

- 1) Co je operační paměť?
- 2) Jaký je rozdíl mezi master a slave funkcí zařízení na nějaké komunikační lince?
- 3) Jaký je rozdíl mezi čtením a zápisem na nějaké komunikační lince?
- 4) Jaký je rozdíl mezi point-to-point a multidrop komunikační linkou, resp. sběrnici?
- 5) Co je adresa zařízení na sběrnici? Musí mít adresu přidělenou všechna zařízení komunikující po nějaké sběrnici?
- 6) Jaký je význam signálních vodičů SDA a SCL na sběrnici I²C?
- 7) Jak se na sběrnici I²C pozná konec IDLE stavu a začátek nějakého přenosu? Může být přenos principiálně libovolně dlouhý? Jak se pozná jeho konec?
- 8) Co se stane, pokud master vyšle požadavek čtení nějakému I²C slave zařízení, a očekává příjem 4 datových bytů – tj. 3 datové byty master potvrdí (ACK), a 4. datový byte negativně potvrdí (NACK), a poté vygeneruje STOP condition, nicméně adresované slave zařízení pošle pouze 2 datové byty s hodnotami 0x42 a 0xF5, a další data již po I²C sběrnici neposílá?
- 9) Vysvětlete, co to je tzv. clock stretching na sběrnici I²C? Jak probíhá, a kdo ho a kdy provádí?
- 10) Jaký je rozdíl mezi bit ordering a byte ordering? Co znamená byte order MSB-first a co LSB-first?
- 11) Musíme bit ordering řešit u každého přenosu na I²C sběrnici? A musíme byte ordering řešit u každého přenosu na I²C sběrnici?

Skupina úloh „AL“ využívající I²C ambient light sensor z 5. přednášky – společné poznámky: Pro níže uvedené úlohy předpokládejte, že máme k I²C sběrnici připojený ambient light sensor *Everlight ALS-PDIC17-57B/TR8*, který jsme jako příklad používali na 5. přednášce. Při řešení úloh je třeba mít k dispozici datasheet tohoto zařízení, viz příložený soubor „05-ALS-PDIC17-57B-TR8 - I2C ambient light sensor.pdf“¹ (stejný jako v příloze 5. přednášky) – z datasheetu jsou pro nás relevantní pouze strany 1, 3, 9 až 14, a tabulka 3 na straně 15.

12AL) Nakreslete časový diagram signálů na vodičích SDA i SCL při průběhu celého přenosu (včetně START a STOP condition), kdy nějaký master posílá ALS senzoru požadavek o konec integrování.

13AL) Nakreslete časový diagram signálů na vodičích SDA i SCL při průběhu celého přenosu (včetně START a STOP condition), kdy nějaký master čte z ALS senzoru naměřenou hodnotu intenzity světla. Předpokládejte, že ALS senzorem naměřená 15-bitová hodnota (count) je rovna 4783.

Skupina úloh „TT“ využívající I²C senzor tlaku a teploty – společné poznámky: Pro níže uvedené úlohy předpokládejte, že máme k I²C sběrnici připojený senzor tlaku a teploty *AMS 6915-1200-B* od firmy *Analog Microelectronics*. Při řešení úloh je třeba mít k dispozici datasheet tohoto zařízení, viz příložený soubor „05-SA-ams6915-datasheet - I2C pressure and temperature sensor (one read).pdf“² – z datasheetu jsou pro nás relevantní pouze strany 2, 7 až 10.

14TT) Napište v šestnáctkové soustavě hodnoty všech bytů, které budou přeneseny po I²C sběrnici v průběhu celého přenosu (tj. veškeré byty poslané mezi START a STOP condition; pro každý přenášený byte uvažujte pouze hodnoty 8 bitů bez ACK bitu), kdy nějaký master čte ze senzoru hodnotu naměřeného tlaku a teploty. Předpokládejte, že senzorem posílaná 14-bitová hodnota tlaku je rovna 16018, a 11-bitová hodnota teploty je rovna 902.

15TT) Je možné si od zařízení *AMS 6915-1200-B* vyžádat pouze hodnotu naměřeného tlaku, nebo pouze hodnotu naměřené teploty, a nikoliv obojí současně? Pokud ano, tak detailně vysvětlete, jak by se takový požadavek lišil od „běžného požadavku“ na obě hodnoty tlaku a teploty současně.

