



دانشگاه تهران
دانشکده‌ی مهندسی برق و کامپیوتر



سیستم‌های نهفته‌ی بی‌درنگ

تمرین دوم
آشنایی با پروتکل‌های ارتباطی و سنسورها

طراحان
گروه سخت افزار

اساتید
دکتر کارگهی و دکتر مدرسی

۱۳۹۷-۱۳۹۸

مقدمه

در این تمرین قصد داریم با رابط‌های (interface) آردوینو آشنا شویم و کار با سنسورهای مختلف را یاد بگیریم. به طور کلی شما در این تمرین باید با استفاده از سه ماژول آردوینو، یک شبکه شامل یک ماژول مرکزی و دو ماژول دیگر که هر کدام تعدادی سنسور روی خود دارند را ایجاد کنید. هر کدام از این ماژول‌ها با ماژول مرکزی در ارتباطند و داده‌هایی که از سنسورهای خود دریافت می‌کنند را به آن ارسال می‌کنند. وظیفه‌ی ماژول مرکزی تنها نمایش اطلاعات دریافت شده روی یک LCD می‌باشد.

برای شبیه‌سازی همانند تمرین قبل از Proteus استفاده می‌کنیم اما این بار بجای استفاده از Arduino Studio از PlatformIO که یک اکوسیستم کامل برای توسعه‌ی پروژه‌های IoT در اختیار ما قرار می‌دهد استفاده می‌کنیم.

هدف از انجام این تمرین یادگیری کار با رابط‌های آردوینو، آشنایی با پروتکل‌های سریال، I2C، سنسورها و یادگیری استفاده از آنها می‌باشد.

پیش‌زمینه

– ارتباط سریال:

یکی از مواردی که در این تمرین با آن آشنا می‌شوید ارتباط سریال است. ارتباط سریال بر خلاف ارتباط موازی، از یک سیم برای ارسال داده‌ها استفاده می‌کند و داده‌ها در فاصله‌های زمانی پشت سر هم روی خط ارسال می‌شوند. امروزه اکثر ارتباطات راه دور برای کاهش هزینه و کاهش نویز به صورت سریال پیاده‌سازی می‌شود. در این پروتکل چند موضوع مهم باید مورد توجه قرار بگیرند:

• نحوه هماهنگی سرعت ارسال و دریافت داده‌ها بین فرستنده و گیرنده

از آنجایی که انتقال داده به صورت بیت به بیت روی یک سیم انجام می‌شود؛ لازم است سرعت ارسال داده توسط فرستنده با سرعت دریافت داده در گیرنده برابر باشد. در نتیجه باید در ابتدای برقراری ارتباط، سرعت ارسال بین طرفین توافق شود. این توافق می‌تواند به صورت همگام (USART¹) یا غیرهمگام (UART²) انجام شود.

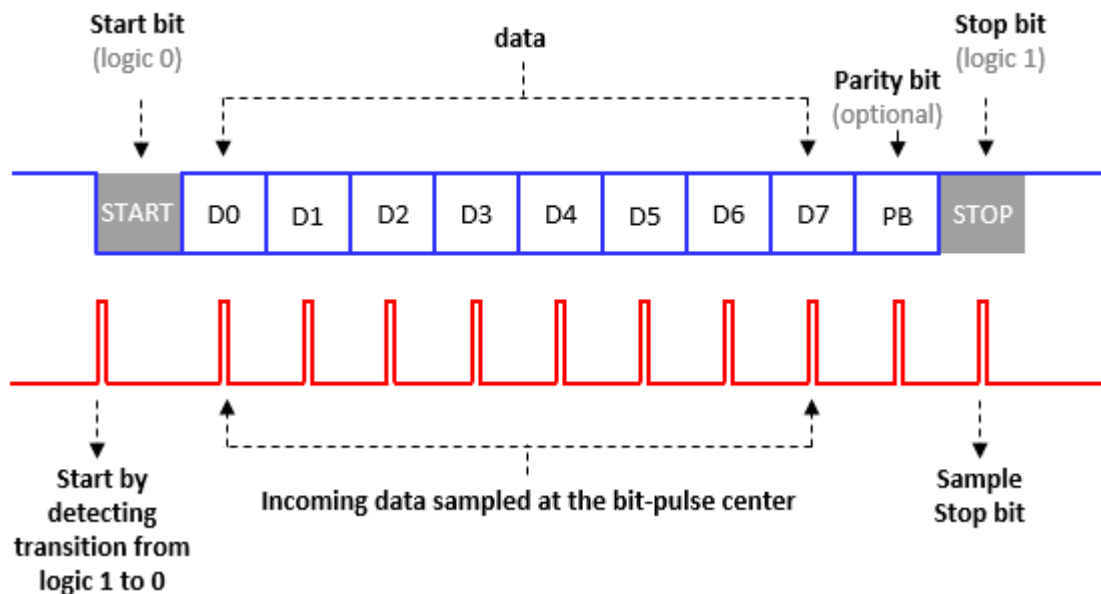
¹ Universal Synchronous Serial Receiver And Transmitter

