

گزارش تمرین اول درس سیستم‌های نهفته بی‌درنگ

سینا کمالی (۸۱۰۱۹۷۵۶۹)

علیرضا آقایی (۸۱۰۱۹۷۶۷۹)

دانشور امراللهی (۸۱۰۱۹۷۶۸۵)

مهیار کریمی (۸۱۰۱۹۷۶۹۰)

طراحی اولیه

پس از بررسی اولیه صورت تمرین و آشنایی با Proteus و Arduino IDE به سراغ طراحی مدار پروژه در Proteus می‌رویم.

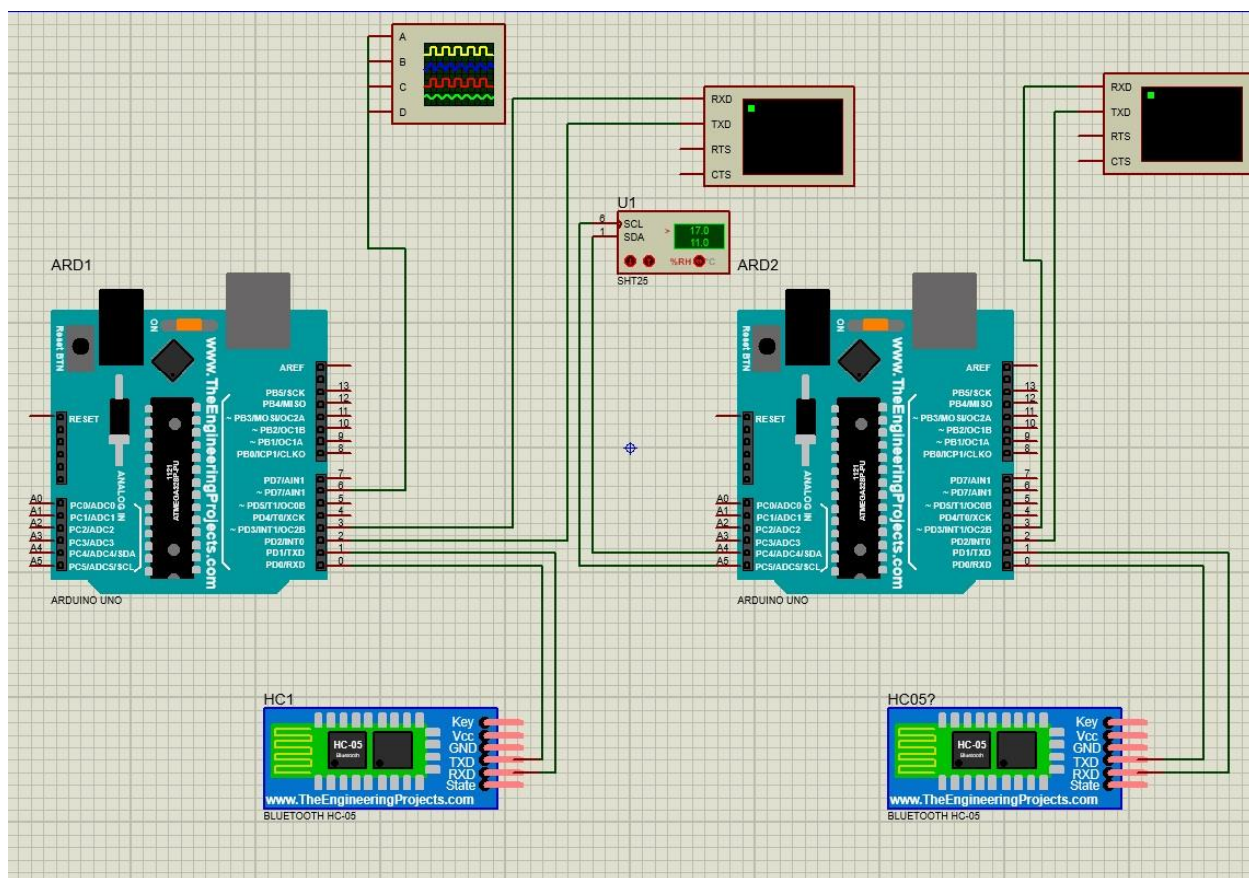
مدار پروژه شامل دو برد Arduino Uno می‌باشد که توسط یک ماژول بلوتوث به هم متصل می‌شوند. یکی از دو برد با نام master شناخته می‌شود و دیگری با نام sensor. هر دو برد دارای یک صفحه نمایش virtual terminal هستند که با استفاده از آن اطلاعات را به ما نشان می‌دهند.

