IDM2040-21R Datasheet

1 Introduction

IDM2040-21R is a 2.1-inch Rotary Dial Display, which was designed using Raspberry Pi RP2040 microcontroller with FT800Q, a rotary encoder with button, RS-485 transceiver and a 2.1-inch round LCD.

1.1 Features

- The host MCU RP2040 is a microcontroller unit (MCU) developed by Raspberry Pi, Dual ARM 32-bit Cortex-M0+ Core processors running at up to 133 MHz, 264KB of SRAM, and 8MB of on-board Flash memory.
- 2.1-inch Round LCD Screen: 480x480 resolution with capacitive touch support.
- 1 x I2C: Inter-Integrated Circuit for connecting
- multiple peripherals.
- 1 x UART: Universal Asynchronous Receiver-Transmitter for serial communication.
- 1 x RS-485: for long distance and reliable communication with other devices.
- 1 x Audio Output: for audio applications.
- RGB666 hardware interface for high-quality color display.
- Rotary Encoder: 24 steps with a push button for input and control.
- Touch Interface: The display includes pins for Touch Panel Interface (SDA, SCL, RST#, INT#, etc.), which are used for I2C communications with the touch controller CST826 which is compatible with FT3267.
- LCD Resolution: 480x480 pixels, display type: IPS TFT LCD, controller: ST7701S, brightness: 300 cd/m² (typical), viewing angle: 80° in all directions.
- Single 5V power supply from USB-C or 6-pin JST connector.
- Supports C/C++, MicroPython, CircuitPython development.
- 1xTYPE C USB Connector: provides 5V and can be used for firmware upgrade.
- Supports USB upgrade firmware.
- Debugging and Programming: BOOTSEL mode for programming via USB (drag-and-drop UF2 support)

1.2 Typical Applications

- Home Automation Systems: Smart Thermostats, Lighting Control.
- Audio Equipment: Mixing Consoles, Hi-Fi Systems.
- Medical Devices: Portable Medical Monitors, Patient Monitoring Devices.
- Industrial Applications: Control Panels, Measurement Instrument.
- Automotive Interfaces: Infotainment Systems, Dashboards
- Consumer Electronics: Smart Speakers and Media Players, portable Gaming Consoles.
- Test and Measurement Equipment such as Oscilloscopes and Multimeters.
- …… and many more.

2 Ordering Information

Part No.	Description
IDM2040-21R	2.1-inch Rotary Dial TFT Display

Table 1 - Ordering Information

3 Block Diagram

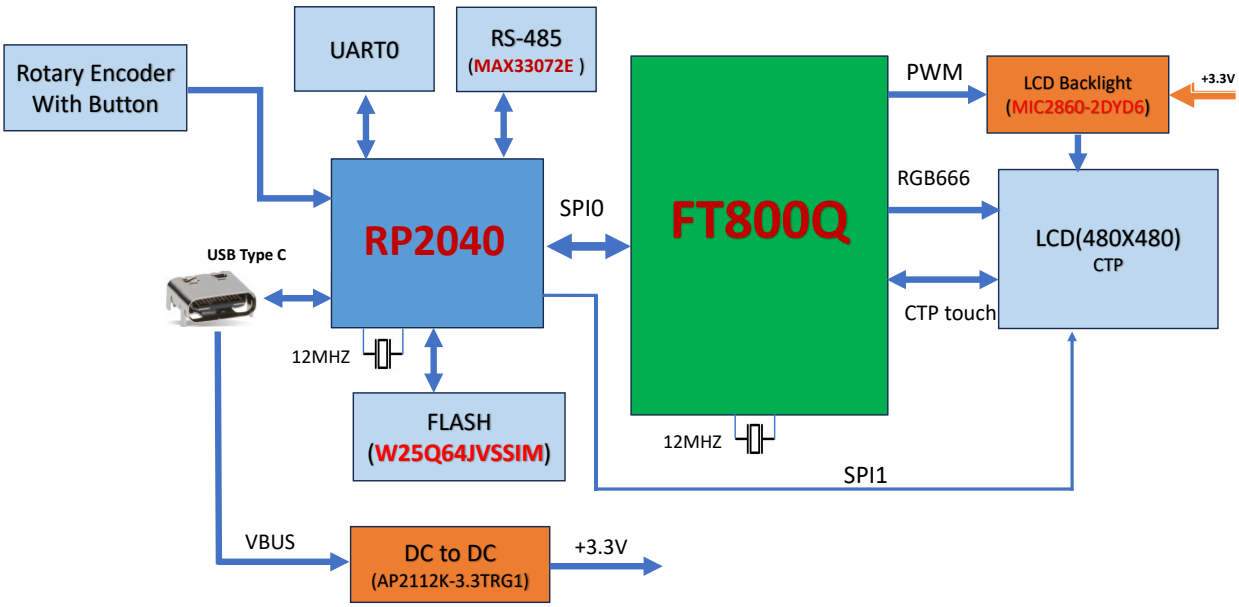

Figure 1 - IDM2040-21R Block Diagram

Figure 2 – IDM2040-21R Front and Back View

Table of Contents

4 Hardware Description

4.1 Schematic Description

4.1.1 DC to DC converter

- A DC-DC step-down converter circuit based on the AP2112K-3.3TRG1, which provides a regulated +3.3V output from a +5V USB VBUS input.
- The AP2112K-3.3TRG1 is a low-dropout linear regulator. It regulates the +5V input down to +3.3V. Pin 3 of U1 (EN, enable pin) is connected to VIN (+5V) to enable the regulator. This ensures that the output is active whenever the +5V input is present.
- The device can provide up to 600mA of output current, which is suitable for powering the RP2040 microcontroller, FT800 graphics controller, and LCD panel.
- Capacitors C3 and C4 are used for filtering the +3.3V output to stabilize the voltage and reduce noise.
- An LED (D1) with a current-limiting resistor R1 is connected across the +3.3V output to indicate power status. The LED lights up when the +3.3V output is active.

4.1.2 Host MCU and Peripherals

- IDM2040-21R uses RP2040 as HOST MCU. RP2040 is designed to work at 3.3V and IO level at 3.3V, all IOVDD PIN (PIN1, 10, 22, 33, 42, 49) and USB_VDD(PIN48), VREG_VOUT(PIN 45), VREG_VIN(PIN 44) and ADC_AVDD PIN(PIN 43) need to connect to +3.3V.
- DVDD PIN(50,23) is digital core power supply, nominal voltage 1.1V can be connected to VREG_VOUT.
- The RP2040 microcontroller supports external Quad-SPI (QSPI) flash memory for storing program code and data. The maximum supported flash size is 16MB (128 Mbit) due to its 24-bit address bus. IDM2040-21R uses 8MB flash, U4 (W25Q64JVSSIQ), to store the user's application code and resources such as pictures, fonts, audios, etc.

Pin	Signals	Description
51	QSPI_SD3	Data I/O line 3
52	QSPI_SCK	Flash clock signal
53	QSPI_SD0	Data I/O line 0
54	QSPI_SD2	Data I/O line 2
55	QSPI_SD1	Data I/O line 1
56	QSPI_SS	Chip select for the flash

Table 2 - RP2040 QSPI Interface with Flash Memory

- SPI0 is used to communicate with the FT800 graphics controller in single SPI mode. The maximum SPI clock can be 31.25MHZ. RP2040 will use SPI0 to communicate with FT800.

Pin	Signals	Description
4	SPI0_SCK	SPI Serial Clock
5	SPI0_MOSI	SPI serial data output to FT800.
6	SPI0_MISO	SPI serial data output from FT800
7	SPI0_CS#	Chip Select output, Active: LOW
8	EVE_INT#	Interrupt signal from FT800, Active: LOW
9	EVE_PD#	Reset signal to FT800, Active: LOW

Table 3 - SPI0 Interface with FT800

- The UART0 (PIN2, PIN3) is routed to CN2 for user access. It can be used to connect to external devices such as Bluetooth or Wi-Fi modules, among others.