² Universal Asynchronous Serial Receiver And Transmitter

- نحوه framing داده‌ها

چیدمان، طول داده‌ی رد و بدل شده و نحوه‌ی آغاز و پایان ارسال داده باید از قبل مشخص باشد.

برای درک بهتر، در ارتباط سریال غیرهمگام، در حالت بیکار (idle) خط ارتباطی حاوی بیت ۱ است. برای شروع ارسال داده، یک بیت صفر بر روی خط قرار داده می‌شود. گیرنده با دریافت این بیت، بیت‌های بعدی را به عنوان داده در نظر می‌گیرد. فرستنده هم پس از ارسال بیت صفر، به ترتیب از بیت کم ارزش داده شروع می‌کند، تعدادی بیت (حداکثر ۸) ارسال می‌کند و در نهایت دو بیت ۱ ارسال می‌کند که نشان توقف یا پایان است.



چیدمان سیگنال‌ها در ارتباط سریال

- مکانیزم کشف خطا

به صورت استاندارد، برای کشف خطا از یک یا دو بیت parity استفاده می‌شود که در طرف گیرنده با دریافت آن می‌تواند از وجود خطا در خط با خبر شود.

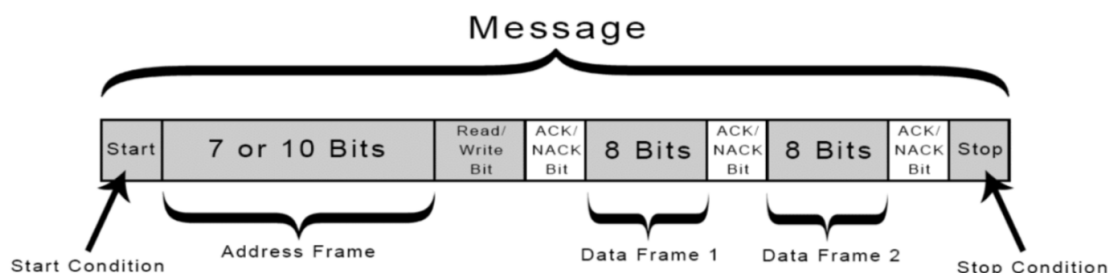
برای اطلاعات بیشتر می‌توانید از این [لینک](#) که به صورت کامل و با جزئیات بیشتر این ارتباط را شرح داده است، استفاده کنید.

- ارتباط I2C:

در ادامه به توضیح یک مدل از ارتباط سریال به نام I2C³ یا مدار مجتمع یکپارچه می‌پردازیم. این پروتکل در قطعاتی مانند شتاب‌سنج و ژيروسکوپ استفاده می‌شود؛ در نتیجه یادگیری آن اهمیت زیادی برای پروژه‌های پایانی شما خواهد داشت.

در ارتباط I2C از دو خط به نام‌های SDA⁴ و SCL⁵ استفاده می‌شود. خط SCL حامل سیگنال clock است و خط SDA برای ارسال و دریافت داده‌ها به کار می‌رود. در این پروتکل خط‌های SDA و SCL همگام (synchronous) هستند و خط کلاک توسط یکی از طرفین که master نامیده می‌شود تنظیم می‌شود.

