XOR of preceding bytes, seeded with 0x55	

0x5C - Reply to 0x5B command with pass/fail indication

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x5C	CMD	Command
0xF8	CHKSM	XOR of preceding bytes, seeded with 0x55

(No payload, just CHKSM)

0x5C -> Read VCOM Registers

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x5C	CMD	Reply (0x58)
0xF8	CHKSM	XOR of preceding bytes, seeded with 0x55

0x5D <- Reply to Read VCOM Registers

0xF5	SOF	Start of frame
0x15	SZ	Packet size (0x15) (21 bytes)
0x5D	CMD	Reply (0x5D)
17 Bytes	PAYLOAD	PMIC register data (17 bytes)
chksm	CHKSM	XOR of preceding bytes, seeded with 0x55

0x60 -> Import font character from host and write to EEPROM

0xF5	SOF	Start of frame
0x85	SZ	Packet size (0x85) (133 bytes)
0x60	CMD	Command
data	PAYLOAD	96b of font character data + 32b nulls
chksm	CHKSM	XOR of preceding bytes, seeded with 0x55

0x61 <- Reply to 0x60 command

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x61	CMD	Reply (0x61)
0xC5	CHKSM	XOR of preceding bytes, seeded with 0x55

(No payload, just CHKSM)

0x62 -> Command to draw text on screen

0xF5	SOF	Start of frame
0xnn	SZ	Packet size (string length + 8)
0x62	CMD	Command
XMSB	X Most Significant Byte	
XLSB	X Least Significant Byte	
YMSB	Y Most Significant Byte	
YLSB	Y Least Significant Byte	
data	PAYLOAD	ASCII string data (not null terminated)
chksm	CHKSM	XOR of preceding bytes, seeded with 0x55

0x63 <- Reply to command 0x62

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x63	CMD	Reply (0x63)
0xC7	CHKSM	XOR of preceding bytes, seeded with 0x55

(No payload, just CHKSM)

0x63 -> Display or hide an icon on the display.

To use the preset icon-position addresses, use 0-9 along the top and 10-19 along the bottom of the E Ink display, set XMSB to FF and 0-19 for the XLSB byte.

0xF5	SOF	Start of frame
0x0A	SZ	Packet size (0x0A)
0x63	CMD	Command
0-0x59	Icon index	
0 or 1	Icon OFF/ON	(0 for Hide Icon; 1 for Display Icon)
XMSB	X Most Significant Display Location Byte	(use FF for the preset icon positions)
XLSB	X Least Significant Byte	
YMSB	Y Most Significant Display Location Byte	(use FF for the preset icon positions)
YLSB	Y Least Significant Byte	
chksm	CHKSM	XOR of preceding bytes, seeded with 0x55

0x64 <- Reply to command 0x63

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x64	CMD	Reply (0x64)
0xC0	CHKSM	XOR of preceding bytes, seeded with 0x55

(No payload, just CHKSM)

0x64 -> Import icon from host

0xF5	SOF	Start of frame
0x90	SZ	Packet size (0x90) (144b)
0x64	CMD	Command
0x0-0x59	PAYLOAD	Index for icon storage locations for icons 0-89 decimal.
0x0	PAYLOAD	Segment 0
138 Bytes	PAYLOAD	138 bytes of icon data (128 + 10)
chksm	CHKSM	XOR of preceding bytes, seeded with 0x55

0x65 <- Reply to command 0x64

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x65	CMD	Reply (0x65)
0xC1	CHKSM	XOR of preceding bytes, seeded with 0x55

(No payload, just CHKSM)

0x65 -> Export icon to host

0xF5	SOF	Start of frame
0x06	SZ	Packet size (0x06)
0x65	CMD	Command
0x0-0x59	PAYLOAD	Index for icon storage locations for icons 0-89 decimal.
0	PAYLOAD	Segment 0
chksm	CHKSM	XOR of preceding bytes, seeded with 0x55

0x66 <- Reply to command 0x65

0xF5	SOF	Start of frame
0x8E	SZ	Packet size (0x8E) (142 bytes)
0x66	CMD	Command
data	PAYLOAD	96 bytes of font character data + 32 null bytes
chksm	CHKSM	XOR of preceding bytes, seeded with 0x55

0x90 -> Copy frame buffer to controller non-volatile storage 1

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x90	CMD	Command
0x34	CHKSM	XOR of preceding bytes, seeded with 0x55

