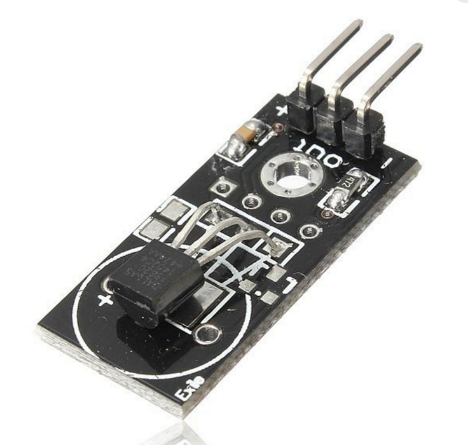
**E paper thermometer**

**Introduction**

In this project we are displaying temperature data on E paper display using digital thermometer.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 DS18B20 is a digital thermometer that accurately measures temperature in the range -10°C to +85°C and also includes alarm functions and trigger points.

**Components**

The parts needed are the following:

* An ESP32 Board
* A 1.5 Inch E-Paper Display
* A DS18B20 Sensor
* A breadboard
* Some wires

**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 temperature 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 <GxEPD.h>

#include <GxGDEP015OC1/GxGDEP015OC1.h>

#include <GxIO/GxIO\_SPI/GxIO\_SPI.h>

#include <GxIO/GxIO.h>

#include <OneWire.h>

#include <DallasTemperature.h>

#include "BitmapGraphics.h"

#include <Fonts/FreeSansBold24pt7b.h>

#define ONE\_WIRE\_BUS 15

boolean METRIC = true; //Set true for metric system; false for imperial

float previousTemperature = -100.0;

float temperature = 0;

OneWire oneWire(ONE\_WIRE\_BUS);

DallasTemperature sensors(&oneWire);

GxIO\_Class io(SPI, SS, 22, 21);

GxEPD\_Class display(io, 16, 4);

void setup() {

Serial.begin(9600);

display.init();

sensors.begin();

display.drawExampleBitmap(gImage\_splash, 0, 0, 200, 200, GxEPD\_BLACK);

display.update();

delay(3000);

if(METRIC)

{

display.drawExampleBitmap(gImage\_gui, 0, 0, 200, 200, GxEPD\_BLACK);

}else

{

display.drawExampleBitmap(gImage\_gui\_f, 0, 0, 200, 200, GxEPD\_BLACK);

}

display.update();

if(METRIC)

{

display.drawExampleBitmap(gImage\_gui, sizeof(gImage\_gui), GxEPD::bm\_default | GxEPD::bm\_partial\_update);

}else

{

display.drawExampleBitmap(gImage\_gui\_f, sizeof(gImage\_gui\_f), GxEPD::bm\_default | GxEPD::bm\_partial\_update);

}

}

void loop()

{

previousTemperature = temperature;

temperature = getTemperatureFromSensor();

if(previousTemperature!=temperature)

{

showPartialUpdate(temperature);

}

printTemperatureToSerial();

delay(5000);

}

float getTemperatureFromSensor()

{

float temperature = 0;

sensors.requestTemperatures();

if(METRIC)

{

temperature = sensors.getTempCByIndex(0);

}else

{

temperature = sensors.getTempFByIndex(0);

}

temperature = round(temperature \* 10) / 10;

return temperature;

}

void showPartialUpdate(float temperature)

{

Serial.println("Updating display ...");

String temperatureString = String(temperature,1);

const char\* name = "FreeSansBold24pt7b";

const GFXfont\* f = &FreeSansBold24pt7b;

uint16\_t box\_x = 60;

uint16\_t box\_y = 60;

uint16\_t box\_w = 90;

uint16\_t box\_h = 100;

uint16\_t cursor\_y = box\_y + 16;

display.setRotation(45);

display.setFont(f);

display.setTextColor(GxEPD\_BLACK);

display.fillRect(box\_x, box\_y, box\_w, box\_h, GxEPD\_WHITE);

display.setCursor(box\_x, cursor\_y+38);

display.print(temperatureString);

display.updateWindow(box\_x, box\_y, box\_w, box\_h, true);

}

void printTemperatureToSerial()

{

Serial.print(temperature);

Serial.print(" C");

Serial.print("\n");

}

Hardware

1. Connect digital temoerature sensor VCC and ground connection to board and signal pin to D15
2. Connect E-Paper display to ESP-01 module and do following connection.

VCC 🡪VCC

GND 🡪GND

DIN🡪 D13

CLK🡪 D10

CS 🡪 D5

RST🡪D11

BUSY🡪D4

