**E-ink based clock**

**Introduction**

In this project we made clock display using E-ink display and arduino nano.Arduinos Nano: one for the Real-Time Clock that will send the data to the main Arduino via Serial. And the main Arduino will print it in the e-ink display.

**Component**

* Arduino nano-2
* DS3231 RTC module
* 4.2 inch E-ink display module

**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 4.2 inch E –ink display module
2. Design algorithm and flowchart to get time and date information
3. Programming 4.2 inch E –ink display module with arduino nano
4. Interfacing 4.2 inch E –ink display module with arduino nano

**Program**

**Code for RTC**

#include <Wire.h>

#include "RTClib.h"

RTC\_DS3231 rtc;

// I'm going to declare two arrays with the name of the days and the names of the months.

// As I should clean the litters, at least twice a week

// I'm not interested into getting the day of the month number instead of the name of the day in the week

String daysOfTheWeek[7] = { "SUNDAY", "MONDAY", "TUESDAY", "WEDNESDAY", "THURSDAY", "FRIDAY", "SATURDAY" };

String monthsNames[12] = {"FEBRUARY", "MARCH", "APRIL", "MAY", "JUNE", "JULY","AUGUST","SEPTEMBER","OCTOBER","NOVEMBER","DECEMBER","JANUARY" };

int monthNumber = 1;

String currentDate = "";

String currentDay = "";

**void** setup() {

delay(4000); // I'm giving it some time before it does starts sending things to the main Arduino

Serial.begin(115200); // I've tried with bigger baud rates, and it eventually worked 1 of every 20 times

**if** (!rtc.begin()) {

**while** (1);

}

**if** (rtc.lostPower()) {

rtc.adjust(DateTime(F(\_\_DATE\_\_), F(\_\_TIME\_\_)));

}

}

**void** printDate(DateTime date)

{

monthNumber = (date.month(), DEC) - 1;

currentDay = (date.day(), DEC);

currentDate = currentDay + " " + (monthsNames[monthNumber]); //I create a string with the current day + the month name

Serial.print(daysOfTheWeek[date.dayOfTheWeek()]);

Serial.print("\n");// I also add this line break charachter for the screen

Serial.print(date.day(), DEC);

Serial.print(" ");

Serial.println(monthsNames[monthnumber]);

}

**void** loop() {

// Get the current date and send it via Serial

DateTime now = rtc.now();

printDate(now);

delay(2000);

}

**Code for main Arduino:**

#include <string.h>

#include <GxEPD.h>

#include <GxGDEH029A1/GxGDEH029A1.h> // 2.9" b/w

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

#include <GxIO/GxIO.h>

#include <Fonts/FreeMonoBold12pt7b.h>

#include <Fonts/FreeMonoBold24pt7b.h>

// constructor for AVR Arduino, copy from GxEPD\_Example else

GxIO\_Class **io**(SPI, /\*CS=\*/ SS, /\*DC=\*/ 8, /\*RST=\*/ 9); // arbitrary selection of 8, 9 selected for default of GxEPD\_Class

GxEPD\_Class **display**(io, /\*RST=\*/ 9, /\*BUSY=\*/ 7); // default selection of (9), 7

String currentDate;

**void** **setup**()

{

pinMode(3, OUTPUT);

digitalWrite(3, LOW); //I turn the realy pin on, so now there's no need to keep the push button pressed

Serial.begin(115200);

delay(2000);

display.init();

display.eraseDisplay(); //This works fine beacuse the screen flashes.

pinMode(4, OUTPUT);

digitalWrite(4, HIGH); //With this instruction I turn the second Arduino on.

}

**void** **drawHelloWorld**()

{

display.fillScreen(GxEPD\_WHITE);

display.setTextColor(GxEPD\_BLACK);

display.setFont(&FreeMonoBold12pt7b);

display.setRotation(15);

display.setCursor(0, 15);

display.print("LITTERS CLEANED:");

display.setCursor(0, 70);

display.setFont(&FreeMonoBold24pt7b);

display.print(currentDate);

delay(10000); // I give it some time to update the e-ink display

digitalWrite(3, HIGH); //Finally I turn down the whole system desactivating the relay

}

**void** **drawHelloWorld0**() // If I call this function as a welcome screen, it always works

{

display.fillScreen(GxEPD\_BLACK);

display.setTextColor(GxEPD\_WHITE);

display.setFont(&FreeMonoBold24pt7b);

display.setRotation(15);

display.setCursor(0, 25);

display.print(" \_/)\n (^.^) MIU\n MIU");

display.setCursor(37, 25);

display.print("/)");

display.setCursor(58, 100);

display.print(">^<");

display.setCursor(15, 120);

display.print("( )");

}

**void** **loop**() {

**while** (Serial.available()) {

currentDate= Serial.readString();

delay(3);

**if** (currentDate.length() > 5) { //just a little checking to see if the is longer that a simple char

display.drawPaged(drawHelloWorld); // version for AVR using paged drawing, works also on other processors

}

}

}

**Hardware**

1. **Connect relay and switch VCC and GND as per power supply pin of Arduino board and D3 pin from Arduino is coonected to realy In pin**
2. **Connect e-ink display such as**

**CLK to D13**

**CS-D11**

**DC-D8**

**RST-D9**

**BUSY-D7**

1. **RTC module connection with board are as**

**SDA-A4**

**SCL-A5**

