

به نام خدا



دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



سیستم های نهفته بی درنگ

گزارش تمرین کامپیوتری اول

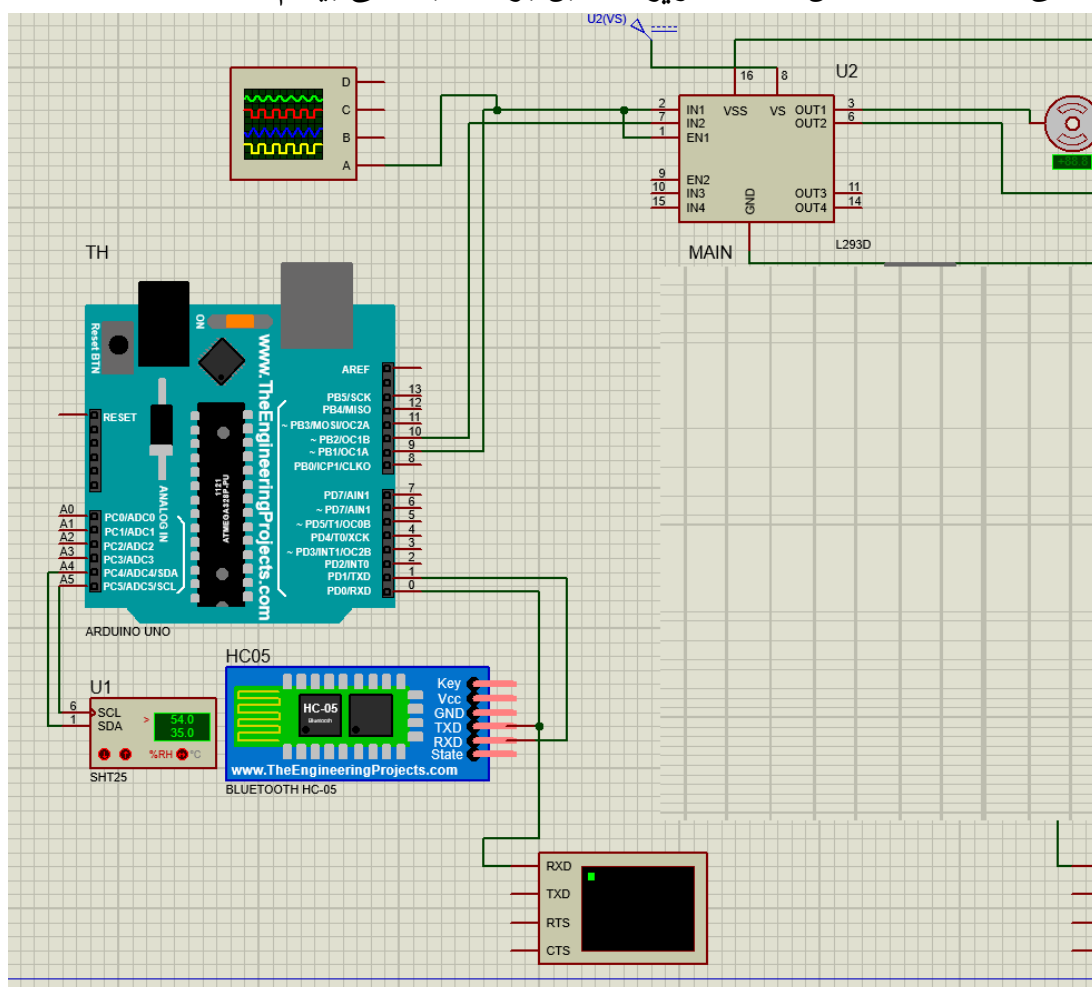
810197627	طنین زراعتی
810197539	امیرحسین عباسکوهی
810197515	آرش رسولی
810197486	کیاوش جمشیدی

مقدمه

در این تمرین، به پیاده سازی یک سیستم به منظور کنترل میزان رطوبت و آب رسانی یک گلدان پرداختیم که در آن با توجه به دما و رطوبت محیط یک موتور شروع به چرخش با سرعت مشخص شده توسط مدار می کند. این موتور نشانگر همان سیستم آب رسانی است. این مدار از دو نود اصلی TH و Main تشکیل شده است که توضیحات آن در ادامه خواهد آمد:

برد TH

این برد اطلاعات محیطی را به برد اصلی ارسال می کند. همچنین باید بر اساس اطلاعاتی که توسط برد اصلی برای آن ارسال می شود، سرعت و فرکانس موتور که همان مدلی از سیستم آب رسانی است را مشخص کند. تصویری از این برد در اینجا می بینیم:



این مدار شامل 05HC که برای ارتباط bluetooth است، یک برد Arduino Uno، موتور و 25SHT می باشد.

ماژول 25SHT

این ماژول یک سنسور با دقت بالا برای اندازه گیری دما و رطوبت است که برای انتقال داده ها سیگنال ها را به فرم I2C ارسال می کند. به همین دلیل این ماژول با دو خط SDA و SCL که به ترتیب برای ارسال دیتا و کلاک می باشد ارتباط برقرار می کند.

ماژول 05-HC

این ماژول برای ارتباط از طریق بلوتوث استفاده می شود. ارسال و دریافت بین هر دو برد Uno انجام می پذیرد. دیتا از طریق پورت 2 برد به پورت RXD ماژول ارسال می شود.

برد TH

اینجا همان برد Arduino Uno را داریم که کد اجرایی و محاسباتی بر روی آن قرار میگیرد و از طریق خروجی ها و ورودی ها با بقیه ماژول های اطراف خود ارتباط برقرار می کند.

