

بسم الله

گزارش کار تمرین کامپیوتری دوم درس سیستم های نهفته بی درنگ

امیراحمد خردادی 810193512

محمد حسین عزیزیان 810195431

محمد مریدی 810195551

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 داوری ای استفاده می شود و چگونه مشکل تداخل را حل کرده است؟

به دلیل مالتیپل اکسسی که روی باس هست امکان تداخل سیگنال ها در این روش وجود دارد. دو روش برای حل این مشکل وجود

دارد:

● Clock stretching: گاهی اسلیو زمان بیشتری برای پردازش داده نیاز دارد یا اینکه می خواهد داده های بیشتری ارسال

کند در این صورت اسلیو کلاک را پایین نگه می دارد و اجازه نمی دهد تا ماستر داده ای ارسال کند. برعکس این حالت هم

ممکن است.

● Arbitration: یک راه دیگر برای جلوگیری از تداخل این است که ماسترها باس را شنود کنند و در صورتی که انتقالی در

جریان نبود داده ی خود را قرار دهند. در صورتی که همزمان دو ماستر داده ای بر روی باس گذاشتند نیاز به یک مکانیزم

داوری داریم تا تعیین کند کدام یک باس را در اختیار بگیرند. این مکانیزم به این صورت است که هر دو خط SDA را

شنود می کنند و مقدار آن را با مقداری که انتظار دارند، داشته باشد مقایسه می کنند، در صورت عدم تطابق آن دستگاه

نمی تواند باس را در اختیار بگیرد و باید صبر کند تا خط خالی شود.

Reference:

<https://www.mikroe.com/blog/i2c-everything-need-know>

۲. مراحل ارتباط دو دستگاه به یکدیگر از طریق این پروتکل (I2C) را بیان کنید.

ابتدا ماستر ارتباط را با فرستادن بیت START آغاز می کند. این بیت به تمامی اسلیوها هشدار می دهد که داده ای قرار است ارسال

شود و می باستی گوش کنند. پس از آن ۷ بیت که تعیین کننده ی آدرس یکتای اسلیو مقصد است توسط ماستر ارسال می شود. هر

اسلیو دارای یک آدرس یکتا است بنابراین تنها یکی از آن ها پاسخ خواهد داد. یک تک بیت نیز ارسال می شود که read/write نام

دارد. اگر این بیت ۰ باشد به این معنی است که مَستَر می‌خواهد در داخل رجیستر اسلیو چیزی بنویسد و اگر ۱ بود یعنی می‌خواهد بخواند. باید توجه داشت که در تمامی ارسال‌ها ابتدا MSB ارسال می‌شود. بیت START به وسیله‌ی پایین آوردن خط SDA هنگامی که خط SCL بالا است (به صورت هم‌زمان) تولید می‌شود. بیت STOP هم به وسیله‌ی بالا آوردن خط SDA هنگامی که خط SCL بالا است (به صورت هم‌زمان) تولید می‌شود. پس هنگامی که یک ماجول می‌خواهد ارسال داشته باشد باید چک کند که SDA بالا باشد.

Reference:

<https://www.mikroc.com/blog/i2c-everything-need-know>

۳. در اتصال UART (که معروفترین پروتکل ارتباط سریال میباشد)، نقش پارامتر baud rate چیست؟ و به چه دلیل این پارامتر اهمیت پیدا میکند؟

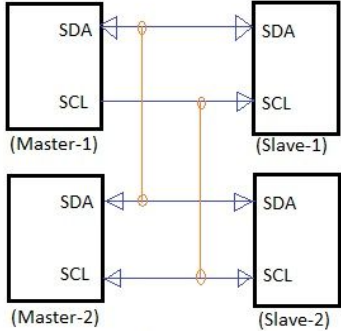
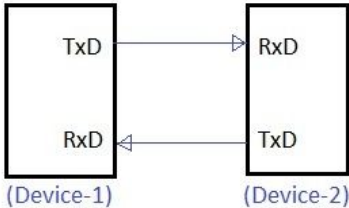
مقدار baud rate نرخ ارسال اطلاعات در یک کانال ارتباطی را در واحد زمان نشان می‌دهد. مثلاً اگر بگوییم مقدار baud rate برابر با 9600 است به این معنی است که می‌توانیم حداکثر 9600 بیت در هر ثانیه ارسال کنیم.

