به نام خدا

ساعت دیجیتال عقربه ای

پروژه درس سیستم های نهفته

جناب آقای دکتر انصاری

سارا آذرنوش ۹۸۱۷۰۶۶۸

علی حاتمی ۹۸۱۰۱۳۸۵

رویا قوامی ۹۸۱۷۱۰۳۱

زمستان ۱۴۰۰

فهرست

مقدمه

در این پروژه یک ساعت عقربه ای با استفاده از امکانات دیجیتال بصورتی که تمام بخش های این ساعت (عقربه ها و محیط دایره ساعت) از ۳ حلقه ی RGB LED ساخته شده است و با استفاده از یک اپلیکیشن موبایل میتوان بصورت بی سیم رنگ و همچنین تنظیمات زمان ساعت این ساعت را تغییر داد.

طرح ساعت

طراحی مد نظر به این صورت است که نمایش زمان با استفاده از سه حلقه از LED های RGB با اندازه‌های ۶۰، ۲۴ و ۱۲ صورت خواهد گرفت. این حلقه ها بر روی یک پایه که حاوی ماژول های مورد نیاز ساعت است قرار میگیرد.

شکل صفحه ساعت (Watch Face) به صورت زیر است:

لوازم مورد نیاز

لوازم مورد نیاز با توجه به ناحیه مصرف آنها عبارت‌اند از:

- **پردازشی :**

Cp2102 با چیپ EPS8266 بر پایه‌ی NodeMCU

- **روشنایی :**

برای روشنایی این ساعت از led های WS2812B پکیج ۵۰۵۰ استفاده خواهد شد. پیشتر استفاده از این تکنولوژی برای کنترل کردن led ها به خصوص در تعداد بالا شبیه به یک کابوس بود اما با استفاده از این led های آدرس پذیر با اتصال آنها به صورت daisy chain و تنها با استفاده از یک خط داده میتوان آنها را کنترل کرد. برای این ساعت از ۹۶ عدد از آنها به شرح زیر استفاده خواهد شد :

- NeoPixel Ring - X60 5050 RGB LED WS2812B
- NeoPixel Ring - X24 5050 RGB LED WS2812B
- NeoPixel Ring - X12 5050 RGB LED WS2812B

- **ساعت:**

Real-time Clock DS1307 with I2C Interface

- **توان**

این ساعت از ۹۶ چراغ LED WS2812B تشکیل شده است که در حالتی که خاموش باشند ۱ میلی آمپر و در حالت روشنایی ۶۰ میلی آمپر مصرف دارند. برای کنترل مصرف این ساعت از این تکنیک (به پیشنهاد مهندس سیادت‌زاده) استفاده خواهد شد که ساعت همیشه روشن نخواهد بود و تنها در رویدادهای خاصی (مثل دریافت صدای کف‌زدن یا بر اساس یک تایمر خاص) زمان را نشان خواهد داد. برای توان بخشی از مدار از ملزومات زیر استفاده خواهد شد:

2200 - 18650 Battery Shield V3:

این شیلد دارای یک، دو یا چهار باطری است که خروجی ثابت ۵ ولت و ۳ ولت به ما تحویل می‌دهد و برای روشن کردن چراغ‌ها و میکروکنترلر استفاده میشود. این شیلد کار تنظیم و مصرف توان را بسیار ساده می‌کند.

Voltage Sensor:

از یک سنسور ولتاژ برای گرفتن ولتاژ خام دو سر باتری استفاده می‌شود تا با توجه به ولتاژ آن بتوان میزان باطری باقی‌مانده درون آن را خواند و با توجه به منحنی شارژ دهی-میزان شارژ باقی‌مانده شارژ باطری را نمایش داد. البته شیلد باطری خود این کار را می کند اما چون می‌خواهیم این اطلاعات از طریق نرم‌افزار قابل دسترسی باشند به این سنسور نیاز خواهیم داشت.

- **صوت :**

در بخش اختیاری است و می‌توان صوت‌های دلخواه را روی یک میکرو sd ریخت یا اینکه برای درایو بهتر اسپیکر از یک آمپلیفایر استفاده کرد. اما می‌توان تنها به اسپیکر اکتفا کرد و با استفاده از تن‌ها، موسیقی‌های ابتدایی برای زنگ و دیگر کارکردها ساخت.