- The UART1 (PIN11, PIN12) is routed to U3, an RS-485 transceiver, allowing the RP2040 to connect to external RS-485 networks and devices. UART1_DE (PIN39) is used for data transmit enable, while UART1_POL (PIN27) controls and swaps the RS-485 output polarity.

- The I2C1 port (PIN29, PIN30) is connected to CN3 for interfacing with external I2C devices.

- The SPI1 port (PIN13, PIN14, PIN15, PIN28, and PIN24) is used to initialize the LCD panel. These pins can also be configured as GPIO outputs and used with GPIO SPI to initialize the LCD.

Pin	Signals	Description
13	SPI1_SCK	SPI Serial Clock
14	SPI1_MOSI	SPI serial data output to LCD
15	SPI1_MISO	Not used
28	SPI1_CS#	Chip Select output, Active: LOW
24	LCD_RESET#	LCD Reset Signal, Active: LOW

Table 4 - SPI1 Interface with LCD

- PIN37 (GPIO25) of the RP2040 is used as an output pin to drive an LED for status indication.

- RP2040 has an integrated USB1.1 PHY and controller which can be used in both Device and Host mode. Two required 27Ω external resistors (R4 and R5) brings this interface to a standard micro-USB port. The USB port can be used to access the USB bootloader (BOOTSEL mode) stored in the RP2040 boot ROM.

- The rotary encoder's output pins SP1 and SP2 are connected to PIN23 and PIN24 of the RP2040, where they are configured as input pins.

- The rotary encoder's button output pin is connected to PIN32 of the RP2040, where it is configured as an input pin.

4.1.3 Touch Interface

4.1.3.1 FT800 with Resistive Touch Panel

- The FT800 typically uses Resistive touch, however in this module we have implemented a capacitive touch panel .

4.1.3.2 FT800 with Capacitive Panel

- The FT800 Embedded Video Engine (EVE) does not directly support capacitive touch panels; however, with additional hardware and proper firmware configuration, it can work with a capacitive touchscreen. Here's an overview of how to connect a capacitive touchscreen to the FT800 and set up the necessary hardware.

- The FT800 also uses PIN 19, 20,21,22 to interface with the capacitive touch panel, see Table 5.

Pin	Resistive Touch Signals	Capacitive Touch Signals	Description
19	X+	RST#	Reset signal for touch panel
20	Y+	INT#	Interrupt signal from touch panel
21	X-	SCL	I2C serial clock
22	Y-	SDA	I2C serial data

Table 5 - FT800 Capacitive Touch Interface

- PIN 19 (CTP_RST#) is used to reset the CTP controller, and it must be driven low for proper reset behaviour. A 100kΩ pull-down resistor (R21) is required to ensure that PIN 19 is pulled low during reset. Since PIN 19 is a low-drive strength pin, a buffer (U7) is used to increase its drive strength. This ensures that the reset signal is properly driven low, ensuring proper reset for the capacitive touch panel.

- The FT800's default firmware supports resistive touch, so to enable capacitive touch, the application must download the appropriate capacitive touch firmware into the FT800's RAM. Refer to 7.3 for the pseudo code to download capacitive touch firmware into FT800.

4.1.4 LCD Backlight

- IDM2040-21R LCD backlight uses 4 LEDs in parallel.

- The LCD backlight is driven by U6 (MIC2860-2D), which has a typical dropout of 52mV at 30.2mA per channel. This allows the LED to be powered directly from 3.3V. Pin 1 of U6 is connected to Pin 24 of the FT800, which is the LCD backlight control pin. This pin can output a PWM signal with an adjustable frequency and duty cycle.

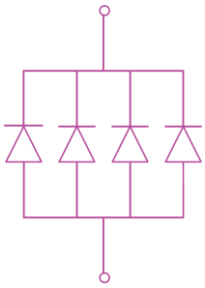

Figure 3 - LCD Backlight LED Connection

4.2 Connector Description

Figure 4 - IDM2040-21R PCB LAYOUT

4.2.1 CN1

CN1 is a Type-C USB connector used to provide +5V to the board and download firmware to the flash.

4.2.2 CN2

CN2 is an RS-485 output and UART output connector, via a 1.5mm pitch, 6-pin connector. See Table 6 for details. It is used to connect to an external RS-485 network.

Pin	Name	Type	Description
1	VCC5V	P	V
2	RS_485 terminal A/Y	I/O	RS_485 terminal A/Y
3	RS_485 terminal B/Z	I/O	RS_485 terminal B/Z
4	UART0_TX	O	UART0 transmit data line
5	UART0_RX	I	UART0 receive data line
6	Ground	P	Ground

Table 6 - RS485 Output

4.2.3 CN3

CN3 is the I2C1 output connector of the RP2040, used to connect peripheral I2C devices.

Pin	Name	Type	Description
1	GND	P	Ground
2	VCC3V3	P	DC +3.3 output
3	I2C1_SDA	I/O	I2C1 serial data
4	I2C1_SCL	O	I2C1 serial clock

Table 7 - CN3 PIN Assignment

4.2.4 CN4

CN4 is the external rotary encoder input.

Pin	Name	Type	Description
1	SP1	I	SP1, rotary encoder input
2	Ground	P	Ground
3	SP2	I	SP2, rotary encoder input
4	NC	-	NC

Table 8 - Rotary Encoder Input

4.2.5 CN5

CN5 is an FPC connector for connecting an external LCD. It has bottom contacts.

Pin	Name	Description
1	Touch Panel Interrupt signal, ACTIVE: LOW	
2	Touch Panel I2C SDA signal	
3	Touch Panel I2C Clock signal	
4	Touch Panel rest signal	
5	Ground	
6	RGB Signal: R5	
7	RGB Signal: R4	
8	RGB Signal: R3	
9	RGB Signal: R2	
10	RGB Signal: R1	
11	RGB Signal: R0	
12	RGB Signal: G5	
13	RGB Signal: G4	
14	RGB Signal: G3	
15	RGB Signal: G2	
16	RGB Signal: G1	
17	RGB Signal: G0	
18	RGB Signal: B5	
19	RGB Signal: B4	
20	RGB Signal: B3	
21	RGB Signal: B2	
22	RGB Signal: B1	
23	RGB Signal: B0	
24	RGB line synchronizing signal	
25	RGB frame synchronizing signal	
26	RGB data enable signal	
27	RGB pixel clock signal	
28	SPI Chip select signal	
29	SPI Clock signal	
30	SPI Data signal	
31	LCD Reset Signal, Active Low	
32	Power supply	
33	LED CATHODE	
34	LED ANODE	

35	Ground	
36	Ground	
37	Ground	

Table 9 - LCD Connector

4.2.6 SW1

- SW1 is the BOOTSEL button, used to select the boot mode from USB or flash.
- Press and Hold SW1 whilst powering up/resetting to enter boot mode.

Figure 5 - BOOTSEL and RESET Button

4.2.7 SW2

SW2 is the reset button for RP2040.

4.2.8 SW3

SW3 is the push button of the rotary encoder input.

5 Graphical User Interface

The IDM2040-21R features four sets of menus for simulating Aircon control, Lighting control, Floor heating control, and Curtain control. Users can interact with the system through the touch panel, rotary encoder, and push button, ensuring an enhanced human-machine interface experience.

5.1 Aircon Control

Touch the icon to access the menu. To return to the previous menu, touch the return icon or press the button.