در برد sensor ما متصل به یک سنسور شناساگر دما و رطوبت هستیم که اطلاعات را به صورت آنالوگ به ما می‌دهد و ما با تبدیل آن به عدد قابل خواندن دما و رطوبت را بدست می‌آوریم و پس از نمایش آن بر روی virtual terminal تلاش می‌کنیم تا این اطلاعات را برای master بفرستیم. برای انجام این کار نیاز به یک فرمتینگ مشخص داریم که هر دو برد به آن آگاه باشند. ما برای این کار از فرمت زیر استفاده کردیم:

H<humidity%>C<temperature>!

پس از H مقدار رطوبت و پس از C مقدار دما می‌آید و در نهایت یک علامت تعجب به معنای انتهای پیغام گذاشته می‌شود.

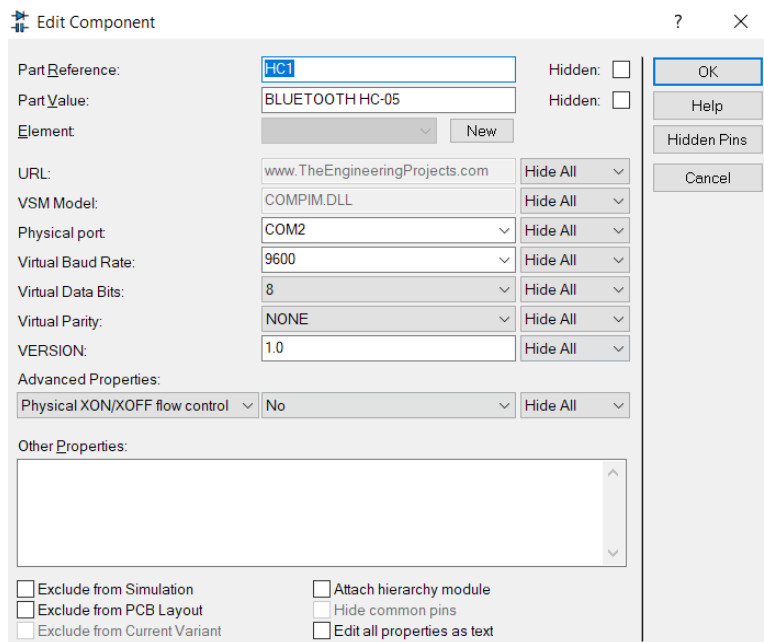
در برد master هم برای نمایش سیگنال pwm که موتور ما را به کار می‌اندازد، از یک virtual oscilloscope استفاده کرده ایم.



تصویر 1: تصویری از طراحی کلی مدارهای پروژه، برد master در چپ و برد sensor در راست تصویر قرار دارند.

