**E-INK display interfacing**

**Introduction**

E-paper display is based on **Microencapsulated Electrophoretic Display** (MED) technology which is one of the famous image display technology. In this technology, the first step is to make microscopic spheres in which charged color pigments are suspended in clear oil and move in response to the electronic charge.

The data communication of this device is using 3 wire, 4 wire SPI Serial Peripheral Interface that runs on 3.3v and generates 4 color, black, white, grey, and red.

|  |  |
| --- | --- |
| Pin | Description |
| VCC | 3.3V/5V |
| GND | Ground |
| DIN | SPI MOSI pin |
| CLK | SPI SCK pin |
| CS | SPI chip selection, low active |
| CS | Data/Command selection (high for data, low for command) |
| RST | External reset, low active |
| BUSY | Busy status output, high active |



Component

* Jumper Wires / DuPont Wires
* E Ink 172×72 Gray Shade Electronic Ink Module
* Arduino nano board

Application

shelf label, industrial instruments, home and appliances, automotive, mobile devices such as E-Paper, E-Book, E-Reader and Smart Watch.

Objective

During this activity ,you will help students to achieve following objectives

1. Understanding the principle and operation of 1.5 inch E –ink display module
2. Design algorithm and flowchart to interface e-ink display and show HELLO WORLD data
3. Programming 1.5 inch E –ink display module with arduino nano
4. Interfacing 1.5 inch E –ink display module with arduino nano

**Program**

#include <SPI.h>

#include <epd1in54.h>

#include <epdpaint.h>

#include "imagedata.h" // Image Data

#define COLORED 0

#define UNCOLORED 1

unsigned char image[1024];

Paint paint(image, 0, 0); // width should be the multiple of 8

Epd epd;

unsigned long time\_start\_ms;

unsigned long time\_now\_s;

void setup() {

Serial.begin(9600);

if (epd.Init(lut\_full\_update) != 0) {

Serial.print("14CORE | E-Paper Initialization Failed ");

delay(1000);

Serial.print("Check wiring ..........................")

return;

}

/\*\*

\* there are 2 memory areas embedded in the e-paper display

\* and once the display is refreshed, the memory area will be auto-toggled,

\* i.e. the next action of SetFrameMemory will set the other memory area

\* therefore you have to clear the frame memory twice.

\*/

epd.ClearFrameMemory(0xFF); // bit set = white, bit reset = black

epd.DisplayFrame();

epd.ClearFrameMemory(0xFF); // bit set = white, bit reset = black

epd.DisplayFrame();

paint.SetRotate(ROTATE\_0);

/\* ------- Change this if your using other sizes of EPaper --------\*/

//paint.SetWidth(129); // Uncommetn if your using 2.9 E-Ink Display

//paint.SetWidth(24); // Uncommetn if your using 2.9 E-Ink Display

paint.SetWidth(200); //Comment if your using 2.9 inch

paint.SetHeight(24); //COmment if your using 2.9 inch

/\* ----- Numerical Coordinates > 2.9 Inch Un-comment -------- \*/

// paint.Clear(COLORED);

// paint.DrawStringAt(0, 4, "14CORE | TEST CODE", &Font16, UNCOLORED);

// epd.SetFrameMemory(paint.GetImage(), 0, 10, paint.GetWidth(), paint.GetHeight());

//paint.Clear(UNCOLORED);paint.DrawStringAt(0, 4, "EPAPER 2.9 Inch DEMO", &Font16, COLORED);

//epd.SetFr

//ameMemory(paint.GetImage(), 0, 30, paint.GetWidth(), paint.GetHeight());

/\* ------------------------------------------------------------- \*/

/\* -------------- Numerical Coordinates > 1.54 Inch ------------ \*/

paint.Clear(COLORED);

paint.DrawStringAt(30, 4, "14CORE | TEST CODE", &Font16, UNCOLORED);

epd.SetFrameMemory(paint.GetImage(), 0, 10, paint.GetWidth(), paint.GetHeight());

paint.Clear(UNCOLORED);

paint.DrawStringAt(30, 4, "EPAPER 1.5 Inch DEMO", &Font16, COLORED);

epd.SetFrameMemory(paint.GetImage(), 0, 30, paint.GetWidth(), paint.GetHeight());

/\* ---------------------------------------------------------- \*/

paint.SetWidth(64);

paint.SetHeight(64);

paint.Clear(UNCOLORED);

paint.DrawRectangle(0, 0, 40, 50, COLORED);

paint.DrawLine(0, 0, 40, 50, COLORED);

paint.DrawLine(40, 0, 0, 50, COLORED);

epd.SetFrameMemory(paint.GetImage(), 16, 60, paint.GetWidth(), paint.GetHeight());

paint.Clear(UNCOLORED);

paint.DrawCircle(32, 32, 30, COLORED);

epd.SetFrameMemory(paint.GetImage(), 120, 60, paint.GetWidth(), paint.GetHeight());

paint.Clear(UNCOLORED);

paint.DrawFilledRectangle(0, 0, 40, 50, COLORED);

epd.SetFrameMemory(paint.GetImage(), 16, 130, paint.GetWidth(), paint.GetHeight());

paint.Clear(UNCOLORED);

paint.DrawFilledCircle(32, 32, 30, COLORED);

epd.SetFrameMemory(paint.GetImage(), 120, 130, paint.GetWidth(), paint.GetHeight());

epd.DisplayFrame();

delay(2000);

if (epd.Init(lut\_partial\_update) != 0) {

Serial.print("e-Paper init failed");

return;

}

/\*\*

\* there are 2 memory areas embedded in the e-paper display and once the display is refreshed, the memory area will be auto-toggled,

i.e. the next action of SetFrameMemory will set the other memory area therefore you have to set the frame memory and refresh the

display twice.

\*/

epd.SetFrameMemory(IMAGE\_DATA);

epd.DisplayFrame();

epd.SetFrameMemory(IMAGE\_DATA);

epd.DisplayFrame();

time\_start\_ms = millis();

}

void loop() {

// put your main code here, to run repeatedly:

time\_now\_s = (millis() - time\_start\_ms) / 1000;

char time\_string[] = {'0', '0', ':', '0', '0', '\0'};

time\_string[0] = time\_now\_s / 60 / 10 + '0';

time\_string[1] = time\_now\_s / 60 % 10 + '0';

time\_string[3] = time\_now\_s % 60 / 10 + '0';

time\_string[4] = time\_now\_s % 60 % 10 + '0';

paint.SetWidth(32);

paint.SetHeight(96);

paint.SetRotate(ROTATE\_270);

paint.Clear(UNCOLORED);

paint.DrawStringAt(0, 4, time\_string, &Font24, COLORED);

epd.SetFrameMemory(paint.GetImage(), 80, 72, paint.GetWidth(), paint.GetHeight());

epd.DisplayFrame();

delay(500);

}

**Hardware**

Interfacing of E-paper Display with Arduino UNO

Now, we are going to interface a 1.5-inch E-Paper display with Arduino nano. The E-Paper display offers a resolution of 200x200pixels which is great and uses the SPI interface. E-Paper display supports a 3.3V display so Vcc must be connected to the 3.3V output of the Arduino Uno. The other pins of the display are 5V tolerant. The second pin is GND and it connects to the GND pin of Arduino . The third pin is named SDA and it connects to Digital Pin 11. The fourth pin is CLK and it connects to Digital Pin 13. The fifth pin (CS) connects to digital pin D6, the 6th pin (DC) connects to digital D5, the 7th pin (RST) connects to digital pin RSTand the last pin connects to digital pin 7.

