M5Paper 環境モニターソース

美都

2021 年 3 月 14 日

目次

1 メイン

```
#include <M5EPD.h>
#define LGFX_M5PAPER
#include <LovyanGFX.hpp>

#include "battery.h"
#include "thermometer.hpp"
#include "infoFromNet.hpp"
#include "tokei.hpp"

static LGFX lcd;

void drawLcd() {
    int bat = drawBattery(960-120-5, 5, &lcd);
    Thermometer t = Thermometer(200,200);
    t.drawTempMeter(500, 100, &lcd);
    t.drawHumMeter(700, 100, &lcd);

    Tokei tokei = Tokei(300, 100);
    tokei.drawDigitalTokei(&lcd, 100, 100);

    delay(1000);
}

// ●分ピッタリまでの秒数
int rest_minute() {
    rtc_time_t time;
    M5.RTC.getTime(&time);
    return 60-time.sec;
}

// シャットダウンを試みる。通電中はすり抜ける
void challengeShutdown() {
        int rest_sec = rest_minute()-6;
        if (rest_sec < 30) rest_sec += 60;
        M5.shutdown(rest_sec); // 一旦停止
}

void setup()
{
    M5.begin(false, true, true, true, true);
    M5.BatteryADCBegin();
    M5.RTC.begin();
    M5.SHT30.Begin();
    lcd.init();
    lcd.setRotation(1);

    rtc_time_t time;
    rtc_date_t date;
    M5.RTC.getDate(&date);
    M5.RTC.getTime(&time);
    if ((time.hour%6==0 && time.min<1) || date.year<2020) {
        GetInfoFromNetwork info;
        info.setNtpTime();
    }
```

```
55
56      drawLcd();
57      challengeShutdown();
58  }
59
60
61  void loop()
62  {
63      delay((rest_minute()+1)*1000);
64      drawLcd();
65      challengeShutdown();
66  }
```

2 バッテリーメーター

2.1 battery.h

```
#include <M5EPD.h>
#define LGFX_M5PAPER
#include <LovyanGFX.hpp>

// バッテリー残量を(x,y)に表示する。
int drawBattery(int x, int y, LGFX *lcd) ;
```

2.2 battery.cpp

```
#include "battery.h"

// バッテリー残量の取得
static int get_rest_battery() {
    const int max_vol = 4350;
    const int min_vol = 3300;
    //M5.BatteryADCBegin();
    int voltage = M5.getBatteryVoltage();
    voltage = max(voltage, min_vol);
    voltage = min(voltage, max_vol);
    float rest_battery_raw = (float)(voltage - min_vol) / (float)(max_vol - min_vol);
    rest_battery_raw = max(rest_battery_raw, 0.01f);
    rest_battery_raw = min(rest_battery_raw, 1.f);
    return (int)(rest_battery_raw * 100);
}

// バッテリー残量計の表示
int drawBattery(int x, int y, LGFX *lcd) {
    LGFX_Sprite battery_meter(lcd);
    int rest_battery = get_rest_battery();

    // バッテリー矩形の表示
    battery_meter.setColorDepth(4);
    battery_meter.createSprite(120, 30);
    battery_meter.fillSprite(15);
    battery_meter.setColor(0);
    battery_meter.drawRect(10, 10, 45, 20);
    battery_meter.fillRect(55, 17, 5, 5);
    battery_meter.fillRect(10, 10, (int)((45*rest_battery)/100), 20);

    // バッテリー残量文字の表示
    battery_meter.setFont(&fonts::lgfxJapanMinchoP_20);
    battery_meter.setTextSize(1, 1); // 縦,横　倍率
    battery_meter.setTextColor(0, 15); // 文字色,背景
    battery_meter.setCursor(62, 10);
    battery_meter.printf("%d%%", rest_battery);

    lcd->startWrite();
    battery_meter.pushSprite(x, y);
    lcd->endWrite();

    return rest_battery;
```

```
43 }
```