- 0.5 Watt 8 Ohm Speaker
- (Optional) LM386 Mini Amplifier
- (Optional) Micro SD Card Module

- **سنسورها:**

- Voice Sensor FC-04
- TTP223 Touch Sensor
- DS18B20 Temperature Sensor

- **ملزومات ساخت:**

- Printed watch face and body
- Mini Breadboard
- Jumper Wires
- Soldering iron + ...

قابلیت‌ها

- نمایش ساعت (با چند نحوه نمایش متفاوت)
- دارای mode هایی برای استفاده به عنوان چراغ خواب یا تزئین
- قابلیت کنترل و تغییر برنامه و اطلاع از وضعیت فعلی توسط اپلیکیشن موبایل
- هشدار (آلارم)
- حالات متفاوت رنگ
- نمایش تاریخ
- نمایش دما

مراحل کار

نوشتن پروپوزال:

نیازمندی‌ها و هدف از انجام پروژه را مشخص کرده و به صورت کتبی در می‌آوریم.

اسلاید تهیه شده و پروپوزال گیت هاب در این لینک‌ها موجود هستند.

تهیه قطعات مورد نیاز:

در ابتدا با توجه به نیازمندی های پروژه، قطعات مورد نیاز گفته شده را تهیه میکنیم.

ساخت قالب موردنظر:

برای ساخت به قالب فیزیکی ساعت به دو بخش زیر نیاز داریم.

ساعت:

سه حلقه ۶۰، ۲۴ و ۱۲ عددی که به ترتیب اندازه به شکل مماس در شمال غربی حلقه ۶۰ عددی قرار میگرند و ابعاد آن متناسب با رشته LED تهیه شده است.

پایه:

یک مکعب مستطیل خالی که ماژول های مورد نیاز گفته شده در داخل آن قرار میگیرند و ساعت روی آن قرار میگرد.

نتیجه نهایی در گیت هاب موجود است.

ساخت فیزیکی و انجام اتصال فیزیکی:

ریسه‌های LEDرا در حلقه مورد نظر قرار داده و میچسبانیم سپس آن ها را به صورت سریالی به یکدیگر لحیم میکنیم. ماژول های داخل نیز که شامل باتری و سنسور ولتاژ متصل به آن، ماژول ساعت و صدا هستند را همگی را به بلوتوث متصل میکنیم.

نوشتن برنامه‌ی سخت افزار:

برای نوشتن کدهای سخت افزاری از Arduino و زبان c++ استفاده شده است.

❋ **کار با میکروفون(MIC):**

شکل ظاهری سخت افزار:

تابع checkTrigger: این تابع برای کار با میکروفون است که دو flag دریافت می کند و داخل loop صدا می شود و دارای دو state است:

۱. "دست زدن اول" (first clap) شناسایی یا capture نشده باشد .

۲. "دست زدن اول" شناسایی شده باشد و منتظر "دست زدن دوم" می ماند و اگر رخ نداد، اولی را نیز حذف می کند.

درواقع بعد از گذشتن زمانی ما بین ۳۵۰ تا ۱۳۰۰ میلی ثانیه اگر "دست زدن دوم" تشخیص داده شد، به عنوان double clap درنظر گرفته می شود و در غیر این صورت، "دست زدن اول" ی حذف می شود و دوباره مراحل داخل لوپ تکرار میشوند.

- **کار با اسپیکر:**

در این پروژه با اتصال یک اسپیکر به پین (pin) یازدهم (می توان پین جدید انتخاب کرد) میتوان آهنگ Super Mario Bros - Overworld theme را پخش کرد. فایل مربوطه به کدهای این قسمت در این لینک قابل مشاهده است.

در این پروژه با اتصال یک زنگ پیزو به پین (pin) یازدهم (می توان پین جدید انتخاب کرد) میتوان آهنگ Super Mario Bros - Overworld theme را پخش کرد. فایل مربوطه به کدهای این قسمت در این لینک قابل مشاهده است.