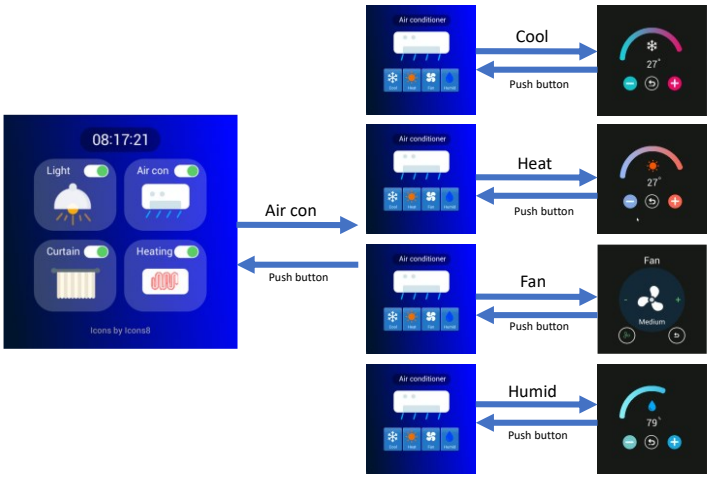

Figure 6 - Aircon Control menu

5.2 Lighting Control

Touch the icon to access the menu. To return to the previous menu, touch the return icon or press the button.

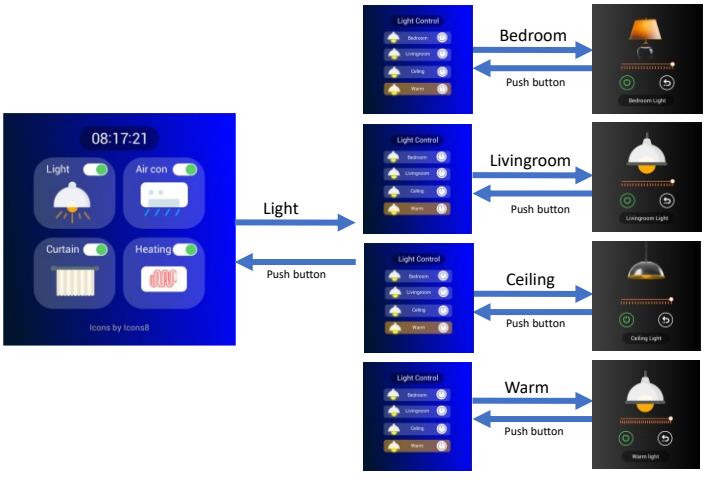

Figure 7 - Lighting Control menu

5.3 Floor Heating Control

Touch the icon to access the menu. To return to the previous menu, touch the return icon or press the button.

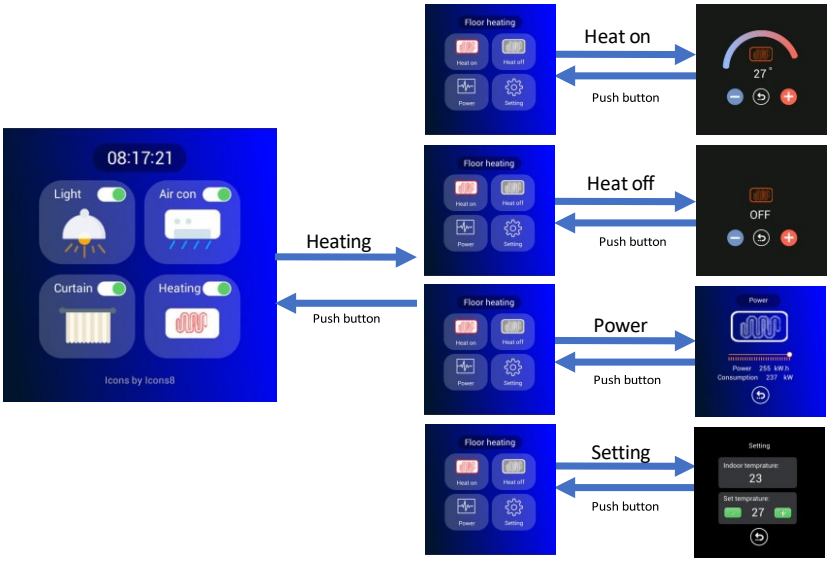

Figure 8 - Floor Control Menu

5.4 Curtain Control

Touch the icon to access the menu. To return to the previous menu, touch the return icon or press the button.

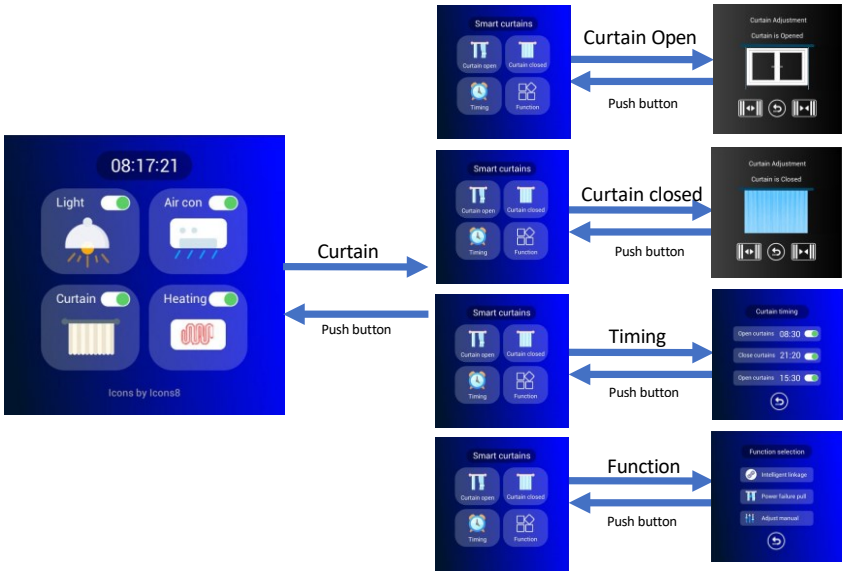

Figure 9 - Curtain Control Menu

6 Software Development

The RP2040 microcontroller can be programmed using C/C++, MicroPython, or CircuitPython.

6.1 C/C++ Environment Installation

- Download PICO C/C++ environment installation tools, pico-setup-windows-x64-standalone.exe from here.
- Double-click pico-setup-windows-x64-standalone.exe to run the installer. Click "Next" to complete the installation process.
- Download and install EVE Screen Designer, version above 4.19.4 or later.
- Use EVE Screen Designer to develop the code.

6.2 MicroPython Environment Installation

- Download the MicroPython UF2 file from MicroPython.
- Press and hold the BOOTSEL button on IDM2040-21R, then connect it to your computer using a USB cable.
- Release the BOOTSEL button once the RTPI-RP2 drive appears on your computer.
- Drag and drop the UF2 file onto the RPI-RP2 drive. The RP2040 will reboot and start running MicroPython.
- Download and install Thonny on your computer.
- Connect the IDM2040-21R to your computer. In Thonny, navigate to Tools > Options and select the "Interpreter" tab. From the drop-down list, choose MicroPython (Raspberry Pi Pico).
- Select the port labelled "USB Serial Device (COMX)" and use MicroPython with Thonny to develop the application code.

6.3 CircuitPython Environment Installation

- Download the CircuitPython UF2 file from CircuitPython .
- Press and hold the BOOTSEL button on IDM2040-21R, then connect it to your computer using a USB cable.
- Release the BOOTSEL button once the RTPI-RP2 drive appears on your computer.
- Drag and drop the UF2 file onto the RPI-RP2 drive. The RP2040 will reboot and start running CircuitPython.
- Download and install Thonny or Mu on your computer. It is recommended to install the Mu editor for CircuitPython applications.
- Connect the IDM2040-21R to your computer and run Mu. Select RP2040, and Mu with CircuitPython will be ready to use.
- Use CircuitPython with Thony or Mu Editor to develop the code.