در پروتکل I2C، داده‌های ارسال شده از بخش‌های مختلفی تشکیل شده است. در ابتدای هر بسته، برای شروع انتقال، داده‌ی خط SDA از یک به صفر تغییر می‌کند و بعد از آن خط SCL هم همین تغییر را اعمال می‌کند تا نشانه‌ی شروع ارسال باشد. در قسمت بعدی، آدرس slave قرار داده می‌شود. با استفاده از این بخش، می‌توان برای ارسال داده به slave‌های مختلف تنها از یک خط انتقال استفاده کرد. بیت بعدی نشان دهنده‌ی این است که آیا مازول master می‌خواهد داده‌ای را ارسال کند و یا داده‌ای از slave دریافت کند. مکانیزم تشخیص خطا در این پروتکل به این صورت است که بعد از بیت‌های کنترلی، و بعد از هر یک از frame‌های داده که در شکل زیر مشاهده می‌کنید؛ یک بیت برای acknowledge وجود دارد. و گیرنده با یک کردن آن، دریافت داده‌ها را تایید خواهد کرد. در نهایت برای نمایش پایان انتقال، خط SCL از صفر به یک و SDA هم به همین صورت تغییر می‌کنند و نشان‌دهنده‌ی پایان بسته‌ی پیام است.



چیدمان سیگنال‌ها در ارتباط I2C

³Inter Integrated Circuit

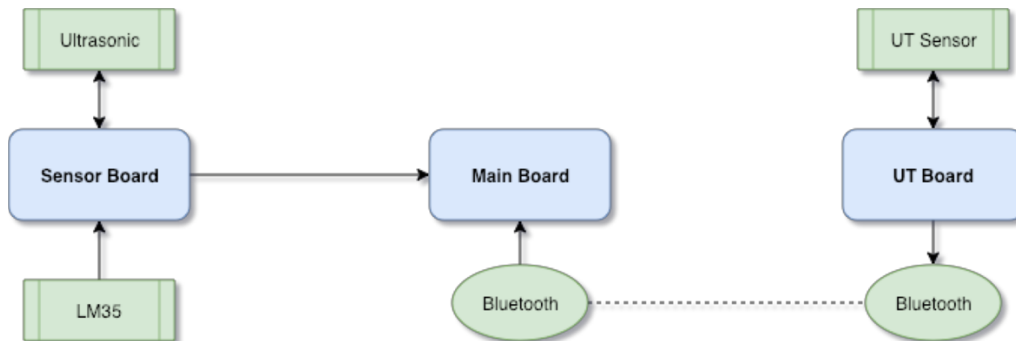
⁴Serial Data

⁵Serial Clock

شرح تمرین

در این تمرین قرار است یک شبکه از سه آردوینو را شبیه‌سازی کنید. شرح هرکدام از ماژول‌ها به صورت زیر است:

- UT-Board:** این ماژول شامل یک برد آردوینو است که از یک طرف به یک سنسور و از طرف دیگر به گرهی مرکزی متصل می‌باشد. این ماژول از طریق ارتباط I2C به یک سنسور (UT Sensor) که در اختیار شما داده شده است متصل می‌شود و اطلاعات سنسور را بعد از پردازش از طریق ارتباط سریال به ماژول بلوتوث ارسال می‌کند. این ماژول بلوتوث در ادامه اطلاعات را به ماژول بلوتوث متصل به ماژول مرکزی ارسال خواهد کرد.
- Sensor-Board:** این گره نیز مانند گرهی قبلی ارتباط بین سنسورها و گرهی مرکزی را برقرار می‌کند، این ماژول به یک سنسور فاصله‌سنج ((Ultrasonic HC-SR04 و یک سنسور دما ((LM35 متصل می‌باشد و باید اطلاعات این دو سنسور را همزمان دریافت کند و با استفاده از ارتباط سریال برای ماژول اصلی ارسال کند. (ارتباط این ماژول با ماژول مرکزی از طریق یک سیم برقرار می‌شود)
- Main-Board:** این ماژول که ماژول مرکزی ماست، به یک LCD مجهز است و در هر خط از آن، مقادیر دریافتی از یکی از سه سنسور گفته شده در بالا را نمایش می‌دهد. (دقت کنید که این ماژول نیاز به یک ماژول بلوتوث هم دارد که داده‌ها را از ماژول اول دریافت کند.)



