



گزارش میانی دوم آزمایشگاه سخت افزار

گروه سه

اعضا:

بهار خدابخشیان ۹۷۱۰۵۹۰۶

محمدامین شریفی ۹۷۱۱۰۰۵۸

درنا دهقانی ۹۷۱۰۵۹۳۹

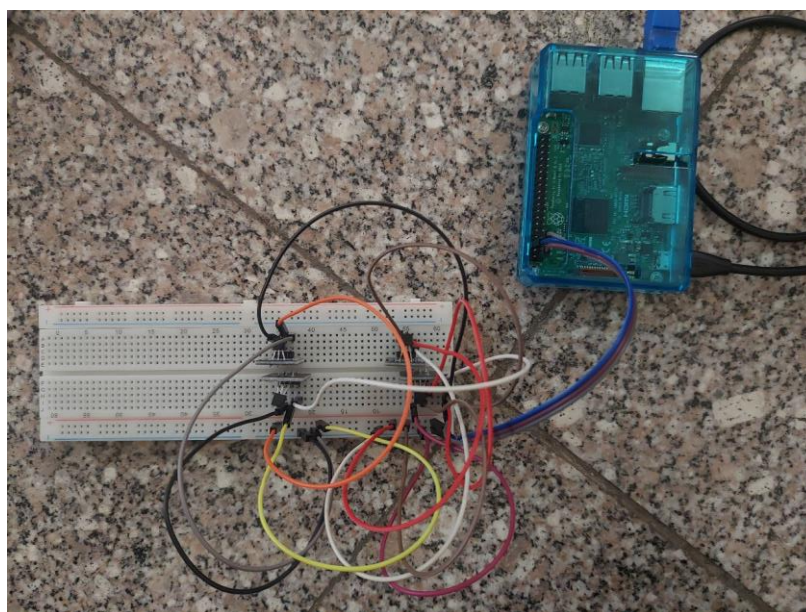
شرح فعالیت‌ها

فعالیت‌های این بخش از پروژه شامل چند قسمت بود که هر یک در هر بخش بطور مجزا توضیح داده خواهند شد. بطور کلی، ساختار اصلی پروژه در این قسمت توسط زبان پایتون پیاده‌سازی شد که شامل گرفتن دما از سنسورهای مخصوص، گرفتن مختصات از سنسور GPS و تشخیص جهت تابش آفتاب به کمک مختصات می‌باشد.

۱. گرفتن دما

برای این بخش از پروژه سنسور DS18B20 را انتخاب کرده بودیم که به شکل ماژول به ما داده شد. در این ماژول خروجی‌های G، R و Y به ترتیب باید به پین‌های GND، VCC ۳،۳ یا ۵ ولت و GPIO 4 (GPCLK0) متصل شوند. پین‌های [GPCLK](#) یا General Purpose Clock در رزبری پای پین‌هایی هستند که خروجی را با فرکانس ثابت و بدون نیاز به کنترل نرم‌افزاری ایجاد می‌کنند.

ابتدا می‌خواهیم نحوه نمایش دما توسط این سنسور را ببینیم. برای تنظیمات اولیه و ایجاد رابط یک طرفه بین ماژول و رزبری پای، پس از روشن کردن رزبری پای با زدن دستور `sudo nano /boot/config.txt`، در انتهای فایل باز شده عبارت `dtoverlay=w1-gpio` را اضافه می‌کنیم. سپس رزبری پای را `reboot` می‌کنیم. پس از روشن شدن مجدد و زدن دو دستور `sudo modprobe w1-gpio` و `sudo modprobe w1-therm` با دستور `cd /sys/bus/w1/devices` دایرکتوری را تغییر می‌دهیم. به کمک `ls` می‌توانیم لیست دایرکتوری‌های موجود در آن را مشاهده کنیم. دایرکتوری که اسم با فرمت `-28` دارند شامل اطلاعات دما هستند. با `cd` وارد آن شده و با دستور `cat w1_slave` فایل موجود در آن را باز می‌کنیم. با هر بار باز کردن این فایل، اطلاعات دمای گرفته شده در هر کلاک را مشاهده می‌کنیم که به شکل `t=XXXXX` داده شده است و نشان می‌دهد دما `xx.xxxx` درجه سانتیگراد است. از آنجایی که در این پروژه نیاز به ۴ ماژول دمای جداگانه داریم، باید برای دریافت اطلاعات از هر یک نیز جداگانه عمل کنیم. برای این کار کافی‌ست هر ۱ یا ۲ ردیف از بردبرد را به یکی از پین‌های رزبری پای اختصاص دهیم و پین‌های هر یک از چهار ماژول دما به آن ردیف‌ها متصل کنیم. نتیجه در تصویر زیر قابل مشاهده است:



تصویر ۱ قرارگیری سنسورهای دما

پس از اتصال صحیح هر چهار ماژول دما، در پوشه **devices** که پیش تر تنها یک دایرکتوری که با فرمت اسم -28 داشت، اکنون چهار دایرکتوری با این مشخصات قابل مشاهده است که فایل **w1_slave** در هر دایرکتوری اطلاعات دمای گرفته شده توسط آن ماژول را نمایش می‌دهد.

```

mmd@raspberrypi: ~/Desktop/project/hw_project-group-3/Code
Sensor 1 => 25.5 (C) - 77.9 (F)
Sensor 2 => 25.375 (C) - 77.675 (F)
Sensor 3 => 25.062 (C) - 77.11160000000001 (F)
Sensor 4 => 27.562 (C) - 81.61160000000001 (F)
Sensor 1 => 25.562 (C) - 78.0116 (F)
Sensor 2 => 25.375 (C) - 77.675 (F)
Sensor 3 => 25.0 (C) - 77.0 (F)
Sensor 4 => 26.562 (C) - 79.8116 (F)
Sensor 1 => 25.5 (C) - 77.9 (F)
Sensor 2 => 25.375 (C) - 77.675 (F)
Sensor 3 => 25.0 (C) - 77.0 (F)
Sensor 4 => 26.062 (C) - 78.9116 (F)
Sensor 1 => 25.5 (C) - 77.9 (F)
Sensor 2 => 25.375 (C) - 77.675 (F)
Sensor 3 => 25.0 (C) - 77.0 (F)
Sensor 4 => 25.75 (C) - 78.35 (F)
Sensor 1 => 25.562 (C) - 78.0116 (F)
Sensor 2 => 25.375 (C) - 77.675 (F)
Sensor 3 => 25.062 (C) - 77.11160000000001 (F)
Sensor 4 => 25.625 (C) - 78.125 (F)
Sensor 1 => 25.562 (C) - 78.0116 (F)
Sensor 2 => 25.375 (C) - 77.675 (F)
Sensor 3 => 25.062 (C) - 77.11160000000001 (F)

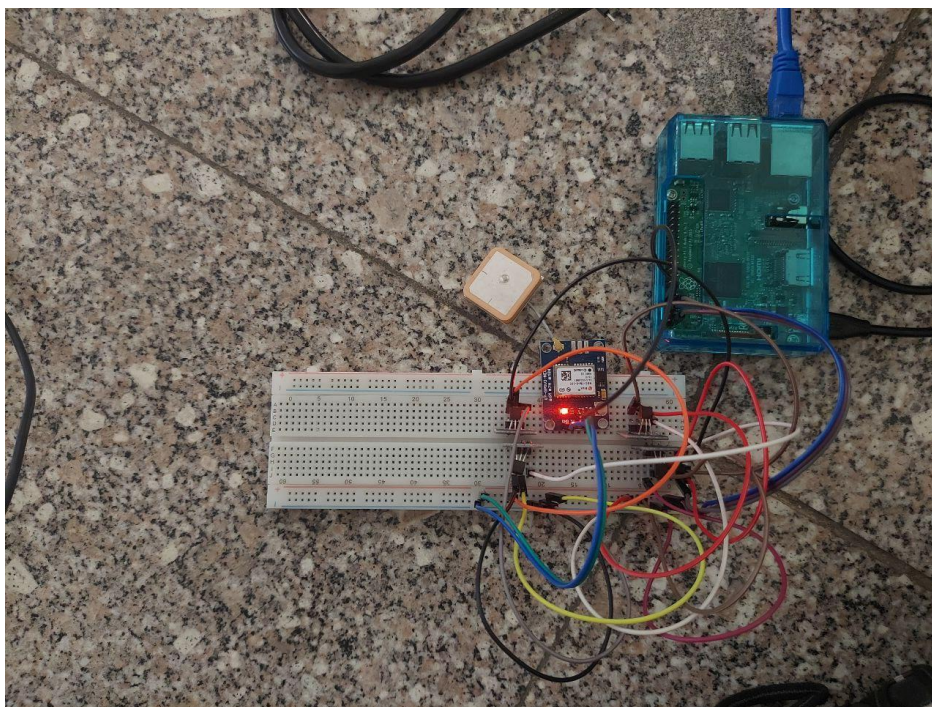
```

