DS1307

#include <Wire.h>

#include <TimeLib.h>

#include <DS1307RTC.h>

const char \*monthName[12] = {

"Jan", "Feb", "Mar", "Apr", "May", "Jun",

"Jul", "Aug", "Sep", "Oct", "Nov", "Dec"

};

tmElements\_t tm;

void setup() {

bool parse=false;

bool config=false;

// get the date and time the compiler was run

if (getDate(\_\_DATE\_\_) && getTime(\_\_TIME\_\_)) {

parse = true;

// and configure the RTC with this info

if (RTC.write(tm)) {

config = true;

}

}

Serial.begin(9600);

while (!Serial) ; // wait for Arduino Serial Monitor

delay(200);

if (parse && config) {

Serial.print("DS1307 configured Time=");

Serial.print(\_\_TIME\_\_);

Serial.print(", Date=");

Serial.println(\_\_DATE\_\_);

} else if (parse) {

Serial.println("DS1307 Communication Error :-{");

Serial.println("Please check your circuitry");

} else {

Serial.print("Could not parse info from the compiler, Time=");

Serial.print(\_\_TIME\_\_);

Serial.print(", Date=");

Serial.print(\_\_DATE\_\_);

Serial.println("");

}

}

void loop() {

}

bool getTime(const char \*str)

{

int Hour, Min, Sec;

if (sscanf(str, "%d:%d:%d", &Hour, &Min, &Sec) != 3) return false;

tm.Hour = Hour;

tm.Minute = Min;

tm.Second = Sec;

return true;

}

bool getDate(const char \*str)

{

char Month[12];

int Day, Year;

uint8\_t monthIndex;

if (sscanf(str, "%s %d %d", Month, &Day, &Year) != 3) return false;

for (monthIndex = 0; monthIndex < 12; monthIndex++) {

if (strcmp(Month, monthName[monthIndex]) == 0) break;

}

if (monthIndex >= 12) return false;

tm.Day = Day;

tm.Month = monthIndex + 1;

tm.Year = CalendarYrToTm(Year);

return true;

}

Arduino Mega

#include "DHT.h"

#include <Wire.h>

#include <RTClib.h>

#define DHTTYPE DHT22

#define DHTPIN 22

#include <LiquidCrystal.h>

#include <DS1307RTC.h>

#include <Time.h>

#include <SoftwareSerial.h>

SoftwareSerial espSerial(30,31);

String str;

DHT dht(DHTPIN,DHTTYPE);

const int rs1 = 12, en1 = 11, d41 = 5, d51 = 4, d61 = 3, d71 = 2;

const int rs2 = 13, en2 = 10, d42 = 9, d52 = 8, d62 = 7, d72 = 6;

LiquidCrystal lcd1(rs1, en1, d41, d51, d61, d71);

LiquidCrystal lcd2(rs2, en2, d42, d52, d62, d72);

void setup(){

Serial.begin(115200);

espSerial.begin(115200);

dht.begin();

lcd1.begin(16,2);

lcd2.begin(16,2);

lcd2.print("This is a Clock");

delay(1000);

lcd2.clear();

}

void loop(){

int temp = dht.readTemperature();

int hum = dht.readHumidity();

lcd1.clear();

lcd1.setCursor(0,0);

lcd1.print("Temp: ");

lcd1.print(temp);

lcd1.print("\*C");

lcd1.setCursor(0,1);

lcd1.print("Humidity: ");

lcd1.print(hum);

lcd1.print("%");

tmElements\_t tm;

// 读出DS1307中的时间数据，并存入tm中

if (RTC.read(tm))

{

// 清除屏幕显示内容

lcd2.clear();

//在LCD第一行输出日期信息

lcd2.setCursor(0, 0);

lcd2.print(tmYearToCalendar(tm.Year));

lcd2.print("-");

lcd2.print(tm.Month);

lcd2.print("-");

lcd2.print(tm.Day);

//在LCD第二行输出时间信息

lcd2.setCursor(8, 1);

lcd2.print(tm.Hour);

lcd2.print(":");

lcd2.print(tm.Minute);

lcd2.print(":");

lcd2.print(tm.Second);

delay(500);

}

// 如果读取数据失败，则输出错误提示

else

{

lcd2.setCursor(0, 1);

lcd2.print("error");

}

//每秒钟更新一次显示内容

delay(500);

lcd1.clear();

lcd1.setCursor(0,0);

lcd1.print("Hello World!");

//tmElements\_t tm;

// 读出DS1307中的时间数据，并存入tm中

if (RTC.read(tm))

{

// 清除屏幕显示内容

lcd2.clear();

//在LCD第一行输出日期信息

lcd2.setCursor(0, 0);

lcd2.print(tmYearToCalendar(tm.Year));

lcd2.print("-");

lcd2.print(tm.Month);

lcd2.print("-");

lcd2.print(tm.Day);