نمایش جایگاه و اتصالات منطقی گره‌ها

در پیاده‌سازی این تمرین، به نکات زیر دقت کنید:

- قبل از شروع پروژه حتما نحوه کارکرد پروتکل I2C و Serial را به صورت دقیق مطالعه کنید. (تسلط شما بر این دو پروتکل در زمان تحویل پروژه سنجیده می‌شود)
- تمرکز این پروژه در کنار یادگیری و استفاده از پروتکل‌های ارتباطی، بر روی کد زدن صحیح به‌عنوان یک مهندس کامپیوتر برای این اجزا می‌باشد، در نتیجه بخشی از نمره این پروژه به تمیزی کد (استفاده از نام‌گذاری قابل فهم، استفاده درست از توابع و ...) تعلق دارد.
- استفاده از [Git](#) برای این پروژه اجباری می‌باشد و شما به‌جای آپلود پروژه ملزم به گذاشتن hash آخرین کامیت خود در محل آپلود می‌باشید. (حتما پوشه‌ی .gitignore را در .vscode قرار دهید)
- با توجه به سنگین‌تر بودن این تمرین نسبت به تمرین قبل، پیشنهاد می‌شود قسمت‌های مختلف این تمرین بین افراد گروه تقسیم شود. البته همه‌ی افراد باید تسلط کافی به مباحث تمامی بخش‌های پروژه داشته باشند.

سوالات

۱. پروتکل I2C قابلیت اتصال multi masters – multi slaves را دارد. توضیح دهید مشکلات پیش روی این حالت چیست؟ در این پروتکل از چه مکانیزم داورای استفاده می‌شود و چگونه مشکل تداخل را حل کرده است؟
۲. مراحل ارتباط دو دستگاه به یکدیگر از طریق این پروتکل (I2C) را بیان کنید.
۳. در اتصال UART (که معروف‌ترین پروتکل ارتباط سریال می‌باشد)، نقش پارامتر baud rate چیست؟ و به چه دلیل این پارامتر اهمیت پیدا می‌کند؟
۴. هرکدام از روش‌های I2C و UART مزایا و معایبی دارند، این دو را با هم مقایسه کنید و برتری‌های هر یک را برشمارید.
۵. در مورد سریال نرم‌افزاری و نحوه‌ی کار آن توضیح مختصری دهید. محدودیت‌های آن نسبت به سریال سخت‌افزاری چیست؟
۶. نحوه کارکرد سنسور فاصله‌سنج گفته شده را توضیح دهید. دقت این سنسور در چه حدودی است؟ حداقل فاصله لازم برای دریافت داده‌ی صحیح از این سنسور و بیشترین فاصله‌ای که می‌تواند پوشش دهد تقریباً چقدر است؟
۷. اگر در پروژه‌ای نیاز به استفاده از دو مازول فاصله‌سنج داشته باشیم چه مشکلی می‌تواند به وجود بیاید؟
۸. اگر نیاز به اتصال چند سنسور مختلف که با پروتکل I2C استفاده می‌کنند بشویم آیا در پیاده‌سازی با مشکلی روبرو می‌شویم؟ حال اگر دو سنسور از یک نوع (مثلا دو تا سنسور 6050MPU) داشته باشیم چطور؟ اگر سه تا از این سنسور داشته باشیم چطور؟ (جواب به ترتیب خیر، خیر، بله است! توضیح دهید چرا)
۹. طراحی مفهومی این تمرین را مانند نمونه‌ای که در تمرین اول دیدید رسم کنید.

نرم‌افزارهای مورد نیاز

ابتدا به نحوی از شر Arduino Studio خلاص شوید!

برای شروع کار، با استفاده از دستورالعمل‌های وبسایت [PlatformIO](https://platformio.io) اقدام به نصب این برنامه کنید. (این برنامه به صورت یک plugin برای ادیتور VSCode در دسترس می‌باشد.)

برای هر یک از سه ماژول، یک پوشه ایجاد کنید و آن‌ها را در پوشه‌ی اصلی پروژه‌ی خود قرار دهید. git شما باید در این پوشه باشد تا کدهای هر سه ماژول را در بر بگیرد.

نکته: برای ایجاد کردن ساختار اولیه هر یک از ماژول‌ها می‌توانید از دستور زیر استفاده کنید:

```
platformio init --board uno
```

جزئیات پیاده‌سازی

UT-Board

این ماژول با استفاده از پروتکل I2C به UT Sensor متصل است، برای این‌کار تنها کافیسیت دو پورت SCL و SDA را به پورت‌های متناظر سنسور متصل کنید (در آردوینو Uno پورت‌های SCL و SDA با پورت‌های ورودی آنالوگ مشترک است)

در ابتدا، از نحوه کار با پروتکل I2C در آردوینو که توسط کتابخانه Wire انجام می‌گیرد آشنا شوید. قبل از هرکاری نیاز است تا با توجه به datasheet سنسور، اقدام به روشن کردن آن کنید. بعد از آن سنسور به کار می‌افتد و می‌توانید از دیتای سنسور استفاده کنید.