کار با LED های ساعت و مودهای مختلف آن:

در این بخش، برای کنترل پیکسل ها و نوارهای LED مبتنی بر تک-سیم (single wire) از کتابخانه آردوینو (Ardunio library) استفاده شده است. این کتابخانه با تمام معماری ها سازگار بوده و بنابراین می توان از آن در تمام بردهای آردوینو استفاده کرد.
در این معماری، از سه حلقه LED با اندازه های ۶۰، ۲۴ و ۱۲ استفاده شده است و در نتیجه برای مپ کردن عدد پیکسل گفته شده به هر یک، map60، map24 و map12 را تعریف می کنیم.
ساعت دارای سه استیت کلی FF_MODE، SWING_MODE و SIMPLE_MODE برای نمایش زمان می باشد. در فایل کدهای مربوط به این قسمت (RGB.ino) به کمک توابع تعریف شده در کتابخوانه آردوینو، مانند setPixelColor و مپ های تعریف شده که بالا به آن اشاره کردیم، بعد از گرفتن زمان حال حاضر و استفاده از لیست colorScheme که در آن ۱۲ رنگ تعریف کردیم، mode های مختلف ساعت را می توان با روشن و خاموش کردن پیکسل ها نمایش داد.

- **کار با RTC (Real Time Clock):**

برای کار با RTC از کتابخانه RTClib استفاده می کنیم. این کتابخانه به آردوینو اجازه می دهد تا RTC داخلی (Real Time Clock) را کنترل و استفاده کند. ساعت بلادرنگ ساعتی است که زمان حال حاضر را ردیابی می کند و می توان از آن برای برنامه ریزی اقدامات برنامه در یک زمان خاص استفاده کرد.
به علاوه، RTC می تواند در هر حالت sleep mode به کار خود ادامه دهد، بنابراین، می توان از آن برای اصطلاحا "بیدار کردن " دستگاه از حالت های خواب استفاده کرد. هر بار که برد روشن می شود، RTC ریست می شود و از یک تاریخ استاندارد شروع به کار می کند.
در فایل مربوط به این قسمت (RTC.ino)، تابعی برای تشخیص آلارم (isAlarm)، تابعی برای به روزرسانی زمان (به کمک تابع rtc.now())، تابعی برای ست کردن RTC به تاریخ و زمان صریح به صورت زیر و در نهایت نیز تابع initializer آن قرار دارند.

January 21, 2014 at 3am rtc.adjust(DateTime(2014, 1, 21, 3, 0, 0))

- **کار با سنسور دما:**

کد مربوط به این قسمت در Temprature.ino قرار دارد. برای کار با سنسور دما، از کتابخانه های DallasTemperature.h و OneWire استفاده می کنیم. ابتدا یک instance از OneWire ساخته تا بتوان با هر وسیله ای که با یک wire می تواند وصل شود ارتباط برقرار کرد و سپس آن را به کتابخانه DallasTemperature.h می دهیم. در نهایت، به کمک تابع getTemprature، دمای محیط را به سیلسیوس بر میگردانیم.

ارتباط با اپ موبایل:

در فایل SaRoyAl.ino، کد بدنه ساعت دیجیتال قرار دارد که به توضیح آن می پردازیم:

- ریکوست هایی که می توان به ساعت فرستاد:

/ : برای تست ارتباط مناسب

status/: پاسخ این ریکواست، درصد باتری، دمای محیط و زمان فعلی است.

adjust/: با فرستادن این ریکواست، ساعت دیجیتال با ساعت محلی موبایل تنظیم می شود.

alarm/: با فرستادن این ریکواست، تنظیمات آلارم برای موبایل فرستاده می شود.

setting/: با فرستادن این ریکواست، تنظیمات استیت ها برای موبایل فرستاده می شوند.

colors/: با فرستادن این ریکواست، تنظیمات رنگ برای گوشی فرستاده می شود.

post/: با فرستادن این ریکواست، تنظیمات مورد نظر کاربر از گوشی برای میکرو ارسال می شود.

- با استفاده از کتابخانه ESPAsyncServer، یک سرور (تقریبا) آسنکرون ساخته و روی ریکواست های بالا set شده است.