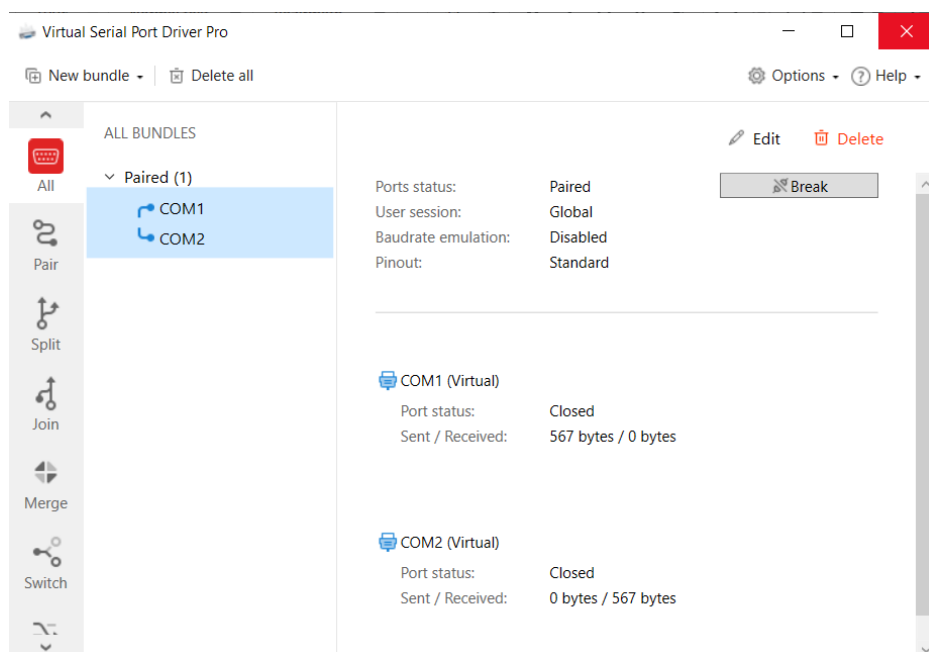
بلوتوث و اتصال مازول ها

برای اتصال دو برد Arduino از دو ماژول بلوتوث استفاده میکنیم. این ماژول ها برای اتصال به یکدیگر ابتدا باید بر روی دو com متفاوت configure شوند. (یکی برای ارسال دیتا از master به sensor و یکی برای برعکس آن)



تصویر 1: تصویری از صفحه configure کردن ماژول بلوتوث در proteus.

پس از configure کردن هر دو ماژول بلوتوث، با استفاده از نرم افزار virtual serial bus driver آنها را با یکدیگر pair میکنیم و کار ما تمام است! این دو ماژول به همین سادگی با یکدیگر pair شدند و می‌توانند به هم دیتا ارسال کنند.

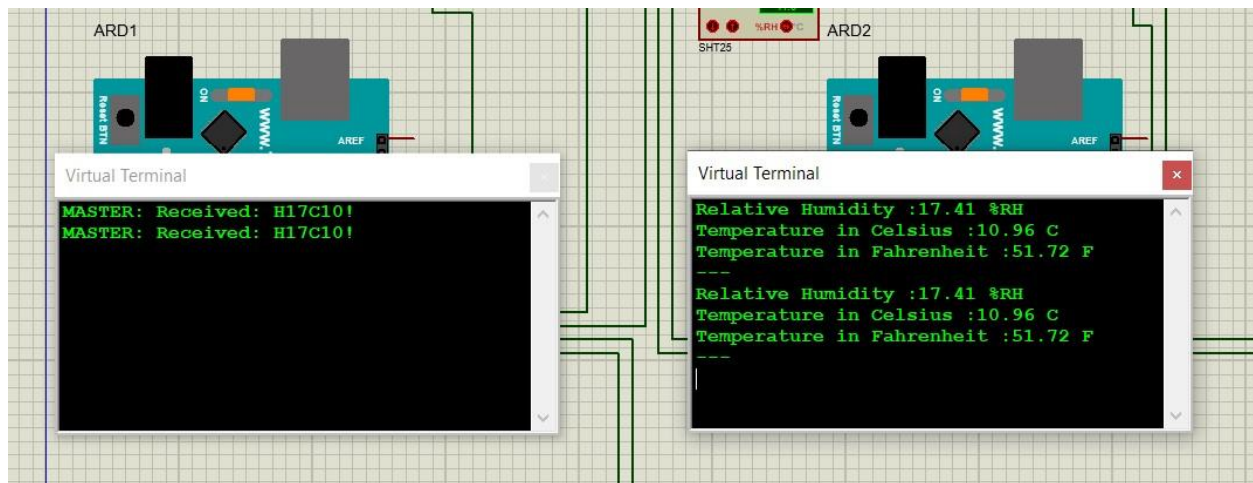


تصویر 1: تصویری از نرم افزار Virtual Serial Port Driver و pair شدن COM1 و COM2.

نتایج شبیه سازی ها

در ادامه به بررسی نتایج شبیه سازی ها می‌پردازیم. این بررسی را در دو بخش مربوط به خوانده شدن و انتقال دیتا و بخش واکنش درست نشان دادن به وضعیت سنسورها بررسی می‌کنیم.