سنسور داده شده در هر لحظه دو مقدار X و Y را در اختیار شما قرار می‌دهد. دیتای این سنسور به صورت یک عدد float می‌باشد که در ۴ بایت ذخیره شده است. برای گرفتن هر کدام از این مقادیر، نیاز است تا آدرس مربوط به ثبات مورد نظر را برای سنسور ارسال کنید و بعد از آن یک بایت را از سنسور درخواست کنید. پس برای گرفتن هر یک از مقادیر به چهار درخواست نیاز دارید. با کنار هم گذاشتن داده‌ی مربوط به سنسور می‌توانید به مقادیر X و Y برسید!

همچنین می‌توانید با اتصال یک Virtual Terminal به پورت debug سنسور از درست بودن کدهای خود اطمینان حاصل کنید. (پورت دیباگ در واقع داده‌ی سنسور را به صورت سریال خروجی می‌دهد)

بعد از گرفتن دو مقدار X و Y باید هر ۱۰۰ میلی‌ثانیه یک‌بار سیگما $Y + 2 * X^2$ را محاسبه کنید و به صورت سریال خروجی دهید و در نهایت خروجی سریال آردوینو (TX) را به ورودی سریال بلوتوث (RX) متصل کنید.

Sensor-Board

این ماژول باید به سنسورهای دما و فاصله متصل شود. از لیست کتابخانه های Proteus دو ماژول 04SRF و 35LM را به پروژه اضافه کنید. با توجه به دیتاشیت های موجود در اینترنت این دو سنسور را به آردوینو متصل کنید.

برای استفاده از این سنسورها می توانید به صورت مستقیم با ورودی ها کار کنید تا داده های آن ها را به دست آورید؛ اما پیشنهاد اکید ما این است که از کتابخانه های در دسترس برای هر کدام استفاده کنید. برای این کار می توانید از قسمت Quick Access -> Libraries -> Registry و جستجوی نام کتابخانه، اقدام به نصب کتابخانه مورد نظر خود کنید. (امکان دارد برای سنسورها چند کتابخانه نوشته شده باشد؛ با توجه به تعداد دانه ها و یا نمونه کدها بهترین را انتخاب کنید)

می توانید با ترکیب نمونه کدهای مربوط به سنسورها، دیتای دو سنسور را گرفته و به شکلی مشخص (مثلا جدا کردن آن ها با یک حرف مناسب) از طریق پروتکل سریال برای ماژول اصلی ارسال کنید.

Main-Board

ماژول اصلی تنها وظیفه ی گرفتن داده های ارسال شده از دو ماژول و نمایش آن ها روی یک LCD را دارد. برای شروع ماژول LM041L با اندازه ی 4x20 را از لیست کتابخانه های Proteus به پروژه اضافه کنید. با استفاده از مستندات موجود در اینترنت، LCD را به آردوینو متصل کنید. (پایه ی En را حتما به GND متصل کنید)

کتابخانه ی LiquidCrystal را مانند قسمت قبلی نصب کنید. با استفاده از تکه کدی از درستی اتصال خود مطمئن شوید. حال با اتصال یک ماژول بلوتوث به ورودی سریال آردوینو، دیتای خروجی UT-Board را دریافت و در اولین خط LCD نمایش دهید، حال باید داده های مربوط به ماژول دوم را نیز دریافت کنید. اگر دقت کنید آردوینو تنها یک پورت RX دارد! برای اینکار می توانید از کتابخانه SoftwareSerial استفاده کنید. این کتابخانه دوتا از پایه های دیجیتال آردوینو را به ورودی و خروجی مجازی سریال تبدیل می کند، با استفاده از این کتابخانه یک پورت RX دیگر به آردوینو اضافه کنید و بعد از گرفتن و جدا کردن دو نوع داده داده ی مربوط به سنسور UltraSonic و 35LM را به ترتیب در خط دوم و سوم LCD چاپ کنید.

