Device: EDR-200200B

This document Version: 1c

Date: 11 November 2016

Matches module version: v3 [25 Aug 2016]

Description: e-Paper Display Driver and 200x200 e-Paper Display

Contents

Introduction

The EDR-200200B is a driver board for the e-Paper displays driven by the SSD1607 driver. The e-Paper display is a 200x200 pixel low-power display.

Features

The EDR-200200B is designed to run on 3.3V and communicates with your microcontroller via SPI. The display only needs to be on during the update process and can be turned off or placed in sleep mode. The display consumes ~900uA of current while updating, but only 18uA-21uA while idle. This pair is a must-have for low power graphical applications.

Hackability

The EDR-200200B is 100% hackable.

At Embedded Adventures, we believe you have the most fun when you have the most control over your hardware. For the EDR-200200B we provide a datasheet and complete schematic. After that, it's all up to you. We'd love to hear about the projects you're using it for – send us information and photos to myproject@embeddedadventures.com

Construction

It's all pre-built! Just add female or male header pins, or solder directly to the board, connect the ePaper display, and away you go.

Connections

VCC	Positive supply, 3.3V
DATA	SPI Data
CLK	SPI clock
CS	Chip Select, active low
D/C	Data – '1' Command – '0'
RST	Reset, active low
BUSY	Busy, active high
GND	Ground connection

Power

The EDR-200200B is powered with 3.0V-3.3V.

NOTE: Powering the display with 5V *will* burn out the display and the magic smoke will come out. Please use only 3-3.3V.

Programming

Commands

Commands are sent by pulling D/C low (command mode), sending the command byte, then setting D/C high to send any data bytes required for the command.

<table>
<tr><th>Cmd</th><th>Function</th><th>Parameters</th></tr>
<tr><td>0x01</td><td>Gate driver output control</td><td>
<table>
<tr><th>Prm</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>0</td><td colspan="8">A<7:0></td></tr>
<tr><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>A<8></td></tr>
<tr><td>2</td><td></td><td></td><td></td><td></td><td></td><td>GD</td><td>SM</td><td>TB</td></tr>
</table>
A<8:0> - MUX setting as A<8:0> + 1. POR = 0x12b + 1
GD Selects first output gate
SM Scanning order of gate driver (0=Non interlaced, 1=interlaced)
TB Scan order (0=G0 to G299, 1=G299 to G0)
Recommended settings: A = 0xc7, GD=0, SM=0, TB=0
</td></tr>
</table>

<table>
<tr><th>Cmd</th><th>Function</th><th>Parameters</th></tr>
<tr><td>0x10</td><td>Deep Sleep</td><td>
<table>
<tr><th>Param</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>A<0></td></tr>
</table>
A<0> - Sleep mode (0= Normal running mode, 1= Deep Sleep mode)
</td></tr>
<tr><td>0x0C</td><td>Soft start control</td><td>
<table>
<tr><th>Param</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>0</td><td colspan="8">A<7:0></td></tr>
<tr><td>1</td><td colspan="8">B<7:0></td></tr>
<tr><td>2</td><td colspan="8">C<7:0></td></tr>
</table>
A<7:0> - Soft start setting for phase 1
B<7:0> - Soft start setting for phase 2
C<7:0> - Soft start setting for phase 3
Recommended settings: A = 0xd7, B = 0xd6, C = 0x9d
</td></tr>
<tr><td>0x2C</td><td>VCOM Voltage</td><td>
<table>
<tr><th>Param</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>0</td><td colspan="8">A<7:0></td></tr>
</table>
A<7:0> - VCOM register setting
Recommended settings: A=0xa8
</td></tr>
<tr><td>0x3A</td><td>Set dummy line</td><td>
<table>
<tr><th>Prm</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>0</td><td>-</td><td colspan="7">A<6:0></td></tr>
</table>
A<6:0> - Length of dummy line period (measured in TGate)
Recommended settings: A = 0x1a
</td></tr>
<tr><td>0x3B</td><td>Gate on-time setting</td><td>
<table>
<tr><th>Prm</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>0</td><td>-</td><td colspan="7">A<6:0> ?</td></tr>
</table>
A<6:0> - Length of dummy line period (measured in TGate)
Recommended settings: A = 0x08 (2uS per line)
</td></tr>
<tr><td>0x11</td><td>RAM data entry mode</td><td>
<table>
<tr><th>Prm</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>0</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>AM</td><td colspan="2">DES<1:0></td></tr>
<tr><td>0</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td colspan="3">A<2:0></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="3"></td></tr>
</table>
DES<1:0> - Data entry mode
00 – Y decrement, X decrement
</td></tr>
</table>