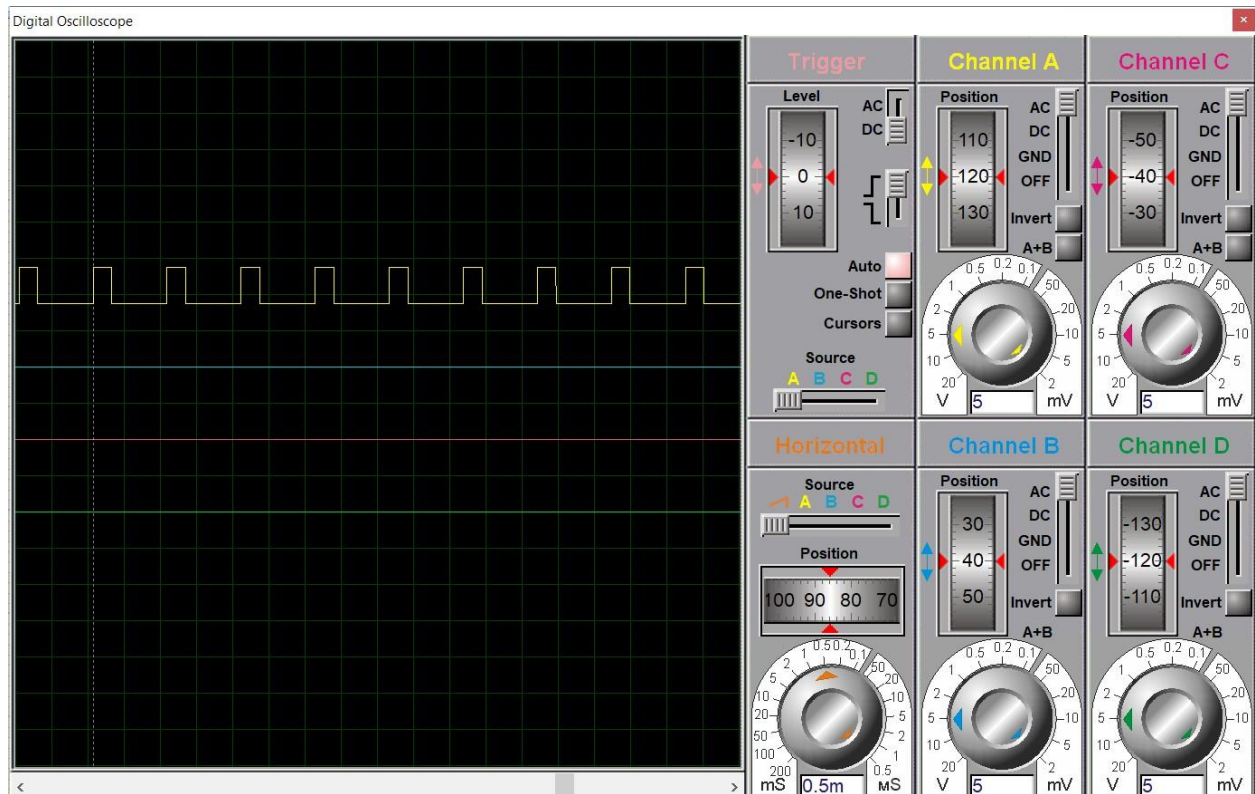
1- بررسی درست خوانده شدن و انتقال دیتا



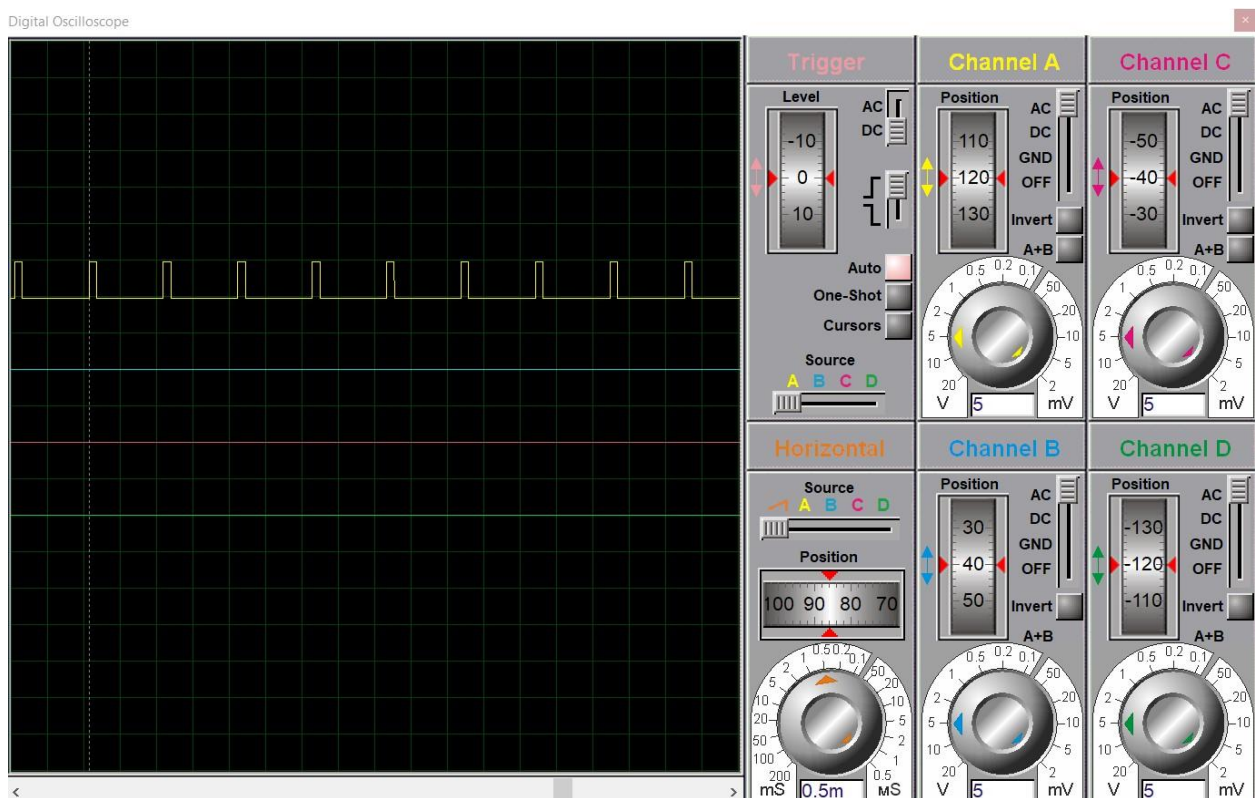
پس از اطمینان حاصل کردن از درستی کارکرد این بخش، به سراغ بررسی بخش دوم می‌رویم.