- استیت های پیاده سازی شده در سیستم نیز به شرح زیر است:

۱. ساعت (دارای سه مدل نمایش)

۲. نشان دادن تاریخ (جلالی و میلادی)

۳. نشان‌دادن درصد باطری (با استفاده از خواندن ولتاژ و مپ کردن به منحنی که شرکت سازنده ارائه کرده)

۴. نشان دادن دما (با استفاده از سنسور دمای ds18b20)

۵. دو مود زیبایی رنگین‌کمان و پک‌من

تمامی استیت های بالا قابلیت تنظیم دارند که نشان داده شوند یا خیر. علاوه بر این، رنگ استیت های ۱ و ۲ که مربوط به نمایش ساعت و نمایش تاریخ می باشند قابل تنظیم است. رنگ استیت های ۳ و ۴ نیز متناسب با وضعیتشان انتخاب می شود (به طور مثال اگر باطری ساعت کم باشد رنگ قرمز اگر کامل باشد رنگ آبی و مابین آن رنگ زرد اختصاص داده می شود و رنگ دما نیز متناسب با دما از رنگ قرمز برای دمای ۶۰ درجه سیلسیوس تا رنگ بنفش برای دمای ۲۰- درجه سیلسیوس تنظیم می شود).

- **روند اجرا**

ساعت زمانی نشان داده می شود که user دوبار دست بزند. درواقع زمانی که ساعت خاموش است و trigger نشده است، در basic state عه IDLE هستیم که منتظر capture کردن "دوبار دست زدن" است که بالاتر توصیح دادیم. سپس زمانی که trigger شد، ساعت شروع به operate میکند و داخل و از IDLEخارج می شویم. سپس به کمک تابع handleStates()، استیت های مختلف کنترل و هندل می شوند و زمانی که کار آن تمام شد، کنترلر استیت بعدی را مشخص می کند.

ساخت برنامه‌ی موبایل:

برای کار با ساعت دیجیتال و فرستادن ریکواست به آن، یک اپ اندروید برای موبایل طراحی شد که با وصل شدن به سرور، کاربر بتواند به طور بصری، از فیچرهای پیاده سازی شده استفاده کند. این اپ شامل:

- دو فیلد برای گرفتن عدد آلارم
- یک قسمت برای انتخاب رنگ و حالت ساعت (یکی از سه حالت نمایش دادن) که شامل چهار فیلد رنگ است
- دو دراپ داون برای انتخاب حالت
- دو فیلد عددی برای آلارم
- هفت سوئیچ می باشد

که همانطور که بالاتر گفته شد، زمانی که کاربر از فیلدهای گفته شده استفاده می کند، عملیات مربوطه به صورت ریکواست به سروری فرستاده می شود که ساعت به آن وصل است و سپس بعد از دریافت ریکواست، پاسخ مناسب به آن انجام می شود.

ارتباط و تست:

اپلیکیشن ساخته شده را به ساعت اتصال داده و ویژگی‌های خواسته شده را برای اطمینان از صحت عملکرد تست میکنیم.

چالش‌ها:

فیزیکی:

- سایز مدل درنظر گرفته شده برای رشته ی LED ها مناسب نبود و از ابتدا قالب را طراح کردیم.
- به علت ظریف بودن حلقه ها، اتصال و لحیم کاری آن دشوار بود و همچنین سیم در پشت آن مشخص میشد بنابراین برای تمیزکاری بیشتر نزدیکترین ها را به یکدیگر متصل کردیم که موجب پشت هم نبودن شماره‌ی LED ها میشد که آدرس دادن در کد را دشوارتر میکرد بنابراین برای راحتی کار شماره‌ی LED ها را در تابع ها برای سادگی مپ کردیم تا شماره های مناسب را خروجی دهد.

اپلیکیشن اندروید:

- در ابتدا از کتابخانه ی RemoteXY استفاده میکردیم که در اواسط کار دچار مشکل شد و مجبور شدیم از پایه بنویسیم.
- برای ساعت REST API نوشته شده است تا انعطاف بالایی سمت کاربر داشته باشد.

نتیجه

ویدو از ساعت ساخته شده.