Sada programovacích úloh – společné poznámky: Vezměte si projekt *PySimulatedI2c-ALS-PressureTempSens-ASSIGNMENT*, který v souboru *PySimulatedI2cMainProgram.py* obsahuje kostru programu, který máte doplnit v řešení úloh 16AL, 17AL, 18TT, 19TT. Součástí projektu je též implementace simulovaných příkazů CPU pro komunikaci po I²C

¹ Původní zdroj (květen 2016): <http://www.everlight.com/file/ProductFile/ALS-PDIC17-57B-TR8.pdf>

² Původní zdroj (říjen 2020): <https://www.analog-micro.com/products/pressure-sensors/board-mount-pressure-sensors/ams6915/ams6915-datasheet.pdf>

sběrnici (resp. přesněji simulovaného řadiče I²C sběrnice), podobně jako jsme ve dřívějších self-assessment úlohách měli simulaci řadiče RS-232 linky. Příkazy pro komunikaci po I²C lince zadávejte pomocí volání funkcí na v programu předpřipravené proměnné `i2c`, která obsahuje odkaz na objekt typu `SimulatedGenericI2cController`. `I2c` poskytující programátorské rozhraní k simulované I²C sběrnici (podobně jako jsme v dřívějších self-assessment úlohách měli proměnnou `serialPort`, která obsahovala odkaz na objekt typu `serial` sloužící jako programátorské rozhraní k simulovanému řadiči RS-232). Simulovaný I²C řadič se na sběrnici I²C chová vždy jako master. Na objektu `i2c` máte k dispozici 2 funkce:

`i2c.send(list-bytů-k-poslání-po-simulované-I2C-sběrnici)`

- Pokusí se po I²C sběrnici poslat byty předané jako seznam (`List`) v parametru – každá položka seznamu by mělo být 8-bitové beznaménkové číslo. Byte v seznamu na indexu 0 se posílá jako první, byte na nejvyšším indexu se posílá jako poslední. Řadič za vás po sběrnici posílá byty právě tak, jak je zadáte, a neprovádí u nich žádné úpravy.
- Předaný seznam bytů musí mít vždy alespoň 1 položku, jelikož byte na indexu 0 je chápán jako „adresový byte“ sběrnice I²C.
- Předaný seznam bytů musí mít právě 1 položku, pokud se jedná o požadavek o čtení.
- Pokud je poslaný byte potvrzený (ACK) zařízením slave, tak následuje posílání následujícího bytu ze seznamu. Pokud je nějaký poslaný byte nepotvrzený (NACK), tak se další byty ze seznamu již neposílají.
- Funkce vrací celé číslo typu `int`, které reprezentuje počet potvrzených (ACK) odeslaných bytů z předaného seznamu. Vracená hodnota tedy nebude nikdy větší, než je počet položek předaného seznamu bytů. Funkce vrátí hodnotu 0, pokud nebyl žádným slave zařízením potvrzen ani „adresový byte“.
- Před prvním poslaným bytem vygeneruje řadič na simulované sběrnici START condition.
- Pokud nedošlo k potvrzení „adresového bytu“, tak řadič na simulované sběrnici vygeneruje STOP condition.
- Pokud k potvrzení „adresového bytu“ došlo, a jedná se o požadavek zápisu, tak řadič na simulované sběrnici vygeneruje STOP condition po přijetí ACK/NACK bitu posledního odeslaného bytu.

`i2c.receive(maximální-počet-bytů-k-přijetí-ze-simulované-I2C-sběrnice)`

- Volání této funkce musí vždy předcházet volání funkce `send` s požadavkem čtení, a „adresový byte“ tohoto požadavku musí být potvrzen (ACK) nějakým slave zařízením.
- Funkce `receive` se pokusí z I²C sběrnice přijmout N bytů, kde N je parametr předaný funkci. Všechny přijaté byty kromě posledního (tj. prvních N-1 bytů) jsou pozitivně potvrzeny (ACK). Poslední přijatý byte je negativně potvrzen (NACK). Po přijetí N-tého bytu vygeneruje řadič na simulované I²C sběrnici STOP condition.
- Funkce vrací seznam přijatých bytů jako seznam (`List`) hodnot typu `uint8`. Na indexu 0 je první přijatý byte, na nejvyšším indexu je byte, který byl přijat jako poslední.

Předpokládejte, že k simulované I²C sběrnici jsou současně připojená virtuální slave zařízení, která simulují chování ambient light sensoru `ALS-PDIC17-57B/TR8` ze skupiny úloh AL, a senzoru tlaku a teploty `AMS 6915-1200-B` ze skupiny úloh TT.

Poznámka: Řešení úloh 16AL, 17AL, 18TT, 19TT zkuste vypracovat „bez podvádění“, tj. pouze s využitím informací z příložených datasheetů, a bez nahlížení do souborů `SimulatedDeviceEverlightALS_PDIC17_57B.py` a `SimulatedDeviceAnalogMicroAMS6915.py`, které obsahují implementaci simulace daných zařízení. V reálném životě by takové použité zařízení bylo pro vás také „černá skříňka“, a museli byste se řídit výhradně jeho datasheetem. Po dokončení a odladění vaší implementace úloh 16AL, 17AL, 18TT, 19TT se samozřejmě můžete do uvedených souborů podívat, a prostudovat si, jak je jejich simulace naprogramována (v implementaci simulace se používá Pythonový typ `datetime` reprezentující datum a čas, a typ `timedelta` reprezentující časový interval – pokud je neznáte, tak několik informací k nim bylo v zadání 2. sady self-assessment úloh).

Poznámka: Pro čekání můžete v programu použít např. standardní funkci `time.sleep(doba-v-sekundách)` – funkce je k dispozici ze standardního Python modulu `time`, je blokující, a čeká zadaný počet sekund. Předaná hodnota je reálné číslo, a je tedy možné čekat i menší časové úseky, než je 1 sekunda.