0x91 - Reply to 0x90 command for process complete

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x91	CMD	Reply (0x91)
0x35	CHKSM	XOR of preceding bytes, seeded with 0x55

(No payload, just CHKSM)

0x92 -> Copy frame buffer to controller non-volatile storage 2

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x92	CMD	Command
0x36	CHKSM	XOR of preceding bytes, seeded with 0x55

0x93 - Reply to 0x92 command for process complete

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x93	CMD	Reply (0x93)
0x37	CHKSM	XOR of preceding bytes, seeded with 0x55

(No payload, just CHKSM)

0x94 -> Copy frame buffer to controller non-volatile storage 3

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x94	CMD	Command
0x30	CHKSM	XOR of preceding bytes, seeded with 0x55

0x95 - Reply to 0x94 command for process complete

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x95	CMD	Reply (0x95)
0xC1	CHKSM	XOR of preceding bytes, seeded with 0x55

(No payload, just CHKSM)

0xA0 - Copy non-volatile storage page 1 to frame buffer

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x94	CMD	Command
0x04	CHKSM	XOR of preceding bytes, seeded with 0x55

0xA1 - Reply to 0xA0 command for process complete

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x95	CMD	Reply (0x95)
0x06	CHKSM	XOR of preceding bytes, seeded with 0x55

(No payload, just CHKSM)

0xA2 - Copy non-volatile storage page 2 to frame buffer

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x94	CMD	Command
0x30	CHKSM	XOR of preceding bytes, seeded with 0x55

0xA3 - Reply to 0xA2 command for process complete

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x95	CMD	Reply (0x95)
0x07	CHKSM	XOR of preceding bytes, seeded with 0x55

(No payload, just CHKSM)

0xA4 - Copy non-volatile storage page 3 to frame buffer

0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x94	CMD	Command
0x00	CHKSM	XOR of preceding bytes, seeded with 0x55

0xA5 - Reply to 0xA4 command for process complete

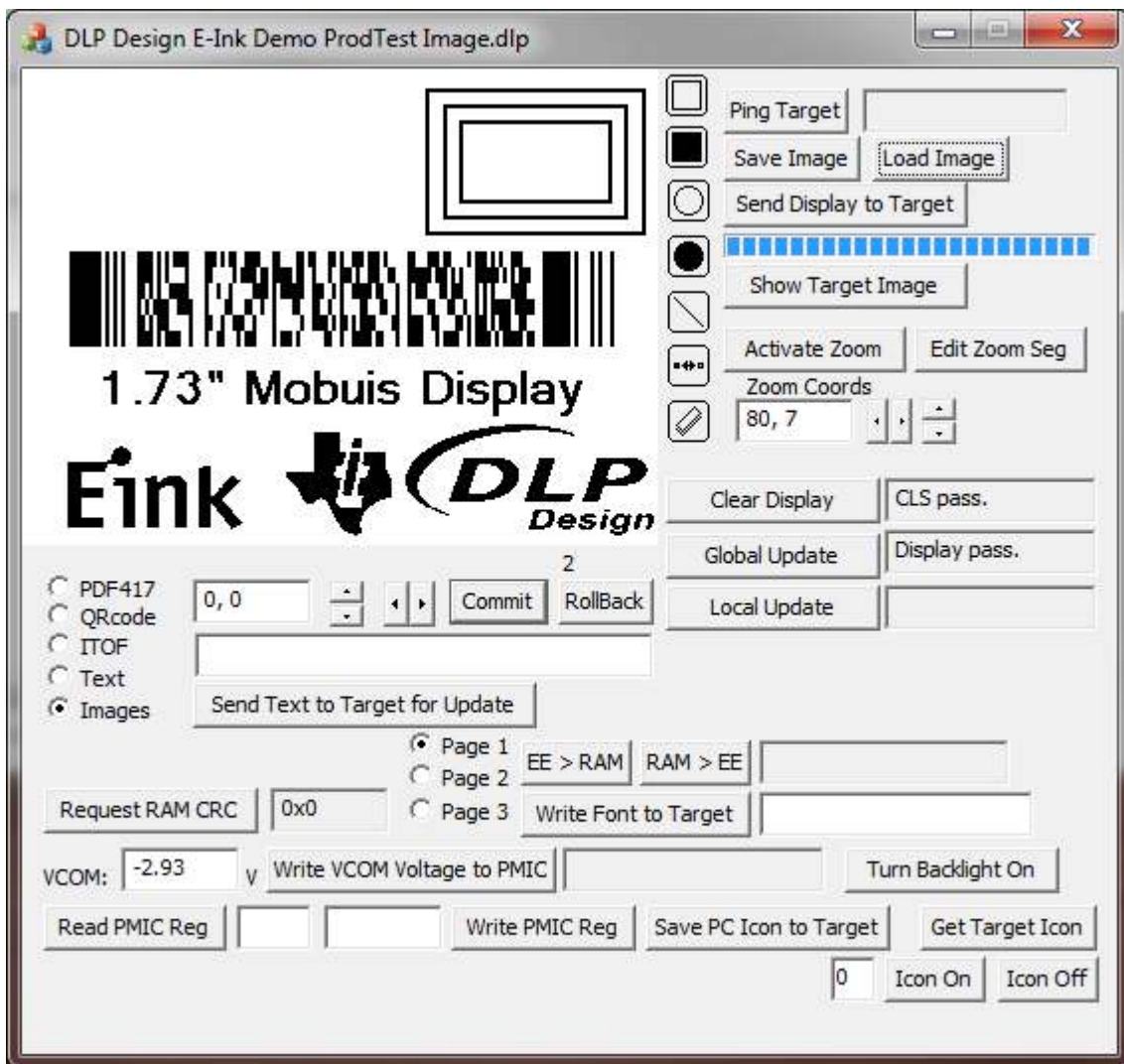
0xF5	SOF	Start of frame
0x04	SZ	Packet size (0x04)
0x95	CMD	Reply (0x95)
0x01	CHKSM	XOR of preceding bytes, seeded with 0x55