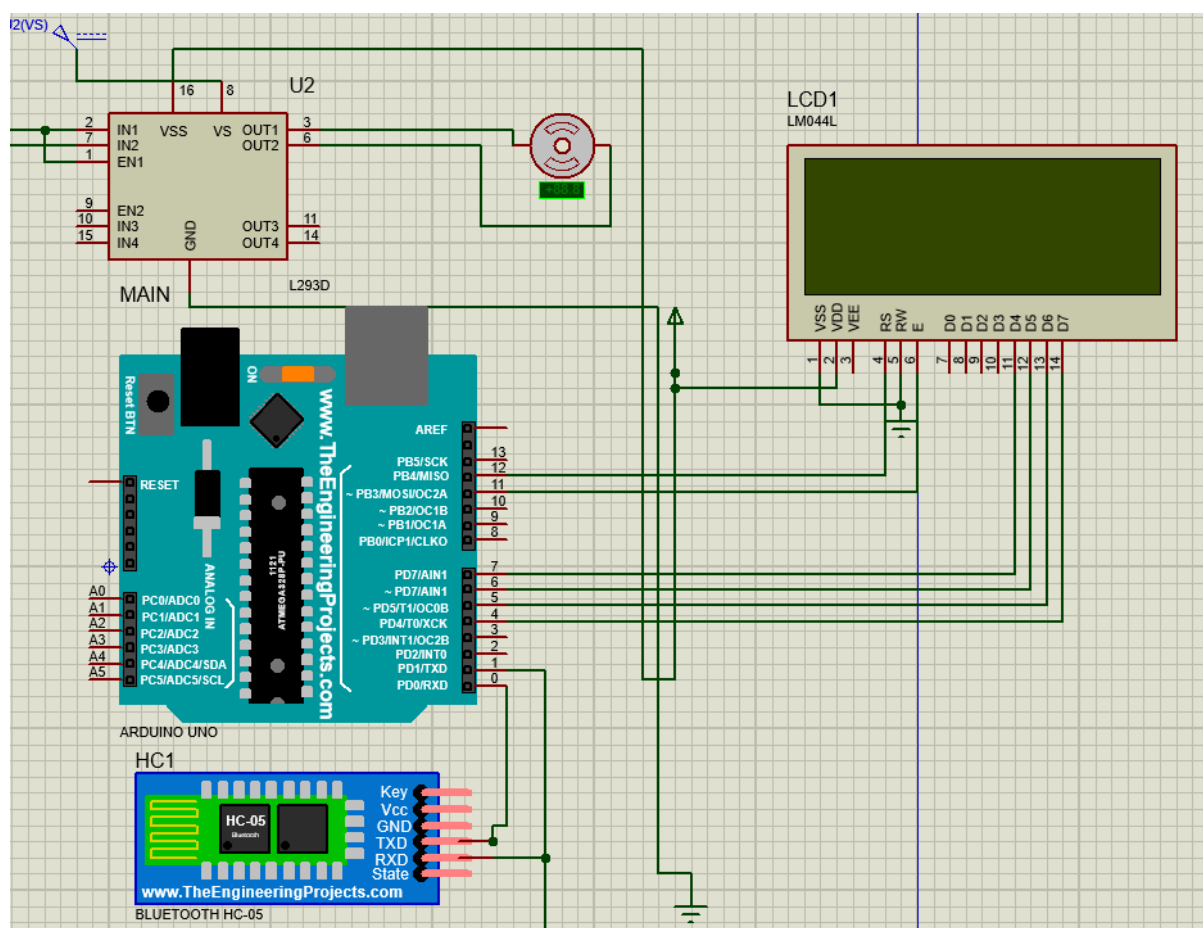
کد اجرایی برد TH

کدی که در برد آردوینو استفاده می شود در بخش code/TH-board/main.cpp قابل دسترسی است. این کد با استفاده از کتابخانه های خود برد، باید اطلاعات دمایی را از طریق سنسور بخواند، از طریق بلوتوث برای برد Main ارسال کند، اطلاعات و دستور را از برد Main از طریق آن بخواند و بر اساس اتفاق صورت گرفته، سرعت موتور را تنظیم کند که هر یک در تابع مشخص در حال انجام می باشد.

این روند در تابع loop صورت می پذیرد. برای تاخیر از تابع has_ticked استفاده می شود که برای 4000 میلی ثانیه تاخیر می اندازد. در صورت پروژه مقدار 5000 میلی ثانیه گفته شده که با توجه به تاخیر 500 میلی ثانیه ای برای دما و رطوبت 1000 میلی ثانیه کمتر در نظر گرفتیم.

برد Main

این برد وظیفه تصمیم گیری را دارد. در واقع باید اطلاعات دما و رطوبت را را برد TH دریافت کند، سپس بر اساس اطلاعات دریافت مشخص می کند موتور باید با چه سرعتی بچرخد و این اطلاعات برای برد TH ارسال میکند تا این برد موتور را بچرخاند.