<table>
<tr><th>Cmd</th><th>Function</th><th>Parameters</th></tr>
<tr><td></td><td></td><td>01 – Y decrement, X increment
10 – Y increment, X decrement
11 – Y increment, X increment
AM - direction in which the address counter is updated automatically after data are written to RAM (0= the address counter is updated in the X direction, 1= the address counter is updated in the Y direction)
Recommended settings: AM=0, DES<1:0> = 01h</td></tr>
<tr><td>0x44</td><td>Set start/end positions for X direction in RAM</td><td>
<table>
<tr><th>Param</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>0</td><td colspan="8">Xstart<7:0></td></tr>
<tr><td>1</td><td colspan="8">Xend<7:0></td></tr>
</table>
Xstart - Start X position /8
Xend - End X position /8
Setting Xstart and Xend allow setting partial (or full) updates on 8 bit boundary
Recommended settings: 0x00, 0x18 (for full width update)
</td></tr>
<tr><td>0x45</td><td>Set start/end positions for Y direction in RAM</td><td>
<table>
<tr><th>Param</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>0</td><td colspan="8">Ystart<15:8></td></tr>
<tr><td>1</td><td colspan="8">Ystart<7:0></td></tr>
<tr><td>2</td><td colspan="8">Yend<15:8></td></tr>
<tr><td>3</td><td colspan="8">Yend<7:0></td></tr>
</table>
Ystart - Start Y position -1
Yend - End Y position -1
Setting Ystart and Yend allow setting partial (or full) updates on a line boundary. Since the recommended settings are for decrementing Y lines send from the buffer, Ystart should be larger than Yend.
Recommended settings: Ystart = 0x00c7, Yend = 0x0000 (for full height update)
</td></tr>
<tr><td>0x4E</td><td>Set RAM X address counter</td><td>
<table>
<tr><th>Param</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>0</td><td colspan="8">Xpos<7:0></td></tr>
</table>
Xpos - Start X position /8
Recommended settings: 0x00 (for full screen write)
</td></tr>
<tr><td>0x4F</td><td>Set RAM Y address counter</td><td>
<table>
<tr><th>Param</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>0</td><td colspan="8">Ypos<15:8></td></tr>
<tr><td>1</td><td colspan="8">Ypos<7:0></td></tr>
</table>
Ypos - Start Y position
</td></tr>
</table>

<table>
<tr><th>Cmd</th><th>Function</th><th>Parameters</th></tr>
<tr><td></td><td></td><td>Recommended settings: 0x00c7 (for full screen write)</td></tr>
<tr><td>0x32</td><td>Write to LUT register (30 bytes)</td><td>30 byte LUT defaults for partial and full updates are provided in EA source code.</td></tr>
<tr><td>0x22</td><td>Set display update sequence</td><td>0xC0 for enabling clock signal</td></tr>
<tr><td>0x20</td><td>Start display update sequence</td><td></td></tr>
<tr><td>0x24</td><td>Write data to RAM</td><td>
<table>
<tr><th>Param</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>0</td><td colspan="8">ScreenData[0]</td></tr>
<tr><td>…</td><td colspan="8">…</td></tr>
<tr><td>n</td><td colspan="8">Ypos[n]</td></tr>
</table>
</td></tr>
<tr><td>0xFF</td><td>NOP</td><td>No operation done, but used to terminate memory read/write commands</td></tr>
</table>