7 Firmware Update

7.1 MCU Firmware Update

- Press and hold the BOOTSEL button on IDM2040-21R, then connect it to your computer using a USB cable.
- An RPI-RP2 drive will appear on the PC.
- Drag and drop the firmware.UF2 file onto the RPI-RP2 drive. The drive will then disappear, and the firmware will be programmed into the FLASH of the RP2040.
- The system will reboot and begin running the updated firmware.
- When the IDM2040-21R is connected to the PC, press and hold the BOOTSEL button, then press the RESET button. The RPI-RP2 drive will appear.

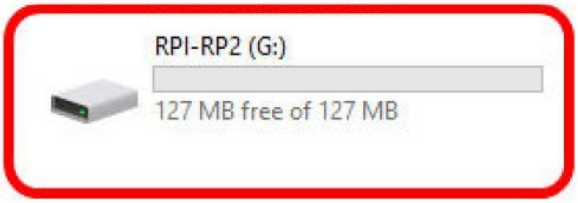

Figure 10 - RPI-RP2 Drive

7.2 C Code for LCD Initialization

The LCD of the IDM2040-21R must be initialized using a series of SPI commands.

Listing 7.1 LCD Initialization

```
#define LCD_SPI1_CS    17
#define LCD_SPI1_SCK   10
#define LCD_SPI1_MOSI  11
#define LCD_RESET      22

/********************************* LCD related ************************************/

void send_data(uint8_t data, bool cmd)
{
    uint16_t trans = 0;

    gpio_put(LCD_SPI1_CS, 0);
    EVE_sleep(1);
    if (cmd)
          trans = data << 7;
    else
          trans = 0x8000 | data << 7;
    spi_write16_blocking(spi1, &trans, 1);// 9 data bits
    EVE_sleep(1);
    gpio_put(LCD_SPI1_CS, 1);
}

void spi_write(spi_inst_t* spi, uint8_t* data, uint32_t length)
{
    //send command
    send_data(data[0], true);

    if (length > 1)
    {
          for (int i = 1; i < length; i++)
                send_data(data[i], false);
```

```
    }
}

void LCD_init()
{
    uint8_t data1[6] = { 0xFF, 0x77, 0x01, 0x00, 0x00, 0x10 };
    uint8_t data2[3] = { 0xC0, 0x3B, 0x00 };
    uint8_t data3[3] = { 0xC1, 0x0B, 0x02 };
    uint8_t data4[3] = { 0xC2, 0x07, 0x02 };
    uint8_t data5[2] = { 0xCC, 0x10 };
    uint8_t data6[17] = { 0xB0, 0x00, 0x11, 0x16, 0x0E, 0x11, 0x06, 0x05, 0x09,
    0x08, 0x21, 0x06, 0x13, 0x10, 0x29, 0x31, 0x18 };
    uint8_t data7[17] = { 0xB1, 0x00, 0x11, 0x16, 0x0E, 0x11, 0x07, 0x05, 0x09,
    0x09, 0x21, 0x05,0x13, 0x11, 0x2A, 0x31, 0x18 };
    uint8_t data8[6] = { 0xFF, 0x77, 0x01, 0x00, 0x00, 0x11 };
    uint8_t data9[2] = { 0xB0, 0x6D };
    uint8_t data10[2] = { 0xB1, 0x37 };
    uint8_t data11[2] = { 0xB2, 0x81 };
    uint8_t data12[2] = { 0xB3, 0x80 };
    uint8_t data13[2] = { 0xB5, 0x43 };
    uint8_t data14[2] = { 0xB7, 0x85 };
    uint8_t data15[2] = { 0xB8, 0x20 };
    uint8_t data16[2] = { 0xC1, 0x78 };
    uint8_t data17[2] = { 0xC2, 0x78 };
    uint8_t data39[2] = { 0xCD, 0x08 };
    uint8_t data38[2] = { 0xC3, 0x8C };
    uint8_t data18[2] = { 0xD0, 0x88 };
    uint8_t data19[4] = { 0xE0, 0x00, 0x00, 0x02 };
    uint8_t data20[6] = { 0xE1, 0x03, 0xA0, 0x00, 0x00, 0x04 };
    uint8_t data21[14] = { 0xE2, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00,0x00, 0x00 };
    uint8_t data22[5] = { 0xE3, 0x00, 0x00, 0x11, 0x00 };
    uint8_t data23[3] = { 0xE4, 0x22, 0x00 };
    uint8_t data24[17] = { 0xE5, 0x05, 0xEC, 0xA0, 0xA0, 0x07, 0xEE, 0xA0, 0xA0,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };
    uint8_t data25[5] = { 0xE6, 0x00, 0x00, 0x11, 0x00 };
    uint8_t data26[3] = { 0xE7, 0x22, 0x00 };
    uint8_t data27[17] = { 0xE8, 0x06, 0xED, 0xA0, 0xA0, 0x08, 0xEF, 0xA0, 0xA0,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };
    uint8_t data28[8] = { 0xEB, 0x00, 0x00, 0x40, 0x40, 0x00, 0x00, 0x00 };
    uint8_t data29[17] = { 0xED, 0xFF, 0xFF, 0xFF, 0xBA, 0x0A, 0xBF, 0x45, 0xFF,
    0xFF, 0x54, 0xFB, 0xA0, 0xAB, 0xFF, 0xFF,0xFF};
    uint8_t data30[7] = { 0xEF, 0x10, 0x0D, 0x04, 0x08, 0x3F, 0x1F };
    uint8_t data31[6] = { 0xFF, 0x77, 0x01, 0x00, 0x00, 0x13 };
    uint8_t data32[2] = { 0xEF, 0x08 };
    uint8_t data33[6] = { 0xFF, 0x77, 0x01, 0x00, 0x00, 0x00 };
    uint8_t data34[2] = { 0x36, 0x00 };
    uint8_t data35[2] = { 0x3A, 0x66,  };
    uint8_t data36 = 0x11;
    uint8_t data37 = 0x29;

    spi_init(spi1, 1000 * 1000);
    gpio_set_function(LCD_SPI1_SCK, GPIO_FUNC_SPI);
    gpio_set_function(LCD_SPI1_MOSI, GPIO_FUNC_SPI);
    spi_set_format(spi1, 16, SPI_CPOL_0, SPI_CPHA_0, SPI_MSB_FIRST);

/* Chip select is active-low, so we'll initialise it to a driven-high state */
    gpio_init(LCD_SPI1_CS);
    gpio_set_dir(LCD_SPI1_CS, GPIO_OUT);
    gpio_put(LCD_SPI1_CS, 1);
```

```
    // reset LCD
    gpio_init(LCD_RESET);
    gpio_set_dir(LCD_RESET, GPIO_OUT);
    gpio_put(LCD_RESET, 1);
    EVE_sleep(20);
    gpio_put(LCD_RESET, 0);
    EVE_sleep(20);
    gpio_put(LCD_RESET, 1);
    EVE_sleep(20);

    spi_write(spi1, data1, sizeof(data1));
    spi_write(spi1, data2, sizeof(data2));
    spi_write(spi1, data3, sizeof(data3));
    spi_write(spi1, data4, sizeof(data4));
    spi_write(spi1, data6, sizeof(data6));
    spi_write(spi1, data7, sizeof(data7));
    spi_write(spi1, data8, sizeof(data8));
    spi_write(spi1, data9, sizeof(data9));
    spi_write(spi1, data10, sizeof(data10));
    spi_write(spi1, data11, sizeof(data11));
    spi_write(spi1, data12, sizeof(data12));
    spi_write(spi1, data13, sizeof(data13));
    spi_write(spi1, data14, sizeof(data14));
    spi_write(spi1, data15, sizeof(data15));
    spi_write(spi1, data16, sizeof(data16));
    spi_write(spi1, data17, sizeof(data17));
    spi_write(spi1, data38, sizeof(data38));
    spi_write(spi1, data39, sizeof(data39));
    spi_write(spi1, data18, sizeof(data18));
    spi_write(spi1, data19, sizeof(data19));
    spi_write(spi1, data20, sizeof(data20));
    spi_write(spi1, data21, sizeof(data21));
    spi_write(spi1, data22, sizeof(data22));
    spi_write(spi1, data23, sizeof(data23));
    spi_write(spi1, data24, sizeof(data24));
    spi_write(spi1, data25, sizeof(data25));
    spi_write(spi1, data26, sizeof(data26));
    spi_write(spi1, data27, sizeof(data27));
    spi_write(spi1, data28, sizeof(data28));
    spi_write(spi1, data29, sizeof(data29));
    spi_write(spi1, data30, sizeof(data30));
    spi_write(spi1, data31, sizeof(data31));
    spi_write(spi1, data32, sizeof(data32));
    spi_write(spi1, data33, sizeof(data33));
    spi_write(spi1, data34, sizeof(data34));
    spi_write(spi1, data35, sizeof(data35));
    EVE_sleep(100);
    spi_write(spi1, &data36, 1);
    EVE_sleep(100);
    spi_write(spi1, &data37, 1);
}
```