در نرخ‌های ارسال بالا، مثلاً بیشتر از 76800، طول کابل باید کوتاه شود تا انتقال میسر شود. هر چه نرخ ارسال بیشتر شود حساسیت ما برای انتخاب رسانه (مثلاً نوع کابل) بیشتر می‌شود چون حتی یک نویز تصادفی کوتاه مدت هم ممکن است تعداد زیادی از داده‌ها را خراب کند. بنابراین سعی می‌شود هر سمبلی که ارسال می‌شود تا جای ممکن حجم زیادتری از اطلاعات را جابه‌جا کند تا دیگر نیازی به افزایش baud rate نداشته باشیم.

Reference:

<https://www.setra.com/blog/what-is-baud-rate-and-what-cable-length-is-required-1>

۴. هرکدام از روشهای I2C و UART مزایا و معایبی دارند، این دو را با هم مقایسه کنید و برتریهای هر یک را برشمارید.

I2C	UART	
Inter-Integrated Circuit	Serial Peripheral Interface	نام کامل
		نحوه اتصال
SDA: Serial Data SCL: Serial Clock	TxD: Transmit Data RxD: Receive Data	نام سیگنال‌ها و کاربرد
نرخ ارسال‌ها متفاوت است: 100 kbps, 400 kbps, 3.4 Mbps و گاهی: 10 Kbps, 1 Mbps	از آنجایی که ارتباط ناهم‌گام است، نرخ ارسال داده بین دو سیستم باید یکسان شود. نرخ حداکثری از 230 Kbps تا 460 kbps است.	نرخ ارسال
بیشتر از UART	کمتر از ۱۵ متر	فاصله
هم‌گام	ناهم‌گام	نوع ارتباط
۱ یا بیشتر	در این نوع ارتباط معنایی ندارد	تعداد Master ها
یک کلاک مشترک بیت تمام Master ها و Slave ها وجود دارد.	کلاک مشترکی وجود ندارد و هر سیستم کلاک مستقل خود را دارد.	کلاک
پیچیده‌تر از UART	ساده	پیچیدگی سخت‌افزاری
در بالا توضیح داده شد	برای هر بایت یک بیت شروع و پایان	پروتکل انتقال
چندین مَستَر و چندین اسلِیو وجود دارد و	چون نحوه‌ی اتصال یک‌به‌یک است	آدرس‌دهی نرم‌افزاری

همه‌ی مَسترها می‌توانند به تمام اِسلوها متصل شوند. تا ۲۷ اِسلو می‌توانند آدرس‌دهی شوند.	آدرس‌دهی معنی ندارد.	
می‌توان بیشتر از یک اِسلو داشت. با توجه به پیچیدگی به تعداد سیم‌کمی احتیاج دارد (۲ سیم). از مکانیزم‌های کنترل جریان استفاده می‌کند.	پروتکلی بسیار ساده و پرتعداد است، و در تقریباً تمامی دستگاه‌هایی که بیش از ۹ پین ارتباطی دارند، پشتیبانی می‌شود.	مزایا
با افزایش تعداد مسترها و اِسلوها پیچیدگی مدار بیشتر می‌شود. نوع ارتباط دوطرفه half duplex است. برای کنترل پروتکل نیاز به تعبیه‌ی نرم‌افزار داریم بنابراین سربار پردازشی نرم‌افزار را متحمل می‌شویم.	فقط ۲ دستگاه را می‌تواند متصل کند. نرخ ارسال ثابت است و قبل از انتقال تعیین می‌شود و نمی‌توان در طول انتقال آن را تغییر داد.	معایب

Reference:

<http://www.rfwireless-world.com/Terminology/UART-vs-SPI-vs-I2C.html>

۵. در مورد سریال نرم‌افزاری و نحوه‌ی کار آن توضیح مختصری دهید. محدودیت‌های آن نسبت به سریال سخت‌افزاری چیست؟
 بوردهای آردوینو از پروتکل ارتباط سریال به صورت سخت‌افزاری پشتیبانی می‌کنند (توسط پین‌های ۰ و ۱) این پشتیبانی توسط سخت‌افزار UART میسر می‌شود. این سخت‌افزار این امکان را می‌دهد که برد آردوینو داده دریافت کند حتی زمانی که به کار دیگری مشغول است (به شرطی که بافر 64B آن جا داشته باشد).