ساعت پیاده سازی شده:

مود اول ساعت:

مود دوم ساعت:

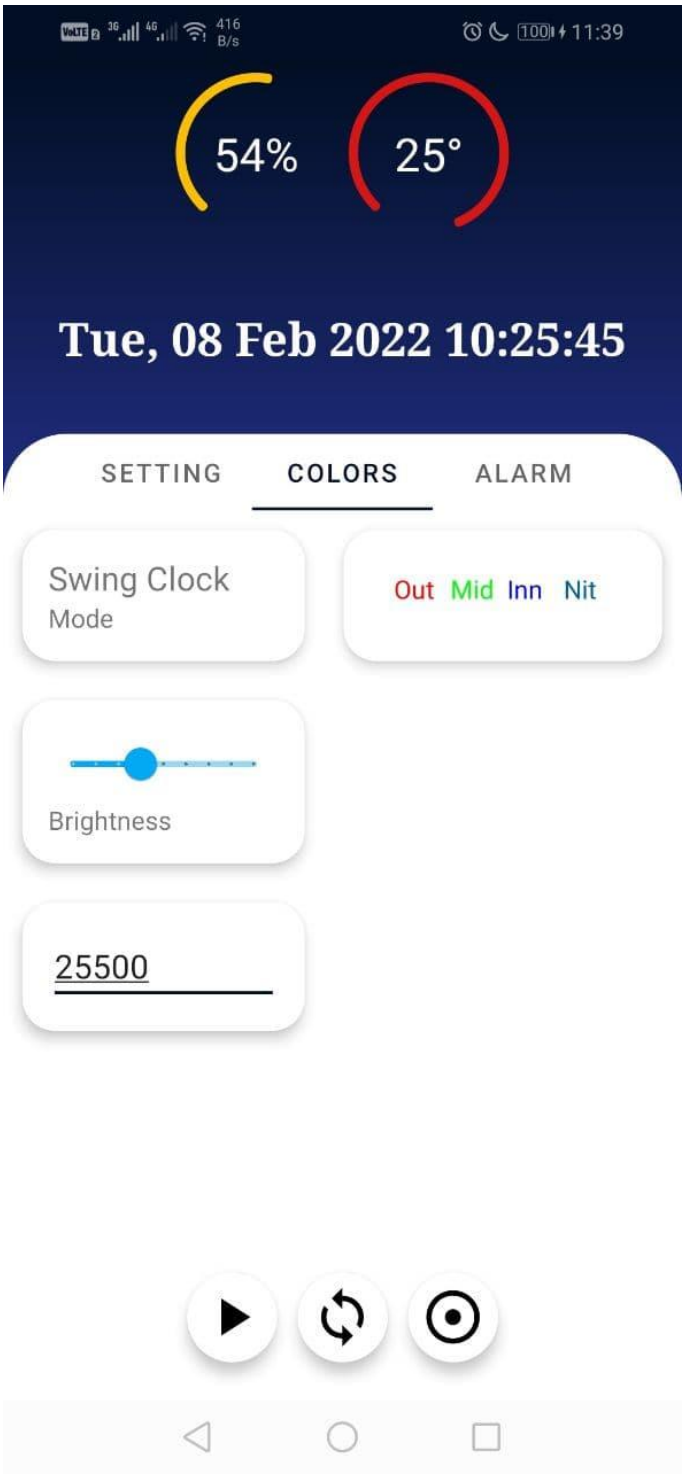
تاریخ جلالی

دما

باتری

پکمن

اپلیکیشن موبایل


11:39
54%
25°
Tue, 08 Feb 2022 10:25:45
SETTING
COLORS
ALARM
Swing Clock
Mode
Out Mid Inn Nit
Brightness
25500

محل شارژ

اسپیکر و میکروفون

منابع

1. https://www.instructables.com/Neopixel-Clock-With-Three-Neopixel-Rings/
2. https://www.youtube.com/watch?v=AR_XrYeBeIU&ab_channel=KieranGleeson
3. https://learn.adafruit.com/neopixel-ring-clock/code-and-clock-faces
4. https://github.com/milesburton/Arduino-Temperature-Control-Library
5. https://www.arduino.cc/en/Reference/RTC
6. https://github.com/me-no-dev/ESPAsyncWebServer
7. https://www.amazon.com/piezo-buzzer/s?k=piezo+buzzer
8. https://adafruit.github.io/RTClib/html/index.html