2- بررسی واکنش master به دیتا

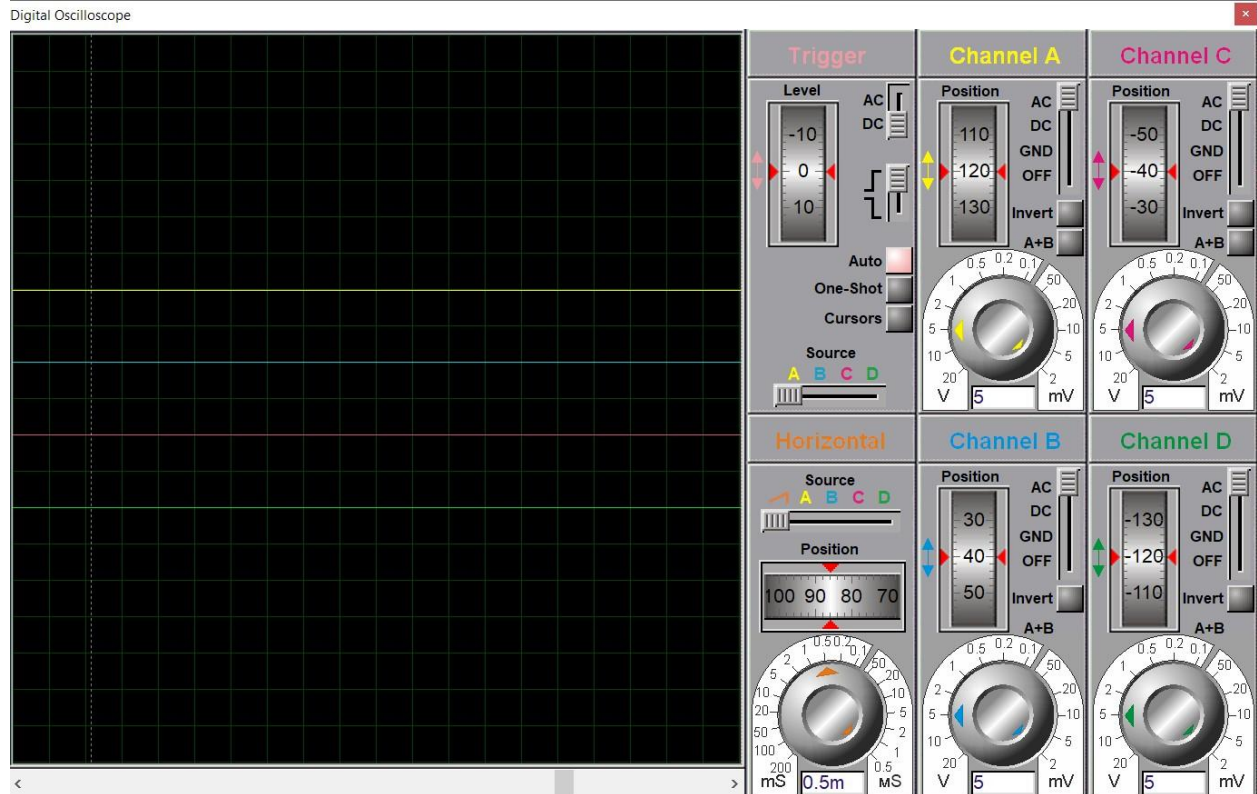
در این بخش باید مطمئن شویم که برد مستر در سه حالت عدم آبیاری، آبیاری با 10% duty-cycle و آبیاری با 25% duty-cycle درست عمل می‌کند.