کتابخانه‌ی SoftwareSerial طراحی شد تا امکان این ارتباط توسط پورت‌های دیجیتالی دیگر را بدهد. به عبارتی نرم‌افزار رفتار سخت‌افزار سریال را تکرار می‌کند (به همین دلیل SoftwareSerial نام گرفت). به کمک این کتابخانه می‌توانیم توسط پورت‌های دیگر تا نرخ 115200 bps داده ارسال کنیم.

محدودیت‌های کتابخانه‌ی SoftwareSerial:

- اگر از چندین پورت استفاده می‌کنید، در هر لحظه تنها از یکی از آن‌ها می‌توانید داده دریافت کنید.
- چون تمام پورت‌ها در برخی Mega ها (Mega and Mega 2560) از change interrupts پشتیبانی نمی‌کنند، فقط پورت‌های زیر می‌توانند برای RX استفاده شوند:

A8 (62), A9 (63), A10 (64), A11 (65), A12 (66), A13, 53, 52, 51, 50, 15, 14, 13, 12, 11, 10
(67), A14 (68), A15 (69)

- چون تمام پورت‌ها در Leonardo و Micro از change interrupts پشتیبانی نمی‌کنند، فقط پورت‌های زیر می‌توانند برای RX استفاده شوند:

8, 9, 10, 11, 14 (MOSI), 16 (SCK), 15 (MISO),

- در Arduino یا Genuino 101 ماکزیمم سرعت RX برابر با 57600bps است.

- در Arduino یا Genuino 101 پین ۱۳ برای RX کار نمی‌کند.

Reference:

<https://www.arduino.cc/en/Reference/softwareSerial>

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؟
این ماجول دارای دو چشم است که به ترتیب در نقش فرستنده و گیرنده‌ی موج فراصوت عمل می‌کنند. سنسور فاصله را به کمک فورمول بسیار ساده زیر محاسبه می‌کند:

$$\text{Distance} = \text{Speed} \times \text{Time}$$

فرستنده یک موج فراصوت می‌فرستد و این موج به هر شی‌ای برخورد کند باز می‌گردد و به گیرنده می‌رسد.



حال برای محاسبه‌ی فاصله به کمک فرمول بالا باید سرعت و زمان را بدانیم. چون از موج فراصوت استفاده می‌کنیم می‌دانیم که سرعت جهانی فراصوت در شرایط اتاق برابر با 330m/s است. مابول هم به کمک مدارات داخلی خود قادر به محاسبه‌ی زمان رفت و برگشت موج می‌باشد. حال به سادگی می‌توان به کمک میکرو فاصله را محاسبه کرد.

Theoretical Measuring Distance: 2cm to 450cm

Practical Measuring Distance: 2cm to 80cm

Accuracy: 3mm

حداقل فاصله لازم برای دریافت داده‌ی صحیح از این سنسور ۲ سانتی‌متر و بیشترین فاصله‌ای که می‌تواند پوشش دهد (در عمل) تقریباً ۸۰ سانتی‌متر است.

Reference:

<https://components101.com/ultrasonic-sensor-working-pinout-datasheet>

۷. اگر در پروژه‌های نیاز به استفاده از دو مازول فاصله‌سنج داشته باشیم چه مشکلی می‌تواند به وجود بیاید؟

در این حالت اگر به صورت همزمان فعال شوند، امواج آن‌ها با هم تداخل پیدا می‌کنند و نتایج عجیب و غریبی را نشان می‌دهند.