Poznámka: Pokud vám nebude v programu správně vycházet časování jednotlivých operací, tak se může stát, že vám výsledné hodnoty nebudou vycházet přesně dle „vzorového výstupu“ uvedeného v komentářích v programu. Zda data z paketů parsujete správně, si v tom případě můžete ověřit zapnutím ladících výpisů odkomentováním řádku

`SimulatedGenericI2cController.enableLog = True`, a ověřením že hodnoty generované v simulaci jsou stejné s vašimi.

Poznámka: V řešení níže uvedených úloh pro jednoduchost neřešte a neošetřujte chybové stavy. Předpokládejte, že jsou všechna zařízení na I²C sběrnici správně připojena, a komunikace s nimi probíhá vždy bez chyb a dle specifikace. Pouze pokud již patříte mezi pokročilejší programátory, tak by mohlo být zajímavé ošetřit v programu i všechny okrajové situace, které by při komunikaci s reálným zařízením připojeným na reálné I²C sběrnici mohlo nastat.

- 16AL)** Doplníte program v `PySimulatedI2cMainProgram.py` tak, aby provedl 15 měření intenzity světla pomocí připojeného ambient light sensoru. Program má u ALS senzoru vynutit 300 ms integraci (měření intenzity světla). Jednotlivá měření mají probíhat hned za sebou. Program má vypisovat v desítkové soustavě přijatou 15-bitovou hodnotu reprezentující naměřenou intenzitu světla.
- 17AL)** *Pro lehce pokročilejší (vyžaduje základní znalost práce s reálnými čísly a Python typem float):* Doplníte vaše řešení úlohy 16AL tak, aby se kromě samotné přijaté hodnoty vypisovala i skutečně naměřená intenzita světla v luxech (jako reálné číslo v desítkové soustavě zaokrouhlené na 2 desetinná místa) dle přepočtu uvedeného v datasheetu zařízení (tj. součástí této úlohy je i správné vyřešení úlohy 16AL).
- 18TT)** Doplníte program v `PySimulatedI2cMainProgram.py` tak, aby provedl 10 měření tlaku a teploty pomocí připojeného senzoru. Jednotlivá měření mají probíhat v 250 ms intervalech. Program má vypisovat v desítkové soustavě přijatou 14-bitovou hodnotu reprezentující naměřený tlak, a 11-bitovou hodnotu reprezentující naměřenou teplotu.
- 19TT)** *Pro lehce pokročilejší (vyžaduje základní znalost práce s reálnými čísly a Python typem float):* Doplníte vaše řešení úlohy 18TT tak, aby se kromě samotných přijatých hodnot vypisovala i skutečně naměřená hodnota tlaku v milibarech (jako reálné číslo v desítkové soustavě zaokrouhlené na 1 desetinné místo), a skutečně naměřená hodnota teploty ve stupních Celsia (jako reálné číslo v desítkové soustavě zaokrouhlené na 1 desetinné místo) – postupujte dle vzorců uvedených v datasheetu zařízení (tj. součástí této úlohy je i správné vyřešení úlohy 18TT).

Poznámka: Cílem těchto úloh je, abyste si doma v rámci opakování látky z přednášky mohli sami ověřit, jak jste látku porozuměli. Úlohy jsou koncipovány víceméně přímočaře, a po projití poznámek a případně video záznamů z přednášek by jejich řešení mělo být zřejmé. Pro řešení úloh tedy není potřeba studium látky nad rámec probraný na přednáškách (nicméně je třeba mít i znalosti a pochopení látky z paralelně probíhajících přednášek a cvičení z předmětu Programování I). Pokud i po detailním a opakovaném projití látky z přednášek máte s řešením těchto úloh problém, tak se na nejasnosti co nejdříve ptejte na on-line konzultaci k přednášce, případně zvažte se mnou domluvit na konzultaci (zvlášť pokud tento stav u vás přetrvává i po dalších přednáškách).

Upozornění: Úlohy jsou vybrány a postaveny tak, abyste si po přednášce a před přednáškou následující mohli rychle připomenout hlavní části probrané látky a ověřit si její pochopení. Nicméně úlohy nejsou vyčerpávajícím přehledem látky z přednášek a tady rozhodně nepokrývají kompletní látku přednášek, která bude vyžadována u zkoušky. Pokud tedy u každé úlohy víte, jak by se měla řešit, tak to ještě neznamená, že jste dostatečně připraveni na zkoušku – nicméně jste jistě na velmi dobré cestě. Každopádně nezapomeňte, že na zkoušce se vyžaduje pochopení a porozumění právě všem konceptům ze všech přednášek, a navíc jsou zkouškové příklady postaveny komplexněji tak, aby ověřily také vaši schopnost přemýšlet nad látkou napříč jednotlivými přednáškami.

(Červeně jsou vyznačeny nové části verze 2 tohoto dokumentu oproti jeho původní verzi 1.)