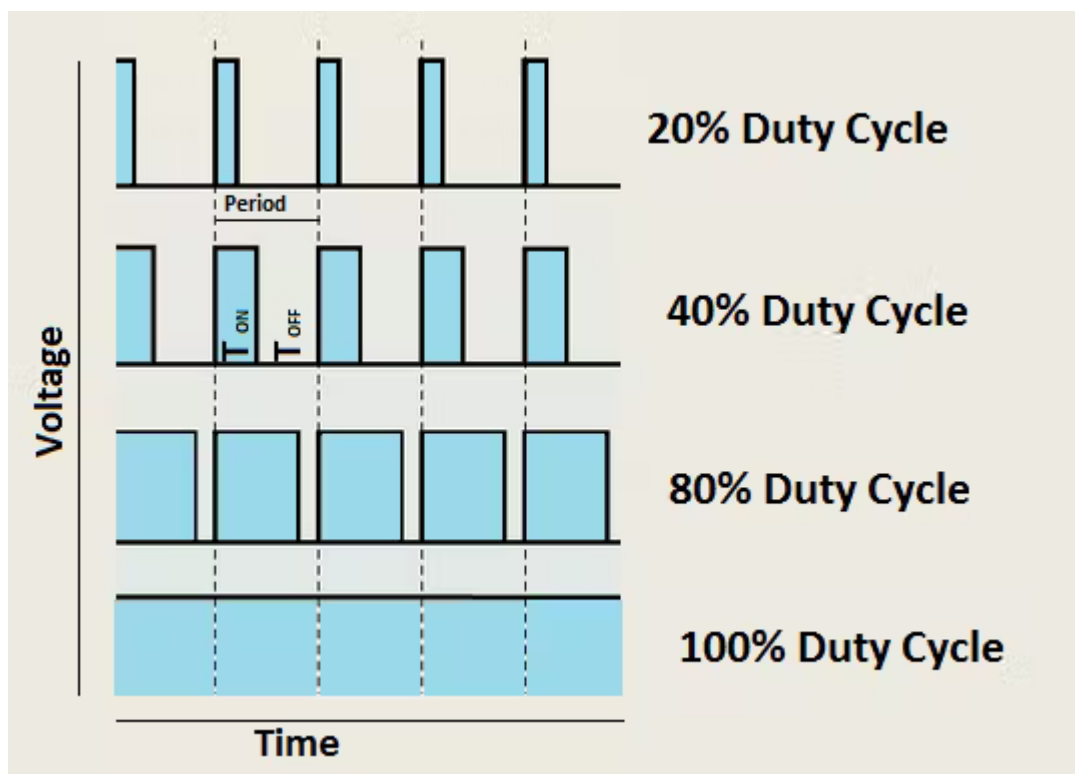


کد اجرایی این برد در `code/Main-board/main.cpp` وجود دارد. همچنین در این مدار ما LCD LM044L داریم که داده های دریافتی از برد TH را نشان میدهد و همچنین تصمیم اتخاذ شده بر اساس شرایط را نشان می دهد. اتصال این برد از طریق پین های مشخصی از LCD و Arduino Uno انجام می شود.

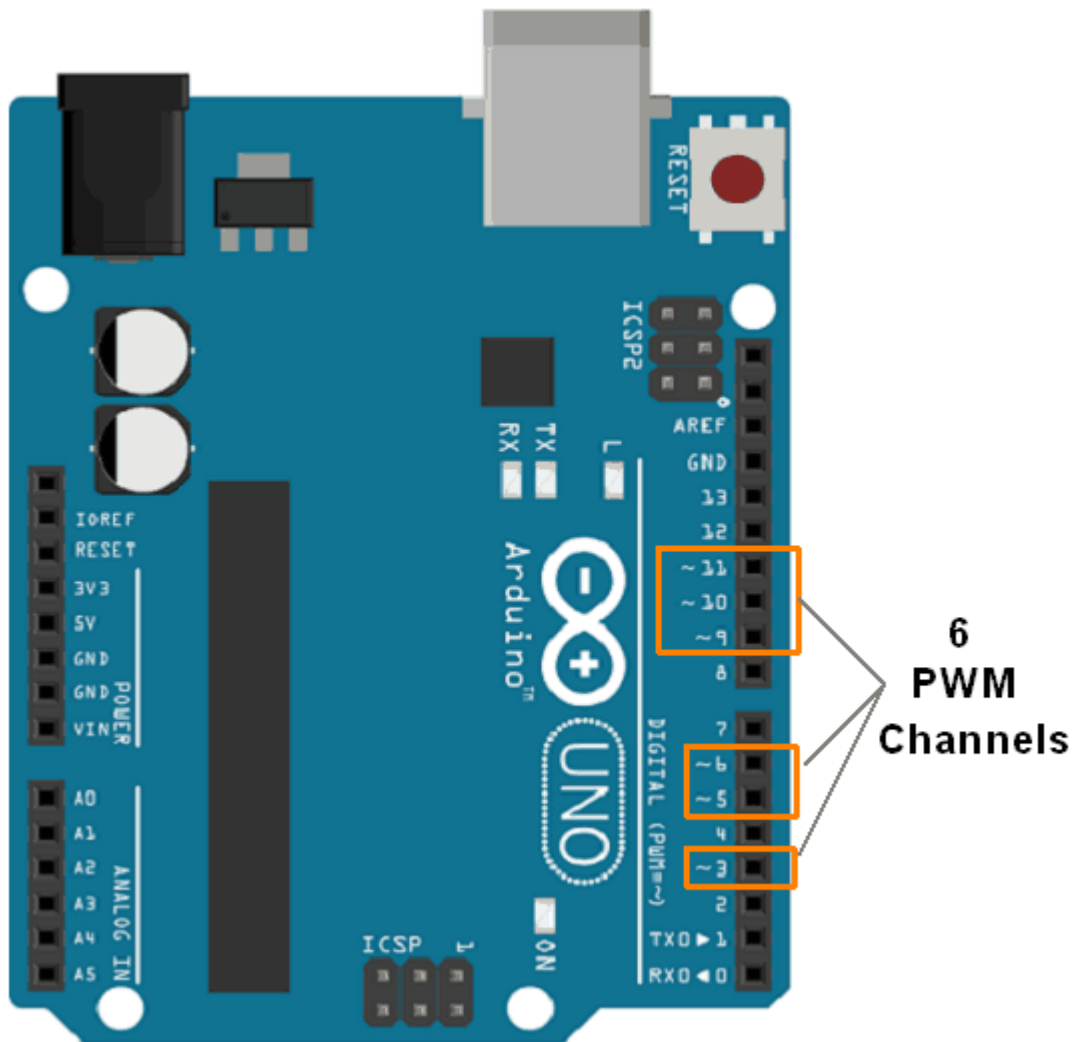
1. برای ارتباط bluetooth از امواج UHF استفاده می شود که بین فرکانس 2.402 تا 2.480 گیگاهرتز است. از طرفی همیشه در انتقال اطلاعات احتمال collision وجود دارد راه هایی برای جلوگیری و حل مشکل وجود دارد. یکی از این روش ها استفاده از مدل های frequency hopping spread spectrum یا FHSS است. برای مثال از مدل Adaptive frequency hopping یا AFH استفاده می کنیم که در آن باند فرکانسی به قطعه های کوچکتر تقسیم بندی می شود به سرعت بین کانال ها که چیزی حدود 1600 مرتبه در ثانیه است افزایش می یابد. فرضا اگر انرژی رو به پایان باشد به 40 کانال کوچکتر تقسیم می شود همچنین کانال های شلوغ یا با نویز زیاد شناسایی می شود و از انجام فعالیت توسط آن ها جلوگیری می شود.