7.3 Upload Capacitive Touch Firmware to FT800

After the FT800 has been initialised to the active state, the Host MCU should read and check REG_CPURESET is zero to make sure FT800 is ready, then it can upload capacitive touch firmware to the FT800.

Listing 7.2 Upload capacitive touch firmware to FT800

```
            EVE_Hal_wr8(phost, REG_CPURESET, 1);
            EVE_Hal_wr8(phost, REG_BIST_EN, 1);
            EVE_Hal_wr8(phost, REG_ROMSUB_SEL, 1);
            uploadTouchFirmware(phost);
            EVE_Hal_wr8(phost, REG_BIST_EN, 0);
            EVE_Hal_wr8(phost, REG_CPURESET, 0);
            EVE_Hal_flush(phost);
            EVE_sleep(100);

static inline void uploadTouchFirmware(EVE_HalContext *phost)
    {
          EVE_Hal_wrMem(phost, RAM_ROMSUB, c_TouchDataU8, TOUCH_DATA_LEN);
          eve_assert_do(EVE_Cmd_waitFlush(phost));
    }

#define TOUCH_DATA_LEN 1544

static uint8_t c_TouchDataU8[TOUCH_DATA_LEN] =
{
201, 66, 0, 128, 140, 102, 1, 128, 143, 98, 2, 128, 143, 98, 39, 97, 16, 97, 29,
107, 156, 97, 16, 97, 39, 97, 16, 97, 29, 107, 140, 96, 0, 128, 143, 103, 16,
97, 157, 97, 17, 97, 157, 97, 3, 97, 143, 97, 39, 97, 39, 97, 20, 64, 29, 107,
29, 107, 7, 64, 39, 97, 7, 64, 29, 107, 140, 96, 29, 64, 22, 0, 39, 97, 140, 96,
39, 97, 17, 97, 29, 107, 156, 97, 3, 103, 140, 102, 16, 64, 140, 102, 0, 128,
143, 104, 46, 64, 140, 102, 0, 128, 16, 97, 143, 104, 3, 104, 140, 102, 51, 64,
140, 102, 16, 97, 143, 111, 0, 106, 140, 102, 59, 64, 143, 98, 17, 96, 46, 64,
67, 32, 59, 0, 140, 96, 20, 64, 3, 104, 73, 32, 3, 97, 74, 0, 143, 96, 140, 96,
20, 64, 51, 64, 70, 0, 17, 96, 143, 98, 1, 128, 143, 105, 17, 96, 0, 108, 16,
97, 3, 64, 140, 108, 10, 128, 51, 96, 143, 97, 7, 64, 87, 64, 51, 96, 143, 97,
17, 96, 46, 0, 1, 128, 0, 128, 7, 64, 3, 98, 39, 97, 17, 97, 3, 98, 17, 96, 7,
64, 3, 111, 110, 32, 29, 107, 3, 64, 111, 0, 29, 107, 140, 96, 0, 102, 16, 97,
0, 102, 16, 97, 96, 0, 17, 96, 46, 64, 121, 32, 112, 0, 140, 96, 39, 97, 0, 128,
0, 128, 29, 107, 17, 96, 141, 32, 36, 96, 1, 128, 3, 99, 134, 32, 24, 64, 98,
64, 29, 64, 20, 64, 98, 64, 29, 64, 29, 107, 80, 64, 126, 0, 3, 97, 34, 0, 1,
128, 16, 97, 51, 96, 143, 97, 0, 128, 16, 97, 51, 96, 143, 97, 0, 128, 143, 64,
0, 96, 0, 96, 0, 96, 0, 96, 0, 96, 0, 128, 147, 64, 0, 96, 0, 96, 0, 96, 140,
96, 0, 128, 17, 101, 169, 32, 3, 64, 165, 0, 22, 0, 1, 128, 143, 64, 9, 128, 0,
108, 3, 128, 3, 99, 3, 128, 3, 98, 151, 64, 0, 106, 0, 128, 0, 103, 178, 32, 3,
97, 1, 128, 147, 64, 151, 64, 151, 64, 151, 64, 151, 64, 151, 64, 151, 64, 151,
64, 151, 64, 151, 64, 151, 64, 3, 128, 140, 108, 20, 64, 19, 128, 82, 64, 7, 64,
122, 64, 39, 97, 39, 97, 17, 128, 82, 64, 7, 64, 122, 64, 29, 107, 29, 107, 98,
64, 21, 128, 82, 64, 98, 64, 3, 96, 11, 64, 25, 128, 82, 64, 7, 64, 122, 64, 39,
97, 39, 97, 23, 128, 82, 64, 7, 64, 122, 64, 29, 107, 29, 107, 98, 64, 27, 128,
82, 64, 98, 64, 143, 96, 120, 131, 4, 128, 51, 96, 143, 97, 186, 131, 4, 128,51,
96, 143, 97, 184, 131, 4, 128, 51, 96, 143, 97, 189, 131, 4, 128, 51, 96, 143,
97, 188, 131, 4, 128, 51, 96, 143, 97, 185, 131, 4, 128, 51, 96, 143, 97, 184,
131, 4, 128, 51, 96, 143, 97, 248, 131, 4, 128, 51, 96, 143, 97, 4, 128, 0, 106,
17, 96, 16, 64, 11, 33, 143, 97, 10, 65, 10, 1, 144, 129, 16, 65, 16, 65, 0,
106, 17, 96, 16, 64, 19, 33, 3, 97, 0, 106, 17, 96, 16, 64, 18, 33, 143, 97, 2,
65, 16, 65, 250, 64, 16, 65, 246, 0, 254, 64, 16, 65, 2, 65, 16, 65, 250, 64,
16, 65, 246, 64, 16, 1, 254, 64, 16, 65, 246, 64, 16, 65, 250, 64, 16, 65, 2, 1,
170, 64, 9, 128, 143, 105, 254, 64, 16, 65, 6, 65, 51, 65, 57, 33, 16, 65, 2,
65, 51, 65, 254, 64, 16, 1, 72, 33, 254, 64, 10, 65, 2, 65, 16, 65, 254, 64, 10,
```

```
65, 78, 1, 246, 64, 10, 65, 250, 64, 16, 65, 246, 64, 10, 1, 140, 96, 17, 96,