3 温湿度計

3.1 thermometer.hpp

```
#include <M5EPD.h>
#define LGFX_M5PAPER
#include <LovyanGFX.hpp>

class Thermometer {
    private:
        float temp;
        float hum;
        int sizex;
        int sizey;
        float radius;
        LGFX_Sprite face;
        LGFX_Sprite scale[2];
        LGFX_Sprite hand;

        void makeMeterFace(int min, int max, const char* unit);
        void makeScale();
        void makeHand();
    public:
        Thermometer(int sizex=200, int sizey=200);

        float get_temp();
        float get_hum();

        void drawTempMeter(int x, int y, LGFX *lcd);
        void drawHumMeter(int x, int y, LGFX *lcd);
        void drawString(int x, int y, LGFX *lcd);
};
```

3.2 thermometer.cpp

```
#include "thermometer.hpp"

Thermometer::Thermometer(int sizex, int sizey)
    : sizex(sizex), sizey(sizey) {
    M5.SHT30.Begin();
    radius = min(sizex, sizey)/2*0.95;
    makeScale();
    makeHand();
};

void Thermometer::makeScale() {
    scale[0].setColorDepth(4);
    scale[0].createSprite(radius/25, radius/5);
    scale[0].fillSprite(0);
    scale[0].setPivot(scale[0].width()/2, scale[0].height());
    scale[1].setColorDepth(4);
    scale[1].createSprite(radius/40, radius/7);
    scale[1].fillSprite(0);
    scale[1].setPivot(scale[1].width()/2, scale[1].height());
}
```

```
21
22  void Thermometer::makeHand() {
23      float height, width;
24      height = radius * 0.8f;
25      width = height * 0.1f;
26
27      hand.setColorDepth(4);
28      hand.createSprite(width, height);
29      hand.fillSprite(15);
30      hand.setColor(0);
31      hand.fillTriangle(width/2.f, 0, 0, height/4.f, width, height/4.f);
32      hand.fillTriangle(0, height/4.f, width, height/4.f, width/2.f, height);
33      hand.setPivot(width/2., height);
34  }
35
36  void Thermometer::makeMeterFace(int min, int max, const char* unit) {
37      face.setColorDepth(4);
38      face.createSprite(sizex, sizey);
39      face.fillSprite(15);
40      face.setColor(0);
41      face.setFont(&fonts::lgfxJapanGothic_36);
42      face.setTextColor(0, 15);
43      face.setTextDatum(middle_center);
44      float center[2] = {sizex/2.0f, sizey/2.0f};
45      face.fillCircle(center[0], center[1], radius);
46      face.fillCircle(center[0], center[1], radius*0.95, 15);
47      float angleInterval = 270.f / (float)(max-min);
48      for (int i = min ; i <= max ; i+=2) {
49          LGFX_Sprite *use_scale = (i%10==0) ? &scale[0] : &scale[1];
50          float angle = (270.f-45.f) - (float)(i-min) * angleInterval;
51          float angleRad = angle * 3.14159265f / 180.f ;
52          float startx = (radius - use_scale->height()) * cos(angleRad) + center[0];
53          float starty = -1.0f * ((radius - use_scale->height()) * sin(angleRad)) + center[1];
54          use_scale->pushRotateZoom(&face, startx, starty, 90.f-angle, 1.f, 1.f);
55          if (i%10==0) {
56              float charsize = (float)scale[0].height() / 36.f;
57              float charx = (radius - use_scale->height() * 1.5f) * cos(angleRad) + center[0];
58              float chary = -1.f * (radius - use_scale->height() * 1.5f) * sin(angleRad) +
                  center[0];
59              face.setTextSize(charsize);
60              face.drawNumber(i, charx, chary);
61          }
62          face.drawString(unit, center[0], sizey/5.f*3.f);
63      }
64
65  }
66
67  void Thermometer::drawTempMeter(int x, int y, LGFX *lcd) {
68      makeMeterFace(0, 50, "℃");
69      float center[2] = {(float)sizex/2.f, (float)sizey/2.f};
70      float angle = 270.f - 45.f;
71      angle -= 270.f / 50.f * get_temp();
72      hand.pushRotateZoom(&face, center[0], center[1], 90.f - angle, 1.f, 1.f);
73      face.pushSprite(lcd, x, y);
74  }
75
76  void Thermometer::drawHumMeter(int x, int y, LGFX *lcd) {
77      makeMeterFace(20, 80, "%");
```

```
78      float center[2] = {(float)sizex/2.f, (float)sizey/2.f};
79      float angle = 270.f - 45.f;
80      angle -= 270.f / 60.f * (get_hum() - 20.f);
81      hand.pushRotateZoom(&face, center[0], center[1], 90.f - angle, 1.f, 1.f);
82      face.pushSprite(lcd, x, y);
83  }
84
85  void Thermometer::drawString(int x, int y, LGFX *lcd) {
86      M5.SHT30.UpdateData();
87      LGFX_Sprite meter(lcd);
88      meter.setColorDepth(4);
89      meter.createSprite(250, 100);
90      meter.fillSprite(15);
91      meter.setColor(0);
92      meter.setTextColor(0, 15);
93      meter.setFont(&fonts::lgfxJapanMinchoP_36);
94      meter.setCursor(10,10);
95      meter.printf("温度:%5.1f℃", this->get_temp());
96      meter.setCursor(10,50);
97      meter.printf("湿度:%5.1f%%", this->get_hum());
98      meter.pushSprite(x, y);
99  }
100
101 float Thermometer::get_temp() {
102     M5.SHT30.UpdateData();
103     this->temp = M5.SHT30.GetTemperature();
104     return this->temp;
105 }
106
107 float Thermometer::get_hum() {
108     M5.SHT30.UpdateData();
109     this->hum = M5.SHT30.GetRelHumidity();
110     return this->hum;
111 }
```