نکته ۱: کتابخانه SoftwareSerial کتابخانه پایه ای برای این کار است. می توانید از کتابخانه های جایگزین مانند [AltSoftSerial](#) که امکانات بیشتری را در اختیار شما قرار می دهند استفاده کنید.

نکته ۲: زمانی که مقدار سریال را می خوانید امکان دارد در میانه ی ارسال داده بوده باشد. مثلا فرض کنید ماژول اول برای شما عدد ۱۲۳۴ را ارسال کند ولی شما زمانی داده را می خوانید که داده به صورت کامل خوانده نشده باشد و تنها مقدار ۱۲ را دریافت کنید! این مشکل را در نظر بگیرید و آن را رفع کنید. (راهکار شما نباید به صورت blocking باشد - به بافر فکر کنید)

در پیاده سازی ارتباطات بین ماژولها و سنسورها به نکات زیر توجه داشته باشید:

- **Sensor-Board** باید دو نوع داده را برای ماژول مرکزی ارسال کند. برای این کار پروتکل مشخصی وجود ندارد. می‌توانید با استفاده از خلاقیت خود راهکاری برای این کار پیشنهاد دهید. برای مثال یک ایده می‌تواند استفاده از **delimiter** برای جداسازی انواع داده باشد.
- ارتباط بلوتوث تفاوتی با ارتباط سریال توسط سیم ندارد. در واقع ارسال بی‌سیم داده‌ها در لایه‌های پایین‌تر مدیریت شده و شما تفاوتی بین این دو انتقال حس نخواهید کرد. (برای ارسال داده‌ها بین دو آردوینو از طریق بلوتوث می‌توانید از [این لینک](#) استفاده کنید.)
- ماژول **UT Sensor** داده‌شده به شما در هر لحظه دو مقدار **X** و **Y** را از محیط محاسبه می‌کند و در اختیار شما قرار می‌دهد، اطلاعات فنی (**datasheet**) این ماژول در ادامه آورده شده است. برای استفاده از داده‌های این سنسور ابتدا نیاز است تا فرستادن ۱ روی رجیستر روشن و خاموش کردن، اقدام به روشن کردن ماژول کنید. بعد از روشن کردن سنسور در هر بار با درخواست رجیسترهای مربوط به متغیرهای **X** و **Y** و کنار هم گذاشتن مقادیر به داده‌ی اصلی برسید.
- ارتباط دو ماژول بلوتوث داده شده در محیط شبیه‌سازی به صورت مجازی برقرار می‌شود. برای اینکار روی این ماژولها دوبار کلیک کنید و **physical port** یکی از ماژولها را برابر **3COM** و دیگری را برابر **4COM** قرار دهید. سپس با استفاده از یکی از برنامه‌های ساخت ارتباط مجازی مثلا **com0com** این دو پورت را به هم متصل کنید)
- خواندن اطلاعات مربوط از دو ماژول نباید به صورت **blocking** باشد. چراکه ماژول اول تقریباً هر ۱۰۰ میلی‌ثانیه برای ما دیتا می‌فرستد ولی ماژول دوم سرعت بیشتری می‌تواند داشته باشد و برای نمایش دیتای مربوط به ماژول دوم نباید منتظر ماژول اول بمانیم. (برای پیاده‌سازی این قسمت متد **available** سریال را مطالعه کنید)

نکات مهم

- این تمرین باید در قالب گروه‌های سه نفره انجام شود.
- تمرین تحویل حضوری خواهد داشت.
- برای ما مهم است که حاصل کار خود را تحویل دهید به همین دلیل به شدت با تقلب برخورد خواهد شد و به طرفین نمره ۱۰۰- تعلق خواهد گرفت.
- گزارش کار باید شامل جواب سوالات و تصاویر کافی برای هر بخش باشد همچنین نوشتن موارد اضافی و بیش از حد مشمول نمره منفی خواهد شد.
- سوالات خود را تا حد ممکن در فروم درس مطرح کنید تا سایر دانشجویان نیز از پاسخ آن‌ها بهره‌مند شوند. توجه داشته باشید که دیگر شبکه‌های اجتماعی مانند تلگرام راه ارتباطی رسمی با دستیاران آموزشی نیست و دستیاران آموزشی موظف به پاسخگویی در محیط‌های غیررسمی نیستند.