Reference:

<https://www.robotshop.com/community/forum/t/two-ultrasonic-sensors-would-they-conflict/11428/2>

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 (مثلاً دو تا سنسور MPU6050) داشته باشیم چطور؟ اگر سه تا از این سنسور داشته باشیم چطور؟
(جواب به ترتیب خیر، خیر، بله است! توضیح دهید چرا)

سنسورهای مختلف در دیتاشیت خود تعیین می‌کنند که مثلاً اگر بورد خارجی‌ای می‌خواهد با من توسط I2C ارتباط برقرار کند باید عدد فلان را (از بین ۱۲۷ عدد مختلفی که می‌توان به عنوان آدرس به کمک ۷ بیت اول بسته I2C تعیین کرد) بر روی باس قرار دهد و چون سنسورهای مختلف اعداد مختلفی را تعیین می‌کنند مشکلی پیش نمی‌آید.

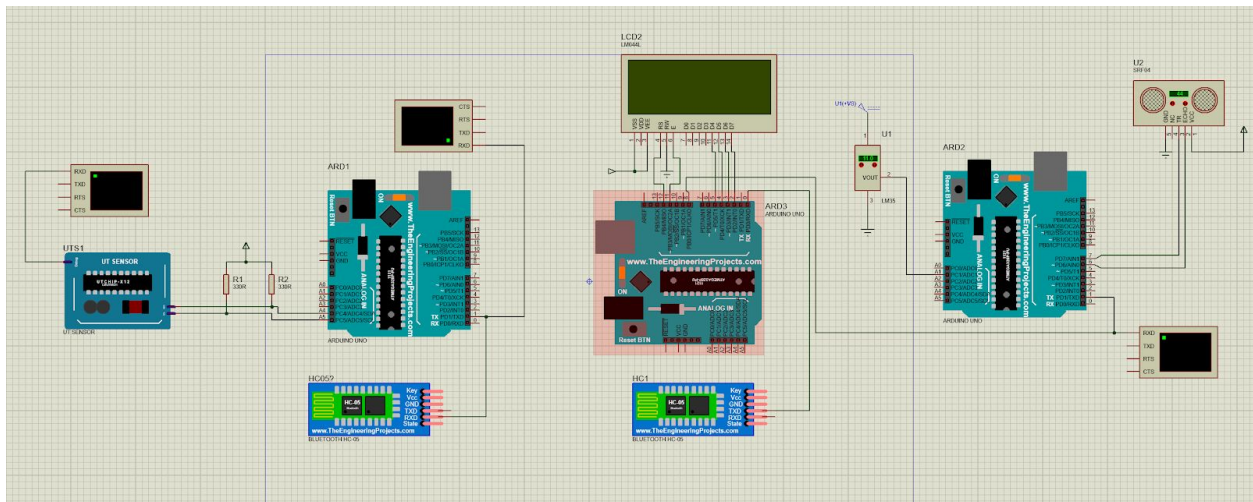
برخی سنسورها مانند MPU6070 این امکان را به برنامه‌نویس می‌دهند که به کمک تغییر دادن یک پین دیجیتال (در این‌جا این پین AD0 نام دارد) آدرس دیگری به عنوان آدرس MPU6070 تعیین شود در نتیجه به کمک یک بیت می‌توان ۲ آدرس مختلف به دو مابول MPU6070 نسبت داد.

همان‌طور که گفته شد، چون AD0 یک پین دیجیتال است و تنها دو حالت دارد بنابراین نمی‌توان بیشتر از دو آدرس برای این ماحول داشته باشیم.

Reference:

<https://components101.com/sensors/mpu6050-module>

۹. طراحی مفهومی این تمرین را مانند نمونه‌های که در تمرین اول دیدید رسم کنید.



main_board:

```
some port and data initializations
set up lcd and serials for Sensor-Board and UT-Board
while true:
    if Sensor-Board data is available:
        parse data (distance & temperature) and show on LCD
    if UT-Board data is available:
        parse data (coordinate) and show on LCD
```

ut_board:

```
some data and pin initialization
setup sensor and serial (I2C)
while true:
    read byte from 4 register
    concating 4 byte to generate X
    read byte from 4 register
    concating 4 byte to generate Y
    calculate result and sent to Main-Board via serial
```

sensor_board:

some data and pin initialization

setup sensor and serial

while true:

 create UltraSonic pulse and calculate distance

 read from temperature sensor and calculate temperature

 concatenate data (distance & temperature) and sent to *Main-Board* via serial