4 ネットからの情報取得

4.1 infoFromNet.hpp

```
// ネットワークより取得する情報関連
// 時計・天気予報
#include <M5EPD.h>
#define LGFX_M5PAPER
#include <LovyanGFX.hpp>

class GetInfoFromNetwork {
    private:

        int wifiOn(void);
        void wifiOff(void);

    public:
        GetInfoFromNetwork();
        int setNtpTime() ;
        void printWifiStatus(LGFX *lcd, int x, int y) ;
        void printDateTime(LGFX *lcd, int x, int y) ;
};
```

4.2 infoiFromNet.cpp

```
#include <M5EPD.h>
#define LGFX_M5PAPER
#include <LovyanGFX.hpp>
#include <WiFi.h>
#include "infoFromNet.hpp"
#include "time.h"

// wifiid.hには、ssid,passwordの各defineを定義を記載すること。
// このファイルは、.gitignoreとする。
#include <wifiid.h>

GetInfoFromNetwork::GetInfoFromNetwork() {
}

int GetInfoFromNetwork::setNtpTime() {
    wifiOn();
    const long gmtOffset_sec = 9 * 3600;
    const int daylightOffset_sec = 0;
    const char * ntpServer = "jp.pool.ntp.org";

    configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
    struct tm timeinfo;
    if (!getLocalTime(&timeinfo)) return -1;

    rtc_time_t rtcTime;
    rtcTime.hour = (int8_t)timeinfo.tm_hour;
    rtcTime.min = (int8_t)timeinfo.tm_min;
    rtcTime.sec = (int8_t)timeinfo.tm_sec ;
    rtc_date_t rtcDate ;
    rtcDate.year = (int8_t)timeinfo.tm_year + 1900;
```

```
    rtcDate.mon = (int8_t)timeinfo.tm_mon + 1;
    rtcDate.day = (int8_t)timeinfo.tm_mday ;
    M5.RTC.setDate(&rtcDate);
    M5.RTC.setTime(&rtcTime);
    wifiOff();
    return 0;
}

int GetInfoFromNetwork::wifiOn(void) {
    WiFi.begin(ssid, password);
    for (int i = 0 ; i < 10; i++) {
        if (WiFi.status() == WL_CONNECTED) return 0;
        delay(500);
    }
    return -1;
}

void GetInfoFromNetwork::wifiOff(void) {
    WiFi.disconnect(true);
    WiFi.mode(WIFI_OFF);
}

// デバッグ用
void GetInfoFromNetwork::printWifiStatus(LGFX *lcd, int x, int y) {
    LGFX_Sprite s(lcd);
    s.setColorDepth(4);
    s.createSprite(100,30);
    s.fillSprite(15);
    s.setFont(&fonts::lgfxJapanMinchoP_20);
    s.setTextColor(0, 15);
    s.setCursor(1,1);
    s.printf("wifi:%s", (WiFi.status() == WL_CONNECTED) ? "on" : "off");
    s.pushSprite(x, y);
}

void GetInfoFromNetwork::printDateTime(LGFX *lcd, int x, int y) {
    LGFX_Sprite s(lcd);
    s.setColorDepth(4);
    s.createSprite(300,80);
    s.fillSprite(15);
    s.setFont(&fonts::lgfxJapanMinchoP_32);
    s.setTextColor(0, 15);
    s.setCursor(1,1);
    rtc_time_t rtcTime;
    rtc_date_t rtcDate;
    M5.RTC.getTime(&rtcTime);
    M5.RTC.getDate(&rtcDate);
    s.printf("%4d年%2d月%2d日\n%2d:%2d:%2d",
            rtcDate.year, rtcDate.mon, rtcDate.day,
            rtcTime.hour, rtcTime.min, rtcTime.sec);
    s.pushSprite(x, y);
}
```