(No payload, just CHKSM)

5.0 WINDOWS PC GUI FEATURES

The provided Windows GUI communicates with the DLP-ECON-BP via USB interface using the FT232RL USB UART IC from FTDI (www.ftdichip.com). While this USB chip has both the VCP (Virtual COM Port) and D2XX (programming API) interfaces, the provided GUI uses the D2XX interface; therefore, the VCP drivers must be disabled before it will work correctly with the hardware.

See Section 2.3.5 of the Quick-Start Guide or the notes at the bottom of the page at www.dlpdesign.com for more details. Further reading on the topic of VCP vs. D2XX drivers can also be found at <http://www.dlpdesign.com/tnt/VCP%20Vs%20D2XX.pdf>.



5.1 PING

Use this feature to verify that you have successfully connected the GUI to the DLP-ECON-BP BoosterPack. This command also returns the current firmware version.

5.2 CREATE FULL-SCREEN IMAGES

The GUI has several features that make it easy to create images that will fill the E Ink display. These images can then be saved to a PC hard drive and loaded to the BoosterPack for presentation on the E Ink display. Full-screen images are about 20K bytes in size and take about 1 second to upload to the BoosterPack.

5.3 CLEAR DISPLAY

This function will cycle the entire display from black to white then black to white again. This serves to both clear the display and rejuvenate any locations in the display that may have been overdriven by other hardware.

5.4 GLOBAL UPDATE

This function is used to take a full-screen image (20K bytes) in the frame buffer and load it to the display all at once. Issuing a Global Update command to the DLP-ECON-F1 will invalidate the CRC for the image that was held in the frame buffer before performing the update to the E Ink display.

5.5 LOCAL UPDATE

This function is called after first having sent one or more small updates to the frame buffer. Examples of this type of data update would be an icon update (on/off) or text placement. The only pixels that change on the display are those required to change by the new data. Issuing a Local Update command will invalidate the CRC for the image that was held in the frame buffer before performing the update to the E Ink display.

5.6 LOAD IMAGE

Used to load icons or images from the PC's hard drive to the working window in the GUI.

5.7 SAVE IMAGE

Used to save icons or full-screen images to the PC's hard drive from the working window in the GUI.

5.8 SEND DISPLAY TO TARGET

This feature takes the current image in the working window of the GUI and sends it to the frame buffer in the DLP-ECON-F1 controller. The image can then be loaded to the display using the Global Update feature.