2. بله از طریق آدرس ها می شود. برای آدرس های I2C دو حالت 7 بیتی و 10 بیتی وجود دارد (که 7 بیتی رایج تر است). از طرفی تمامی ماژول ها و تراشه ها این 7 بیت را دارند. این 7 بیت به ما میگوید که تا 128 دستگاه می توانند از طریق I2C اتصال برقرار کنند. البته علاوه بر این 7 بیت، یک بیت اضافه هم ارسال شده که برای مشخص کردن نوع عمل که خواندن و یا نوشتن است ارسال می شود. (7 بیت یاد شده در بخش اول قرار دارد یعنی MSB هستند).

3. PWM مخفف Pulse Width Modulation است و تکنیکی است که در کنترل روشنایی LED، کنترل سرعت موتور DC، کنترل سروو موتور یا جایی که باید خروجی آنالوگ را با وسایل دیجیتال دریافت کنید استفاده می شود.



در واقع برای مشخص کردن duty cycle موتور از آن استفاده می کنیم. یک سری پورت مشخص برای تعیین کردن آن در برد Arduino Uno وجود دارد که آن را در اینجا میبینیم:



ما از پورت های 9 و 11 استفاده کردیم.
برای تنظیم duty cycle از کد زیر استفاده می شود:

```
void handlePWM() {  
    analogWrite(dc_pinA, pwm_velocity);  
    digitalWrite(dc_pinB, 0);  
}
```

ولتاژ سر و ته این دو پین مشخص کننده duty cycle است. سرعت موتور از طرفی 256 است و بر اساس شرایط ذکر شده در صورت سوال ما ولتاژ pwm_velocity را مشخص میکنیم. اگر جای دو پین A و B در کد تغییر کند بر خلاف ساعتگرد، موتور به صورت منفی یا پادساعتگرد میچرخد.