//在LCD第二行输出时间信息

lcd2.setCursor(8, 1);

lcd2.print(tm.Hour);

lcd2.print(":");

lcd2.print(tm.Minute);

lcd2.print(":");

lcd2.print(tm.Second);

delay(500);

}

// 如果读取数据失败，则输出错误提示

else

{

lcd2.setCursor(0, 1);

lcd2.print("error");

}

//每秒钟更新一次显示内容

delay(500);

//delay(2000);

lcd1.clear();

lcd1.setCursor(0,1);

lcd1.print("Erzhuang Gao");

//tmElements\_t tm;

// 读出DS1307中的时间数据，并存入tm中

if (RTC.read(tm))

{

// 清除屏幕显示内容

lcd2.clear();

//在LCD第一行输出日期信息

lcd2.setCursor(0, 0);

lcd2.print(tmYearToCalendar(tm.Year));

lcd2.print("-");

lcd2.print(tm.Month);

lcd2.print("-");

lcd2.print(tm.Day);

//在LCD第二行输出时间信息

lcd2.setCursor(8, 1);

lcd2.print(tm.Hour);

lcd2.print(":");

lcd2.print(tm.Minute);

lcd2.print(":");

lcd2.print(tm.Second);

delay(500);

}

// 如果读取数据失败，则输出错误提示

else

{

lcd2.setCursor(0, 1);

lcd2.print("error");

}

//每秒钟更新一次显示内容

delay(500);

//delay(2000);

//tmElements\_t tm;

// 读出DS1307中的时间数据，并存入tm中

if (RTC.read(tm))

{

// 清除屏幕显示内容

lcd2.clear();

//在LCD第一行输出日期信息

lcd2.setCursor(0, 0);

lcd2.print(tmYearToCalendar(tm.Year));

lcd2.print("-");

lcd2.print(tm.Month);

lcd2.print("-");

lcd2.print(tm.Day);

//在LCD第二行输出时间信息

lcd2.setCursor(8, 1);

lcd2.print(tm.Hour);

lcd2.print(":");

lcd2.print(tm.Minute);

lcd2.print(":");

lcd2.print(tm.Second);

delay(500);

}

// 如果读取数据失败，则输出错误提示

else

{

lcd2.setCursor(0, 1);

lcd2.print("error");

}

//每秒钟更新一次显示内容

delay(500);

Serial.print("Temp: ");

Serial.print(temp);

Serial.println("\*C");

Serial.print("Humidity: ");

Serial.print(hum);

Serial.println("%");

str = String("Coming from Arduino: ") + String("Temp: ") + String(temp) + String("\*C") + String(" ") + String("Humidity: ") + String(hum) + String("%")+ String(" ")

+ tmYearToCalendar(tm.Year) + String("-") + tm.Month + String("-") + tm.Day + String(" ") + tm.Hour + String("-") + tm.Minute + String("-") + tm.Second;

espSerial.println(str);

delay(1);

}

ESP8266 WIFI Module

#include <Arduino.h>

#include <ESP8266WiFi.h>

#include <ESPAsyncTCP.h>

#include <ESPAsyncWebServer.h>

#include <LittleFS.h>

#include <FS.h>

// Replace with your network credentials

const char\* ssid = "Curry";

const char\* password = "Yuzihao0923";

// Create AsyncWebServer object on port 80

AsyncWebServer server(80);

String file\_name = "/index.html";

String txt;

void initFS() {

if (!LittleFS.begin()) {

Serial.println("An error has occurred while mounting LittleFS");

}

else {

Serial.println("LittleFS mounted successfully");

}

}

void initWiFi() {

WiFi.mode(WIFI\_STA);

WiFi.begin(ssid, password);

Serial.print("Connecting to WiFi ..");

while (WiFi.status() != WL\_CONNECTED) {

Serial.print('.');

delay(1000);

}

Serial.println(WiFi.localIP());

}

void setup() {

Serial.begin(115200);

while(!Serial){

;

}

initWiFi();

initFS();

//

server.on("/", HTTP\_GET, [](AsyncWebServerRequest \* request) {

request->send(LittleFS, "/index.html", "text/html");

});

server.serveStatic("/", LittleFS, "/");

server.begin();

}

void loop() {

}

HTML/CSS

<!DOCTYPE html>

<br lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Internet Clock/Weather Station</title>

<style>

body{

background-image: url(background.png);

background-repeat:no-repeat;

background-position: center top;

background-color: rgba(220, 38, 38, 0.2);

}

.center{

text-align: center;

}

.kevin{

display: flex;

justify-content: space-around;

align-items: center;

}

h1 {

color: rgb(0, 255, 64);

font-size: 50px;

}

h2{

color: blueviolet;

font-size: 30px;

}

table{

align-items: center;

}

figure{

margin: 0;

}

.wrap{

display: grid;

grid-template-columns: repeat(5, 1fr);

overflow-x: auto;

padding-inline: 6vmin;

padding-block: 12vmin;

gap: 3vmin;

scroll-snap-type: x mandatory;