تصویر ۲ نمایش دمای چهار سنسور دما

در نهایت، برای استفاده از این دماها برای مراحل آتی از کد پایتون استفاده می‌کنیم که دما را از فایل‌های گرفته شده می‌خواند. این کد در فایل **temp_sensor.py** قابل مشاهده است. اطلاعات ذکر شده در این بخش عمدتاً از [این لینک](#) گرفته شده است.

۲. گرفتن مختصات توسط GPS

برای دریافت مختصات خودرو از سنسور Neo-6m-C استفاده کردیم که ۴ پین **VCC**، **GND**، **TX** و **RX** آن به ترتیب به پین‌های ۳، ۵ یا ۳، ۵ ولت، **GND**، **RX** (GPIO 15) و **TX** (GPIO 14) متصل می‌شوند. در شکل زیر می‌توان این ماژول را پس از قرارگیری در بردبورد مشاهده کرد:



تصویر ۳ قرارگیری سنسور GPS

برای برقراری اتصال بین سنسور و رزبری پای لازم است تنظیماتی انجام گردند. پس از روشن کردن رزبری پای با زدن دستور

`sudo nano /boot/config.txt`، در انتهای فایل باز شده عبارات

```
dtoverlay=spi=on
dtoverlay=pi3-disable-bt
core_freq=250
enable_uart=1
force_turbo=1
```

را اضافه کرده و ذخیره می‌کنیم. از آنجایی که سیستم عامل رزبری از UART به عنوان یک کنسول سریال استفاده می‌کند، برای

خاموش کردن این عملکرد باید فایل `/boot/cmdline.txt` را تغییر دهیم. برای این کار ابتدا از نسخه اصلی آن با دستور

`sudo cp /boot/cmdline.txt /boot/cmdline_backup.txt` یک فایل پشتیبان تهیه می‌کنیم. سپس با دستور

`nano /boot/cmdline.txt` آن را باز کرده و محتوای آن را با عبارت زیر جایگزین کرده و ذخیره می‌کنیم.

```
dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline
fsck.repair=yes rootwait quiet splash plymouth.ignore-serial-consoles
```

سپس رزبری پای را `reboot` می‌کنیم.