128, 128, 3, 99, 64, 65, 78, 0, 254, 64, 10, 65, 79, 65, 79, 65, 79, 65, 79, 65,
79, 65, 79, 65, 79, 65, 79, 65, 3, 97, 54, 1, 78, 64, 54, 65, 143, 98, 0, 128,
96, 65, 96, 65, 96, 65, 96, 65, 96, 65, 96, 65, 96, 65, 96, 65, 96, 65, 16, 97, 64, 1,
113, 33, 0, 128, 36, 0, 140, 96, 44, 129, 18, 65, 0, 228, 31, 128, 51, 96, 3,
97, 44, 129, 18, 65, 17, 96, 31, 128, 51, 96, 3, 97, 114, 1, 140, 96, 0, 128,
99, 1, 128, 65, 8, 128, 3, 113, 128, 65, 143, 100, 255, 255, 0, 102, 157, 96,
135, 65, 20, 64, 20, 64, 20, 64, 20, 0, 31, 128, 51, 96, 3, 97, 35, 128, 51, 96,
3, 97, 34, 128, 90, 64, 33, 128, 90, 64, 30, 128, 90, 64, 32, 128, 90, 0, 31,
65, 112, 128, 84, 65, 110, 65, 84, 65, 110, 1, 1, 64, 99, 65, 3, 97, 44, 1, 36,
96, 0, 128, 3, 103, 174, 33, 32, 128, 90, 64, 207, 1, 17, 107, 1, 128, 3, 103,
181, 33, 30, 128, 90, 64, 207, 1, 17, 107, 2, 128, 3, 103, 188, 33, 33, 128, 90,
64, 207, 1, 17, 107, 3, 128, 3, 103, 195, 33, 34, 128, 90, 64, 207, 1, 17, 107,
4, 128, 3, 103, 206, 33, 31, 128, 51, 96, 3, 97, 35, 128, 51, 96, 3, 97, 207, 1,
22, 64, 12, 96, 140, 96, 17, 96, 5, 128, 3, 111, 218, 33, 1, 128, 16, 97, 3,
113, 3, 100, 219, 1, 143, 97, 140, 96, 39, 97, 17, 96, 229, 33, 16, 97, 17, 107,
36, 64, 16, 97, 0, 106, 221, 1, 3, 97, 12, 96, 163, 65, 17, 96, 0, 128, 17, 96,
5, 128, 3, 104, 251, 33, 39, 97, 17, 96, 1, 128, 3, 99, 16, 64, 247, 33, 135,
65, 17, 107, 167, 65, 80, 64, 29, 107, 3, 64, 234, 1, 22, 0, 130, 65, 255, 143,
3, 99, 130, 65, 17, 96, 255, 143, 3, 99, 16, 97, 12, 128, 3, 105, 15, 128, 3,
99, 39, 97, 130, 65, 3, 97, 198, 64, 17, 107, 167, 65, 29, 107, 209, 1, 1, 128,
157, 65, 44, 65, 31, 65, 113, 128, 84, 65, 110, 65, 128, 65, 3, 97, 0, 128, 128,
65, 252, 129, 220, 65, 143, 97, 1, 64, 1, 64, 255, 255, 0, 102, 17, 96, 255,
255, 140, 96, 31, 128, 51, 96, 3, 97, 32, 128, 90, 64, 20, 64, 30, 128, 90, 64,
33, 128, 90, 0, 2, 128, 157, 65, 44, 65, 31, 65, 113, 128, 84, 65, 110, 65, 128,
65, 66, 34, 130, 65, 255, 143, 3, 99, 130, 65, 255, 143, 3, 99, 20, 64, 198, 64,
200, 128, 67, 2, 30, 66, 163, 65, 37, 2, 1, 128, 37, 128, 51, 96, 143, 97, 234,
64, 16, 1, 11, 128, 0, 108, 0, 128, 3, 103, 143, 100, 73, 66, 17, 96, 36, 64,
75, 66, 110, 65, 51, 65, 16, 64, 81, 34, 143, 97, 73, 66, 17, 96, 36, 64, 75,
66, 110, 65, 51, 65, 90, 34, 143, 97, 13, 128, 0, 108, 0, 128, 143, 103, 97, 66,
140, 102, 101, 130, 80, 66, 16, 66, 69, 66, 101, 130, 89, 66, 103, 2, 140, 96,
0, 128, 39, 97, 97, 130, 80, 66, 47, 66, 69, 66, 97, 130, 89, 66, 113, 2, 140,
96, 31, 65, 232, 128, 84, 65, 110, 65, 84, 65, 110, 1, 128, 65, 0, 106, 39, 97,
130, 65, 130, 65, 130, 65, 3, 97, 198, 64, 17, 107, 167, 65, 29, 107, 209, 1,
31, 65, 233, 128, 84, 65, 110, 65, 0, 128, 128, 65, 7, 128, 3, 99, 127, 130,
220, 1, 31, 65, 233, 128, 84, 65, 110, 65, 128, 65, 7, 128, 3, 99, 169, 34, 128,
65, 128, 128, 3, 99, 163, 34, 30, 66, 168, 2, 130, 65, 130, 65, 20, 64, 198, 64,
200, 128, 170, 2, 30, 66, 163, 65, 37, 2, 97, 130, 80, 66, 97, 130, 89, 66, 149,
66, 69, 66, 172, 2, 140, 96, 0, 128, 101, 130, 80, 66, 101, 130, 89, 66, 139,
66, 18, 64, 3, 100, 190, 34, 69, 66, 181, 2, 140, 96, 73, 66, 0, 128, 3, 64, 32,
206, 3, 103, 51, 65, 3, 100, 194, 34, 143, 97, 242, 64, 11, 128, 0, 108, 0, 128,
42, 64, 202, 34, 101, 66, 17, 96, 213, 34, 30, 66, 37, 66, 215, 2, 138, 65, 143,
65, 69, 66, 238, 64, 9, 128, 0, 108, 4, 128, 3, 99, 16, 64, 228, 34, 5, 128, 18,
65, 242, 64, 44, 129, 18, 65, 101, 66, 42, 64, 110, 65, 1, 128, 18, 65, 31, 65,
112, 128, 84, 65, 44, 65, 16, 64, 244, 34, 101, 66, 243, 34, 111, 66, 244, 2,
103, 66, 192, 66, 31, 65, 232, 128, 84, 65, 44, 65, 16, 64, 0, 35, 101, 66, 255,
34, 172, 66, 0, 3, 180, 66, 10, 128, 18, 65, 1, 64, 110, 1
};
```