5 時計の表示

5.1 tokei.hpp

```
/************************************************
 * 時計の表示
 ************************************************/

#include <M5EPD.h>
#define LGFX_M5PAPER
#include <LovyanGFX.hpp>

class Tokei {
    private:
        int year, month, day;
        int hour, min, sec;
        int dayOfTheWeek;
        int width, height;

        void getDateTime();
        int getDayOfTheWeek(int year, int month, int day) ;

    public:
        Tokei(int sizex=200, int sizey=200);
        void drawDigitalTokei(LovyanGFX *lcd, int x, int y);
};
```

5.2 tokei.cpp

```
/************************************************
 * 時計の表示
 ************************************************/

#include <M5EPD.h>
#define LGFX_M5PAPER
#include <LovyanGFX.hpp>

#include "tokei.hpp"

Tokei::Tokei(int width, int height)
        : width(width), height(height) {
    getDateTime();
    dayOfTheWeek = getDayOfTheWeek(year, month, day);
}

// RTCより現在時刻を取得する。
void Tokei::getDateTime() {
    rtc_time_t time;
    rtc_date_t date;

    M5.RTC.getTime(&time);
    M5.RTC.getDate(&date);

    year = date.year;
    month = date.mon;
```

```
27      day = date.day;
28      hour = time.hour;
29      min = time.min;
30      sec = time.sec;
31  }
32
33  // 曜日の計算。日曜日を0とする。
34  int Tokei::getDayOfTheWeek(int year, int month, int day) {
35      int y = year % 100;
36      int c = y / 100;
37      int ganma = 5 * c + c / 4;
38      return (day+(26+(month+1))/10+y+y/4+ganma+5)%7;
39  }
40
41  // デジタル時計を描画する
42  void Tokei::drawDigitalTokei(LovyanGFX *lcd, int x, int y) {
43      LGFX_Sprite tokei;
44      tokei.setColorDepth(4);
45      tokei.createSprite(width,height);
46      tokei.fillSprite(15);
47      tokei.drawRect(0,0,width,height,0); // レイアウト検討用外枠
48      tokei.setFont(&fonts::Font7); // font高さ:48
49      tokei.setTextColor(0,15);
50      char strTime[6];
51      sprintf(strTime, "%02d:%02d", hour, min);
52      float mag = (height*0.8) / 48.f;
53      tokei.setTextSize(mag, mag);
54      tokei.drawString(strTime, 0.f, height*0.2f);
55      tokei.pushSprite(lcd, x, y);
56  }
```