پس از این تنظیمات می‌توانیم کارکرد سنسور را مشاهده می‌کنیم. پس از اتصال سنسور به رزبری پای، یک LED قرمز رنگ روی

آن روشن می‌شود که بیانگر روشن بودن سنسور است. پس از مدتی نسبتاً طولانی، یک LED سبز رنگ شروع به چشمک زدن

می‌کند که بیانگر دریافت مختصات است. اگر این زمان از ۵ دقیقه بیشتر شد باید کارکرد آن در هوای آزاد یا نزدیک پنجره بررسی

شود. پس از شروع چشمک زدن، با اجرای دستور `sudo cat /dev/ttyAMA0` اطلاعات مختصات دریافت شده توسط سنسور

مطابق تصویر زیر نمایش داده می‌شود:

```

mmd@raspberrypi:~/Desktop/project/hw_project-group-3/Code $ sudo ~
sudo: /home/mmd: command not found
mmd@raspberrypi:~/Desktop/project/hw_project-group-3/Code $ cd ~
mmd@raspberrypi:~$ ls
bookshelf Documents Music Public Videos
Desktop Downloads Pictures Templates
mmd@raspberrypi:~$ cd ..
mmd@raspberrypi:/home$ ls
mmd
mmd@raspberrypi:/home$ cd ..
mmd@raspberrypi:/ $ ls
bin dev home lost+found mnt proc run srv tmp var
boot etc lib media opt root sbin sys usr
mmd@raspberrypi:/ $ sudo cat /dev/ttyAMA0
$GPRMC,,V,,,,,,,,,N*53
$GPVTG,,,,,,,,,N*30
$GPGGA,,,,,0,00,99.99,,,,,,,,,48
$GPGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99*30
$GPGLL,,,,,,,,,V,N*64
$GPRMC,,V,,,,,,,,,N*53
$GPVTG,,,,,,,,,N*30
$GPGGA,,,,,0,00,99.99,,,,,,,,,48
$GPGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99*30
$GPGLL,,,,,,,,,V,N*64
$GPRMC,,V,,,,,,,,,N*53
$GPVTG,,,,,,,,,N*30
$GPGGA,,,,,0,00,99.99,,,,,,,,,48
$GPGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99*30
$GPGLL,,,,,,,,,V,N*64
$GPRMC,,V,,,,,,,,,N*53
$GPVTG,,,,,,,,,N*30

```

تصویر ۴ مختصات خام دریافتی از سنسور GPS

مطابق بخش قبل، برای استفاده از مختصات دریافتی سنسور باید از کد پایتون کمک بگیریم، اما پیش از آن نیاز به چند تنظیم دیگر داریم، چون رزبری پای به طور پیش فرض از پورت سریال برای کنسول لاگین استفاده می‌کند و ما به این پورت برا یدریافت اطلاعات موقعیت مکانی نیاز داریم؛ پس باید کنسول لاگین را غیرفعال کنیم. در رزبری پای دو پورت سریال serial0 و serial1 وجود دارد که serial0 به GPIO 14 و GPIO 15 متصل است. برای مشاهده پورت‌های متصل به serial0 از دستور `ls -l /dev` استفاده می‌کنیم.

برای غیرفعال کردن کنسول از دستورات زیر استفاده می‌کنیم:

```

sudo systemctl stop serial-getty@ttyAMA0.service
sudo systemctl disable serial-getty@ttyAMA0.service

```

نهایتاً به کمک کد پایتون در فایل `gps.py` می‌توانیم اطلاعات موقعیت مکانی را خوانده و در بخش‌های بعدی از آن استفاده کنیم. اطلاعات ذکر شده در این بخش عمدتاً از [این لینک](#) گرفته شده است.

۳. تشخیص جهت تابش آفتاب به کمک مختصات

در نهایت، به کمک خروجی گرفته شده در بخش قبل می‌خواهیم زاویه تابش آفتاب را بیابیم. بدین منظور، علاوه بر اطلاعات مذکور به تاریخ و زمان نیز نیاز داریم. سپس به کمک کتابخانه [astropy](#) در پایتون می‌توانیم زاویه تابش آفتاب را بیابیم. قطعه کد لازم برای این بخش در فایل `gps.py` قابل مشاهده است. برای پیاده‌سازی این بخش از [این لینک](#) کمک گرفتیم.