8 Schematic Diagram

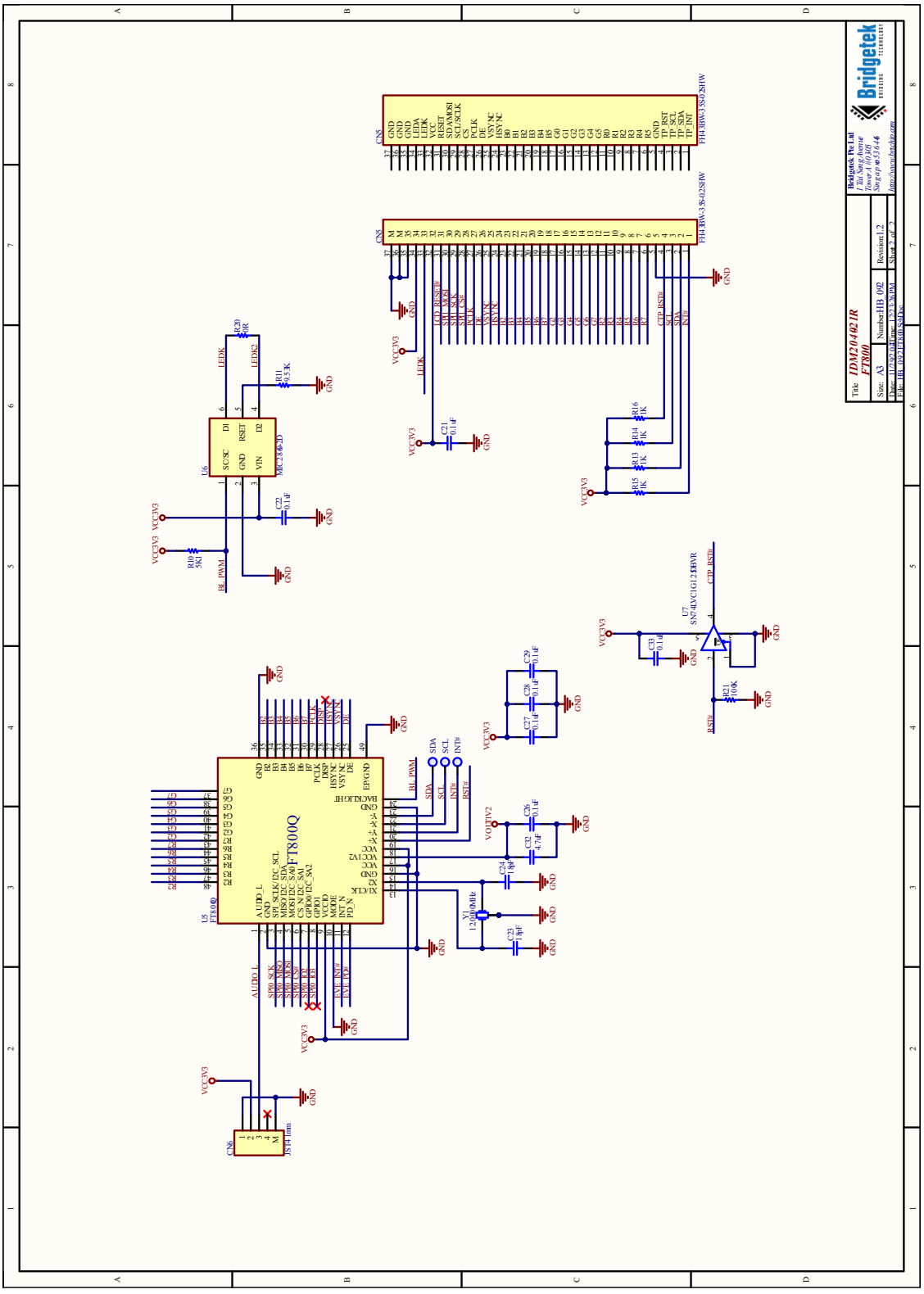

Figure 11 - RP2040 Schematic

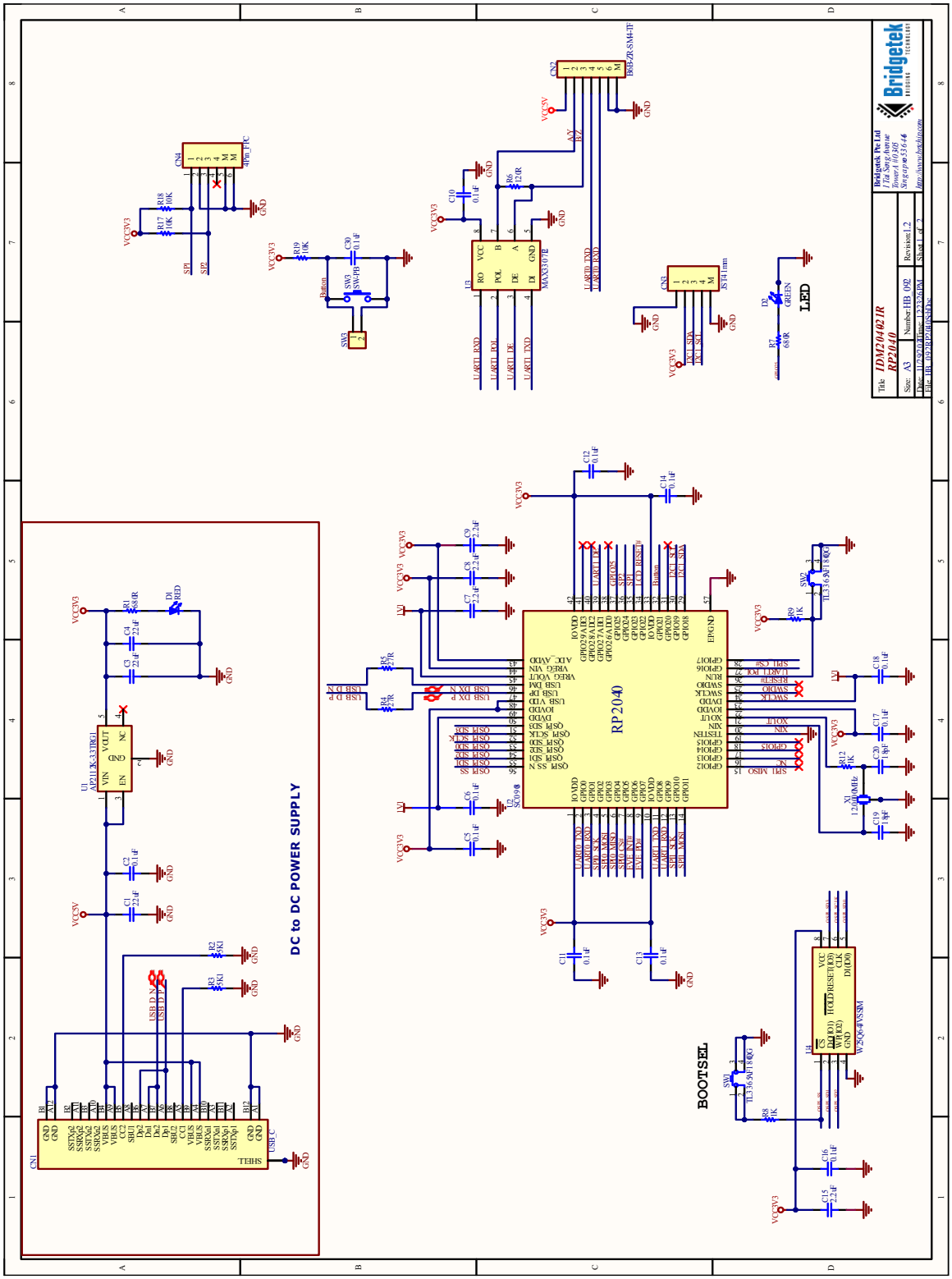

Figure 12 - FT800 Schematic

9 Hardware Parameters

Display Parameters	Parameters	Data	Description
	Display Size	Φ53.28mm	-
	Resolution	480× 480	-
	View Angle	IPS	Typical 80°/80°/80°/80° (L/R/U/D)
	Brightness	300cd/m²	-

Table 10 - Display parameters

Parameters		Test Condition		Min	Typical	Max	Unit
Electrical Parameters	Working Voltage	-	-	-	5.0	-	V
	Working Current	VCC =+5 V	Backlight, Full Brightness	-	237	-	mA
			Backlight Off	-	63	-	mA
Reliability	Working Temperature	5V, Humidity 60%		-20	25	70	**℃**
	Storage Temperature	-		-30	25	80	**℃**
	Working Humidity	-		10%	60%	90%	RH
Interface	RS-485 Baud Rate	-		1200	115200	-	bps
	USB Port	Type C					