5.9 SHOW TARGET IMAGE

This feature reads the current image data from the frame buffer in the DLP-ECON-F1 and displays it in the GUI's working window. (Note that the image data cannot be read from the E Ink display itself.)

5.10 EE > RAM

This feature reads a full-screen image from one of three non-volatile memory locations and loads it into the frame buffer for presentation to the display via a Global Update.

5.11 RAM > EE

This feature saves the data in the frame buffer to one of three non-volatile memory locations.

5.12 BARCODES / TEXT / BITMAPS

The GUI has the ability to create custom barcodes and place them anywhere in the GUI edit window or the E Ink display. It will create PDF417, QR code and ITOF-type (Interleave Two of Five) barcodes.

- Text can be entered and placed anywhere on the screen using the built-in character set.
- Monochrome bitmap files can be loaded from the PC's hard drive and placed anywhere within the GUI edit window or E Ink display.
- Left, Right, Up and Down arrow buttons are used to position these items before they are committed to a specific position in the GUI working window.

5.13 TURN BACKLIGHT ON/OFF

This GUI feature toggles the E Ink display backlight on and off.

WARNING: The LED backlight is very bright. Do not look directly at the backlight.

5.14 REQUEST FRAME BUFFER CRC

Use this feature to request a CRC check of the frame buffer. Note that this feature returns valid CRC's just after transferring images to or from the target device. Once a Global Update or Local Update has been performed, the CRC value internal to the target device becomes invalid.

5.15 PROGRAM VCOM VOLTAGE

Each E Ink display requires a unique VCOM voltage level to be programmed into the power supply IC (PMIC) on the DLP-ECON-F1 controller. This voltage is printed on each E Ink display. The ability to program this voltage into the controller is provided in the GUI even though this programming step has already been performed at the factory during production of the DLP-ECON-BP BoosterPack.

5.16 PMIC READ/WRITE

The ability to read and update the control registers in the PMIC is provided by the GUI. Changes to these registers is neither recommended nor supported.

5.17 ACTIVATE ZOOM

This button creates a 32x32-pixel edit window for creating and editing icons. The position of the edit window can be changed in the GUI work area using the GUI's Left, Right, Up and Down buttons.

5.18 EDIT ZOOM SEG

This button opens a second dialog window showing a 32x32-pixel area which is enlarged for editing purposes. This allows for toggling single pixels as well as editing other drawing shapes within the enlarged area.

5.19 SAVE ICON TO TARGET

This button sends a newly created icon to non-volatile memory within the DLP-ECON-F1 for storage. Icons are stored using a zero-based access numbering system (0-89). Before clicking this button, choose which of the 90 memory locations in the target device where the icon is to be stored. Enter this location selection value in the Edit box below the Save Icon to Target button.

Icons are usually displayed in one of 20 locations on the display of the target device (0-9 along the top and 10-19 along the bottom). When you click this button, you can choose an icon file to send to the display. Once you choose the file, the icon is sent to the target device. The icon is stored at a memory location 0-89, and displayed at location 0-19 when requested later.

5.20 GET TARGET ICON

This button retrieves the specified icon from the target device and displays it in the GUI. The icon will display in the GUI at the location stored within the icon itself when it was sent to the target device. Note that this may have changed from the original file on your computer's hard drive if you chose an icon location of 0-19 as described in the directions for SAVE ICON TO TARGET above.

5.21 DRAWING FUNCTIONS: LINE, BOX, BOX FILLED, CIRCLE, CIRCLE FILLED, PIXEL

These simple drawing tools can help put the finishing touches on a new image design. They are presented in a vertical column of buttons along the right side of the GUI edit window with icons showing what each button does. For example, to draw an open circle, click on the button with the open circle, then move the cursor into the display. To begin drawing the circle, hold down the left mouse button while dragging the cursor, and release the button to freeze the circle in place. (If you don't like the circle drawn, just click on the Rollback button and start over.)

5.22 ICON ON/OFF

These buttons will display or hide an icon on the E Ink display. The desired icon number must be entered in the Edit box below the Save Icon to Target button. The icon is loaded to the display at the position stored with the icon itself.

5.23 SEND TEXT TO TARGET FOR UPDATE

This button will take the text found in the text edit box and send it the target device for later display at a location specified in the location edit box of the GUI. Note that the text will NOT instantly appear on the target device display. In order to display the text on the target device, use either the LOCAL UPDATE button or the GLOBAL UPDATE button.