Setup

The EDR-200200B comes with an Arduino library to make it easier to use the ePaper display. The driver requires a series of commands and configuration settings before getting started:

- Reset the device by pulling the RST line low for 100ms
- Initialize gate driver output control settings
- Initialize soft start settings
- Initialize VCOM settings
- Initialize dummy line settings
- Initialize gate time settings
- Initialize GDRAM data entry settings
- Initialize X and Y RAM settings
- Enable display update sequence
    - Enable clock signal with 0xC0

Don't let this overwhelm you: we've wrapped up all this in the Arduino library and made it easier for you to start using the display. Only two commands necessary for the display:

- *Invert(true)*
- *Init()*

NOTE: the driver does not have an internal invert command so *invert()* simply inverts the data bits that will go into the GDRAM.

Updating Display

- Updating the RAM
  - Set RAM pointer to the first RAM address
  - 0x24 – Write RAM command
  - Follow command with the graphic data bytes for display
  - Display update and enable clock signal
  - Activate display update sequence
  - Terminate memory read/write with NOP

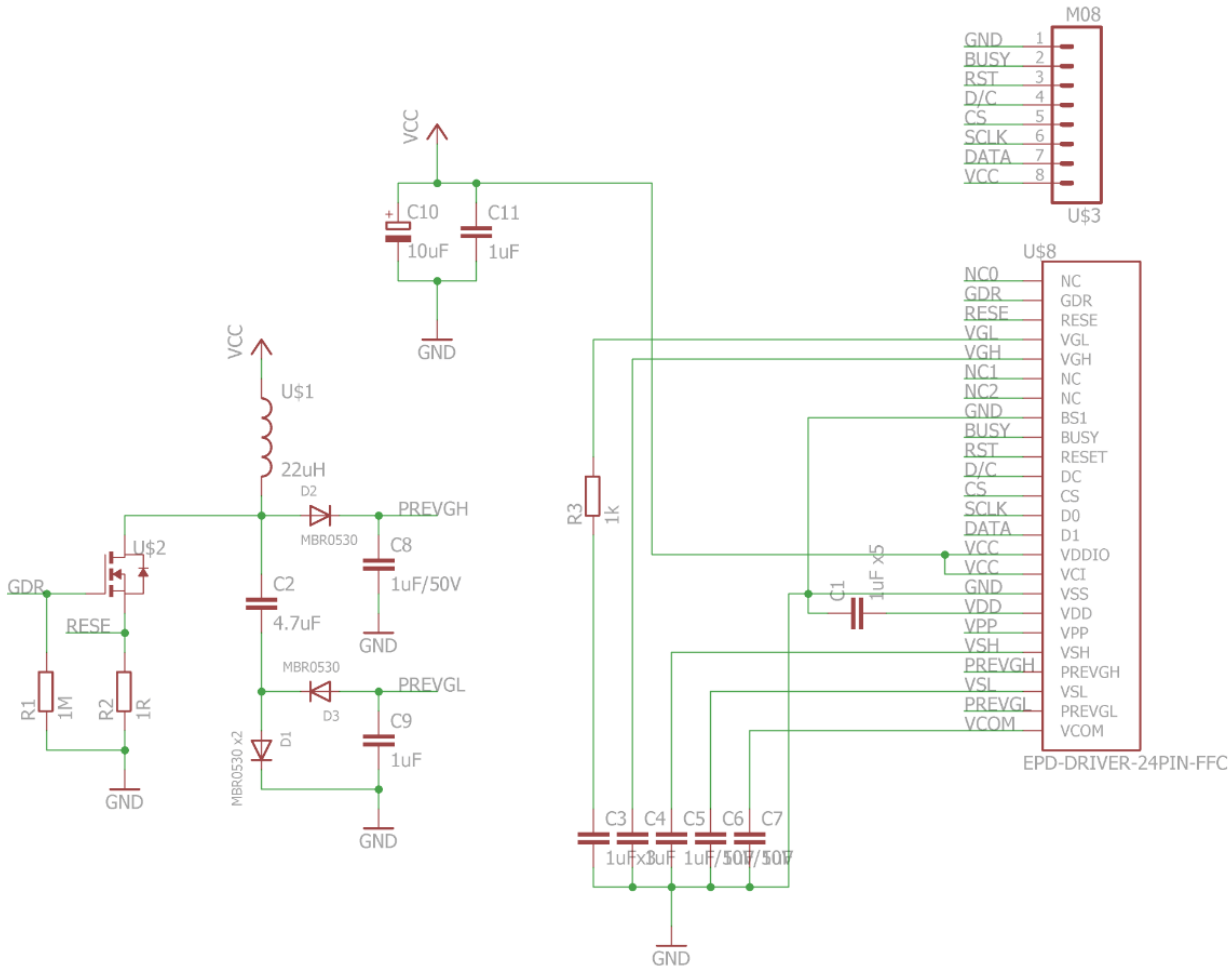
Low power mode:

Turn off oscillator clock and DC/DC & regulator

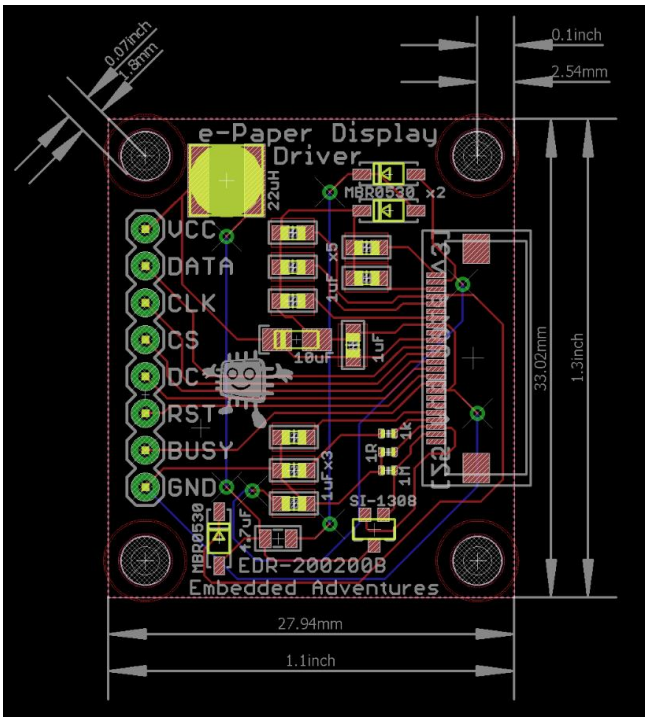
Deep sleep: DC/DC off No clock No input load Ram data not retain (command 0x10)

Sleep: DC/DC off No clock No input load Ram data retain

Schematic


M08
GND 1
BUSY 2
RST 3
D/C 4
CS 5
SCLK 6
DATA 7
VCC 8
U$3
U$8
NC0
GDR
RESE
VGL
VGH
NC1
NC2
GND
BUSY
RST
D/C
CS
SCLK
DATA
VCC
VCC
GND
VDD
VPP
VSH
PREVGH
VSL
PREVGL
VCOM
NC
GDR
RESE
VGL
VGH
NC
NC
BS1
BUSY
RESET
DC
CS
D0
D1
VDDIO
VCI
VSS
VDD
VPP
VSH
PREVGH
VSL
PREVGL
VCOM
EPD-DRIVER-24PIN-FFC
VCC
C10
10uF
C11
1uF
GND
VCC
U$1
22uH
D2
MBR0530
PREVGH
C8
1uF/50V
GND
C2
4.7uF
MBR0530
D3
PREVGL
C9
1uF
GND
MBR0530 x2
D1
U$2
GDR
RESE
R1
1M
R2
1R
GND
R3
1k
C1
1uF x5
C3
C4
C5
C6
C7
GND

PCB



Versions

Version	Date	Comments
1	22 Oct 2016	Initial Version for board v3 [25 Aug 2016]
1c	3 Nov 2016	Updated with more detail on EPD commands