Table 11 - Other Parameters

10 Mechanical Information

10.1 Mechanical Drawing

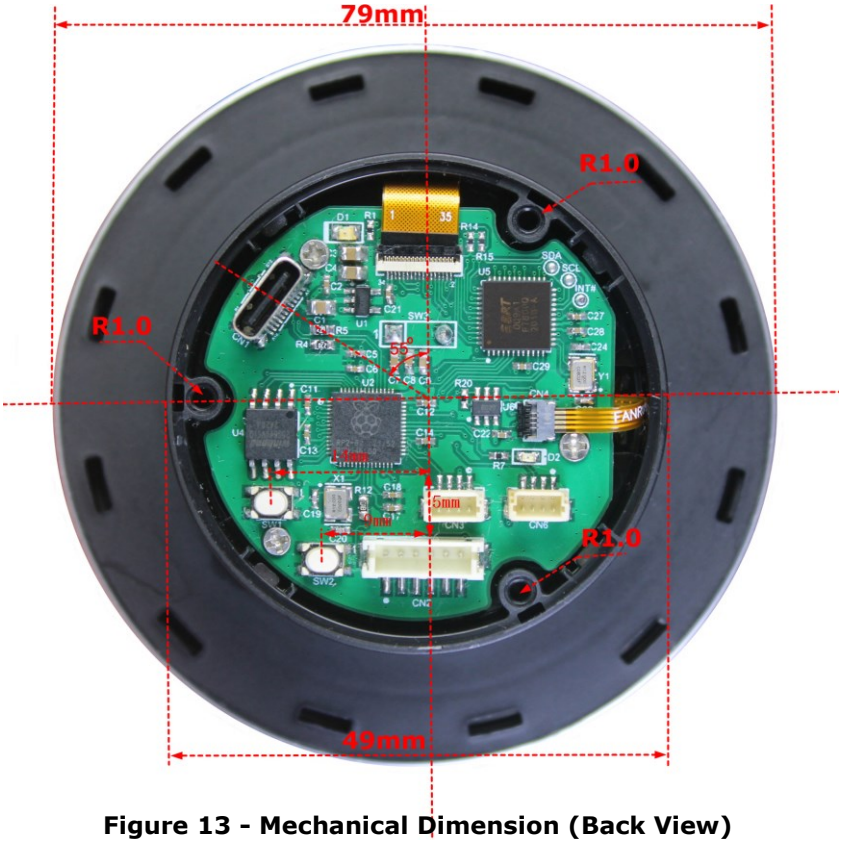

Figure 13 - Mechanical Dimension (Back View)

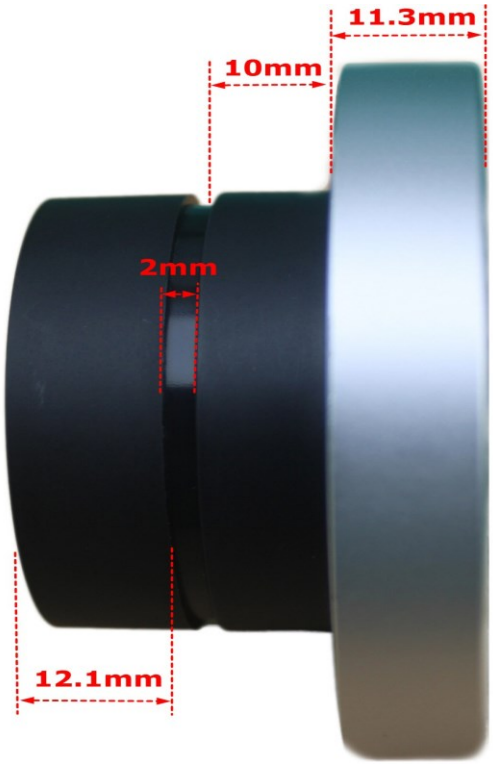

Figure 14 - Mechanical Dimension (Side View)

10.2 PCB Dimension

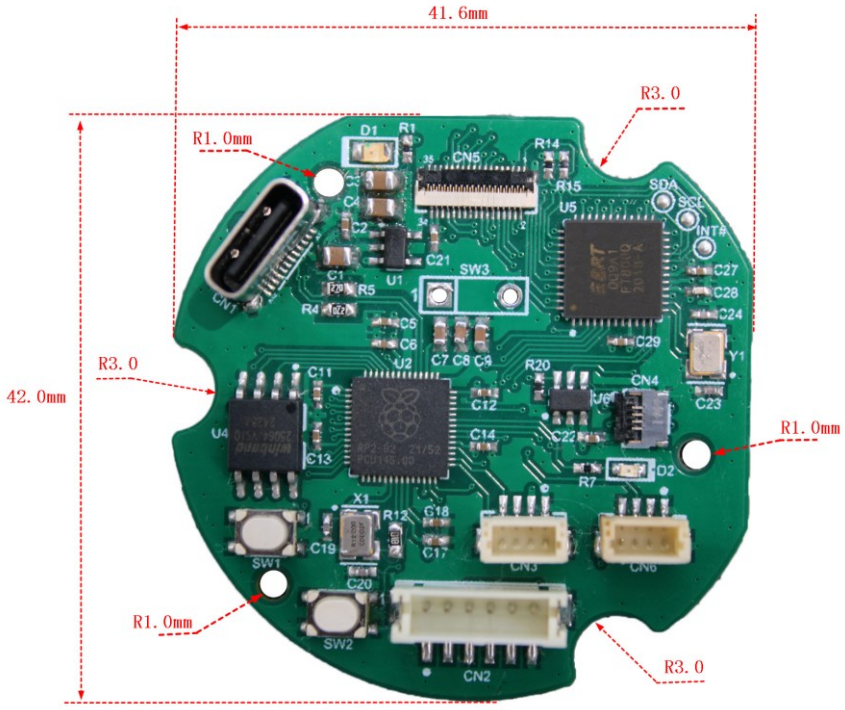

Figure 15 - PCB Dimension

10.3 Packing List

No.	Item	Quantity	Description
1	IDM2040-21R	1	2.1-inch Rotary Dial Display
2	JST ZH1.5 to Dupont 2.54 female 6 pin cable	1	JST ZH1.5mm pitch 6 pin to Dupont 2.54mm Female 6 pin jumper wire. The length of each cable is approx 20 cm/7.87 inches.

11 Disclaimer Notice: Use of Third-Party Software or Websites

We may recommend use of software or web sites that are owned or operated by other companies. We offer or facilitate this recommendation by hyperlinks or other methods to aid your access to the third-party resource. While we endeavour to direct you to helpful, trustworthy resources, we cannot endorse, approve, or guarantee software, information, products, or services provided by or at a third-party resource or track changes in the resource. Thus, we are not responsible for the content or accuracy of any third-party resource or for any loss or damage of any sort resulting from the use of, or for any failure of, products or services provided at or from a third-party resource.

We recommend these resources on an "as is" basis. When you use a third-party resource, you will be subject to its terms and licenses and no longer be protected by our privacy policy or security practices, which may differ from the third part policy or practices or other terms. You should familiarize yourself with any license or use terms of, and the privacy policy and security practices of, the third-party resource, which will govern your use of that resource

12 Contact Information

Refer to https://brtchip.com/contact-us/ for contact information.

Distributor and Sales Representatives

Please visit the Distribution Network – IC & Module (brtchip.com) page for the contact details of our distributor(s) and sales representative(s) in your country.

Appendix A – References

Document References

FT800 Embedded Video Engine Datasheet

FT800 Series Programmer Guide

EVE Toolchains

RP2040 Datasheet

Acronyms and Abbreviations

Terms	Description
EVE	Embedded Video Engine
GPIO	General Purpose Input Output
I2C	Inter-Integrated Circuit
LCD	Liquid-Crystal Display
LED	Light Emitting Diode
MCU	Microcontroller Unit
PCB	Printed Circuit Board
PWM	Pulse-Width Modulation
QSPI	Quad Serial Peripheral Interface
RGB	Red, Green, Blue
SPI	Serial Peripheral Interface
SRAM	Static random-access memory
TFT	Thin Film Transistor
UART	Universal Asynchronous Receiver / Transmitter
USB	Universal Serial Bus

Appendix B – List of Figures and Tables

List of Figures

List of Tables

Appendix C – Revision History

Document Title: IDM2040-21R Datasheet
Document Reference No.: BRT_000444
Clearance No.: BRT#217
Product Page: https://brtchip.com/product-category/products/
Document Feedback: Send Feedback

Revision	Changes	Date
1.0	Initial Release	15-01-2025