تصویر 1: تصویری از duty-cycle 25% که در حالت humidity زیر 20% اتفاق می افتد.



تصویر 1: تصویری از duty-cycle 10% که در صورتی که humidity میان 20 تا 50 درصد باشد و دما بالای 25 درجه باشد اتفاق می افتد.



تصویر 1: تصویری از خاموش بودن موتور که در تمامی حالات باقی مانده اتفاق می افتد.

با توجه به نتایج حاصل شده در این بخش مطمئن میشویم که ماژول ما در حال کار کردن به نحو احسن میباشد و پروژه ما آمادگی تجاری سازی است!

پرسش‌ها

1. پروتکل ارتباطی بلوتوث در محدوده‌ی فرکانسی ۲۰۴۰۰۰ تا ۲۰۴۸۳۵ گیگاهرتز کار می‌کند [1]. برای جلوگیری از تداخل داده‌های ارسالی توسط چند دستگاه بلوتوث که در یک منطقه هستند، از مکانیزم AFH¹ استفاده می‌شود. در این روش، پس از اتصال دو دستگاه، در هر رویداد اتصال²، دو دستگاه یک کانال رادیویی را به صورت قطعی³ انتخاب می‌کنند و انتقال در این کانال صورت می‌گیرد. محدوده‌ی فرکانسی بلوتوث برای همین منظور به تعداد مشخصی کانال تقسیم شده است. به کمک مکانیزم AFH، احتمال تداخل در ارتباط چند دستگاه بلوتوث که در یک محدوده‌ی کوچک همزمان مشغول به کار هستند بسیار کاهش می‌یابد [2].
2. در پروتکل I²C، هر دو دستگاهی که در اتصال شرکت کرده‌اند بایستی آدرس‌های متمایزی داشته باشند. در صورت برقرار بودن شرط بالا، می‌توانیم از پورت‌های I²C تعبیه شده روی بُرد Arduino برای اتصال چند دستگاه متفاوت استفاده کنیم. ارتباط برد با هر یک از دستگاه‌های متصل شده به کمک آدرس دستگاه امکان‌پذیر است [3].
3. یکی از روش‌های کنترل سرعت در موتورهای DC، روش PWM⁴ است. در این روش، پالس‌های متوالی روشن / خاموش به موتور ارسال می‌شوند. میانگین توان منتقل شده به موتور نیز توسط Duty Cycle پالس‌های ارسال شده کنترل می‌شود [4].

مراجع

- [1] <https://en.wikipedia.org/wiki/Bluetooth#Implementation>
- [2] <https://www.bluetooth.com/blog/how-bluetooth-technology-uses-adaptive-...>
- [3] <https://learn.adafruit.com/i2c-addresses>
- [4] <https://www.electronics-tutorials.ws/blog/pulse-width-modulation.html>

¹ Adaptive frequency hopping

² Connection event

³ deterministic

⁴ Pulse width modulation