}

.product-card{

width: 86vw;

box-shadow: 0 2px 40px rgba(0, 0, 0, 15%);

padding: 1.2rem;

border-radius: 10px;

scroll-snap-align: center;

scroll-snap-stop: always;

}

.product-card\_hardware{

font-size: 0.8rem;

color: rgb(102, 102, 102);

}

.product-card\_software{

font-size: 0.8rem;

color: #666;

}

.product-card\_HTML{

font-size: 0.8rem;

color: #666;

}

</style>

</head>

<body>

<h1 id="top">

<div class="top">

<div>

<a href="http://www.ait.ie" target="\_blank">

<img src="https://www.ait.ie/assets/favicon/favicon-196x196.png" />

</a>

</div>

<div>

<a href="https://2021.elearning.ait.ie/login/index.php" target="blank">

<img src="TUS.png" width="196" height="196"/>

</a>

</div>

<span class="center">

<strong>Project - Internet Clock/Weather Station </strong>

</span>

</div>

<br />

<br />

<input type="file" name="inputfile"

id="inputfile">

<br>

<pre id="output"></pre>

<script type="text/javascript">

document.getElementById('inputfile')

.addEventListener('change', function() {

var fr=new FileReader();

fr.onload=function(){

document.getElementById('output')

.textContent=fr.result;

}

fr.readAsText(this.files[0]);

})

</script>

<br />

<br />

<br />

<div>

<h1><strong>Introducion</strong></h1>

<h2>

<span>

<strong>Teacher:</strong>

<div class="kevin">

<img src="kevin.jpg" width="300">

</div>

</span> </div></br>

</h2>

<h2>

<span>

<strong>Student:</strong>

<table align="center" border="1">

<tr>

<th>Zihao Yu</th>

<th>Erzhuang Gao</th>

<th>Ernest</th>

</tr>

<tr>

<td align="center"><img src="zihao.jpg" width="100"></td>

<td align="center"><img src="erzhuang.jpg" width="100" height="100"></td>

<td><img src="Ernest.jpg" width="100" height="100"></td>

</tr>

</table>

</span></br>

</h2>

</div>

<br>

<span>

<h1>

Details for our Project

</h1>

</span></br>

<span>

<h2>

(i):Overview of Project

</h2>

</span></br>

<span>

<h2>

(ii):The compent we used

</h2>

<table border="1" cellpadding="10">

<thead>

<tr>

<th>Name</th>

<th>Quality</th>

<th>Function</th>

</tr>

</thead>

<tbody>

<tr>

<td><a href="http://store.arduino.cc/products/arduino-mega-2560-rev3#" target="blank">Arduino Megia</a></td>

<td>1</td>

<td><a href="Arduino\_mega.html" target="\_blank">The main compenent to control all the parts</a></td>

</tr>

<tr>

<td><a href="https://www.espressif.com/en/products/socs/esp8266" target="blank">Esp8266</a></td>

<td>1</td>

<td>Like a server connect Arduino and website</td>

</tr>

<tr>

<td><a href="https://learn.adafruit.com/dht" target="blank">Dht-22</a></td>

<td>1</td>

<td>Detects temperature and humidity</td>

</tr>

<tr>

<td><a href="https://create.arduino.cc/projecthub/najad/interfacing-lcd1602-with-arduino-764ec4" target="blank">LED 1602A</a></td>

<td>2</td>

<td><a href=1602A.html target="\_blank">Display the words</a></td>

</tr>

<tr>

<td><a href="https://www.maximintegrated.com/cn/products/analog/real-time-clocks/DS1307.html" target="blank">DS1307</a></td>

<td>1</td>

<td>Create a clock to Arduino</td>

</tr>

<tr>

<td>Wires</td>

<td>several</td>

<td>Connect between components</td>

</tr>

</tbody>

</table>

</span></br>

<span>

<h2>

(iii):The process recored

</h2>

<section class="wrap">

<div class="product-card\_hardware">

<div class="product-card\_name">

Erzhuang Gao

</div>

<figure class="product-card\_image" >

<img src="picture1.jpg" width="500xp">

</figure>

</div>

<div class="product-card\_software">

<div class="product-card\_name">

Ernest

</div>

<figure class="product-card\_image" >

<img src="picture2.jpg" width="500xp">

</figure>

</div>

<div class="product-card\_HTML">

<div class="product-card\_name">

Zihao Yu

</div>

<figure class="product-card\_image">

<img src="picture3.jpg" width="500xp">

</figure>

</div>

</section>

</span></br>

</div>

</h1>

</body>

</html>

Python

# encoding=utf-8

import serial

import time

if \_\_name\_\_ == '\_\_main\_\_':

com = serial.Serial('COM5', 115200)

over\_time = 10000000000

start\_time = time.time()

while True:

end\_time = time.time()

f = open("hello1.txt", 'a')

if end\_time - start\_time < over\_time:

data = com.read(67)

data = str(data)

if data != '':

print(data)

f.write(data + "\n")

f.flush()