5.24 COMMIT

This button will "freeze" whatever is on the display so that you can begin using another feature such as loading in an image file to overlay an existing item. Using the Commit button will once again freeze the new arrangement, and so on.

5.25 ROLLBACK

This button will undo the most recent Commit action(s) described above. It will do so for each item or items for which you had used the Commit button. In other words, if you used Commit three times, you could back down the list three times, thereby recovering three previous edits to the GUI.

6.0 ICON TUTORIAL

This tutorial will walk the user through creating a new icon, saving it to the hard drive, writing it to non-volatile storage on the target and displaying it on the E Ink display.

6.1 CREATE AND SAVE ICON

- Activate Zoom and reposition the 32x32 pixel window if desired.
- Click on Edit Zoom Segment.

- Add lines, boxes, circles and pixels as desired to create the icon.
- Click on Save Image. (Create a folder for icons if you don't already have one.)
- Give the image a file name (the extension .dip is automatically applied).
- Click on the Done Editing button in the Zoomed Segment window.
- Click on the Deactivate Zoom button to close the zoom window.

6.2 DISPLAY ICON IN GUI DISPLAY

- Click on the Eraser button to start with a blank GUI display.
- Click on Load Image, select the file just created and click on Open.
- Reposition the icon in the GUI as desired, and click on Commit.

6.3 SAVE ICON TO TARGET MEMORY

- Click on the Eraser button to start with a blank GUI screen.
- Click on Load Image, select an icon and click on Open.
- Reposition the icon in the GUI to the upper-left corner (0,0) and click on Commit.
- Save the icon to the hard drive again. This time the icon will be save with a screen location of 0,0. (You may wish to give the new file a different name.)
- Click on Activate Zoom, and move the Zoom Segment to Zoom Coordinate 0,0.
- Click on Save Image, and assign a name to the file.
- Click on De-Activate Zoom.
- Enter a zero in the edit box (for icon number 0 of icons 0-89 in non-volatile storage) under the Save Icon to Target button.
- Click on Save Icon to Target. Select the new icon and click on Open. The new icon is now stored in location 0 of the icon storage area in the target.

6.4 DISPLAY ICON ON E INK DISPLAY

- With a 0 in the edit box, click on "Icon On".
- The icon image will be read from non-volatile memory in the target and displayed on the E Ink display at the location saved with the icon.

6.5 DISPLAY ICON ON E INK DISPLAY USING LAUNCHPAD

- Program demonstration firmware into the G2 LaunchPad.
- Move the jumpers on the BoosterPack from GUI to LP.
- Press Button 1 to clear the display.
- Press Button 2 to start reading the humidity sensor.
- Place a finger on the humidity sensor to make the humidity level rise. Once the humidity rises above 50%, the new icon will turn on. This icon was stored at Location 0 in non-volatile memory. Location 0 is called by the G2 (MSP430G2553) firmware in the LaunchPad to indicate a high-humidity condition.

- Remove your finger from the sensor, and the humidity will start to fall. As the level falls below 50%, the icon will turn off. (Consult the G2 firmware for this BoosterPack to learn how to read the sensor and control the icons.)

7.0 BOOSTERPACK FIRMWARE SOURCE CODE

The firmware source code for the G2 microcontroller in the LaunchPad is available for download upon purchase of the BoosterPack. (Refer to the URL located on the outer packaging of the DLP-ECON-F1.)

8.0 BOOSTERPACK SCHEMATICS

Schematics for the DLP-ECON-BP are located at the end of this datasheet.

9.0 DISCLAIMER

Neither the whole nor any part of the information contained herein nor the product described in this datasheet may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder.

This product and its documentation are supplied on an as-is basis, and no warranty as to their suitability for any particular purpose is either made or implied. DLP Design will not accept any claim for damages whatsoever arising as a result of the use or failure of this product. Your statutory rights are not affected.

This product or any variant of it is not intended for use in any medical appliance, device or system in which the failure of the product might reasonably be expected to result in personal injury.

This document provides preliminary information that may be subject to change without notice.

10.0 CONTACT INFORMATION

DLP Design, Inc.
1605 Roma Lane
Allen, TX 75013

Phone: 469-964-8027
Fax: 415-901-4859
Email: support@dlpdesign.com
Internet: <http://www.dlpdesign.com>