

Лабораторно упражнение №6

“Изследване на сериен интерфейс I²C на микроконтролера MSP430FR6989”

6.1. Въведение.

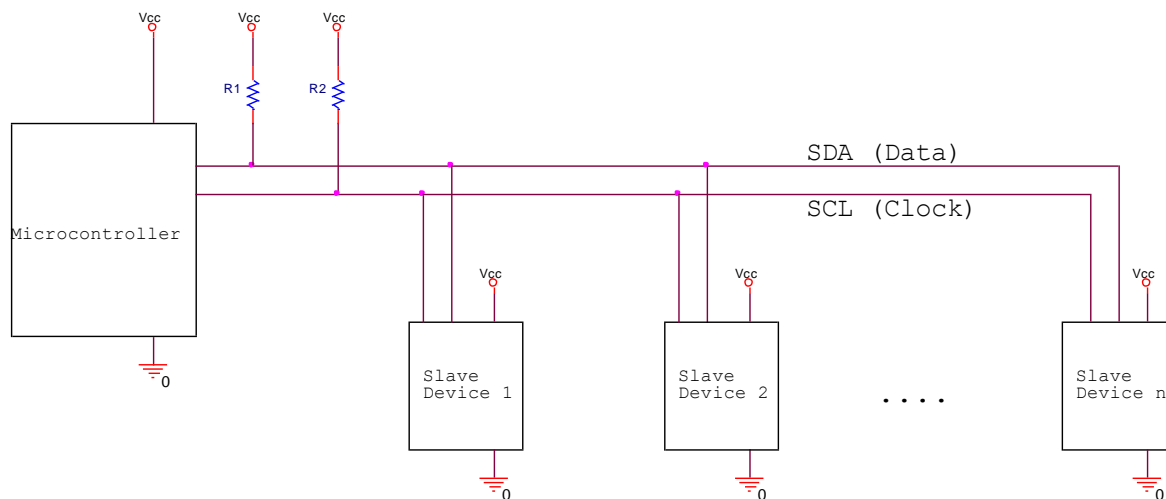
I²C е синхронен сериен интерфейс, който използва два (+ 1 маса) проводника за предаване на информация между два и повече чипа. Името му идва от ИС – Inter-Integrated Circuit и е създаден от Philips през 1982 г. От 2006 година използването на протокола за този интерфейс е безплатно. Платено е единствено запазването на уникален адрес за чип от даден вид, използващ I²C (това засяга само фирмите-производители на чипа).

Връзката между два чипа се реализира на принципа master-slave, т.е. има една главна интегрална схема (master) и една подчинена (slave) такава. Обикновено master чипа е микроконтролерът в системата, а slave е периферен чип, с който ще се осъществява връзката. Съществува и режим multi-master, при който към I²C интерфейса са свързани два или повече главни (master) чипа.

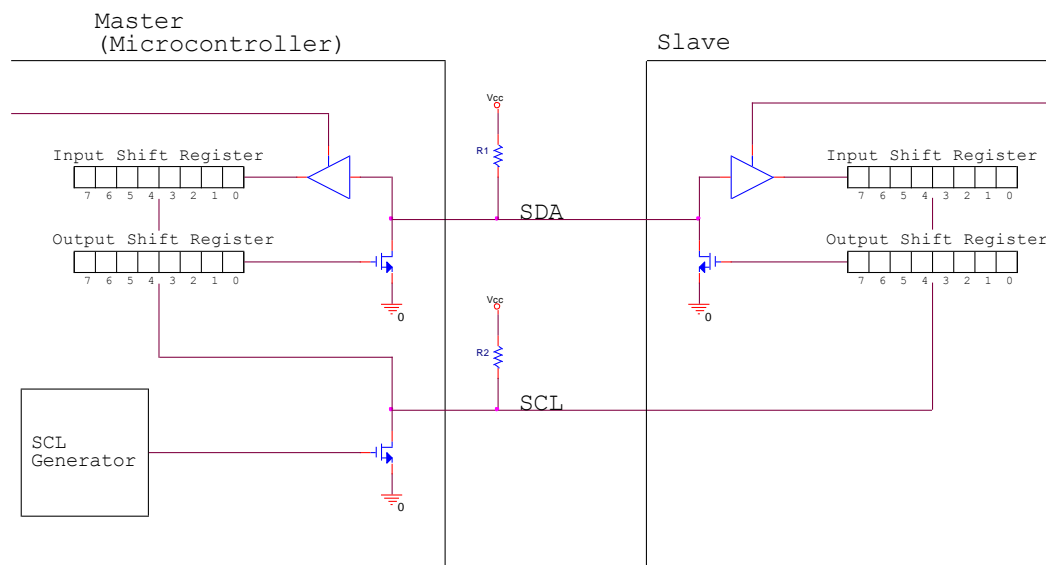
На **фиг. 6.1** е показана блокова схема на система, използваща I²C интерфейса. Предаването на информацията се осъществява посредством сигналите:

- SDA – данните, предавани серийно по този проводник.
- SCL – тактовият сигнал, по който е синхронизирано изпращането бит по бит на данните.

Към двата проводника са свързани изтеглящи (pull-up) резистори към захранването на системата V_{cc}. Те са необходими, защото предаването на информация е двупосочно – от микроконтролера към подчинената схема и от подчинената схема към микроконтролера. Това означава, че микроконтролерът ще използва изводът си, свързан към SDA, веднъж като вход и веднъж като изход. Аналогично – подчиненото устройство ще използва SDA като вход, когато иска да приеме информация и след това като изход, когато иска да предаде информация (**фиг. 6.2**). Тази конфигурация води до една теоретична конфликтна ситуация, в която ако и двата чипа са установили изводите си SDA като изходи и единият е в логическа 1, а другият – в логическа 0, то ще се получи късо съединение между захранващия извод V_{cc} и масата GND. За да се избегне тази възможност се използват изходи с отворен дрейн (или колектор), които изискват режимен pull-up резистор. SCL също включва такъв резистор, защото протокола I²C дефинира ситуация, в която някой от чиповете задържа тактовия сигнал в логическа 0.



Фиг. 6.1. Блокова схема на система, използваща I²C интерфейса



Фиг. 6.2. Хардуерна реализация на интерфейса

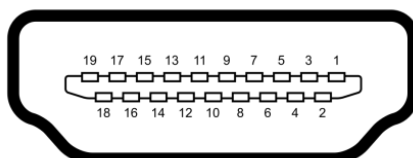
Аналогични на I2C са интерфейсите:

- SMBus – създаден от Intel през 1995. Дефинира по-строго изискванията за ниво и времеви параметри (timings) на сигнала.
- I2S – използва се за предаване на цифров звуков сигнал. Съдържа един допълнителен проводник (Word Select), указващ изпращаният байт на кой канал принадлежи (при стерео предаване – ляв или десен).

I2C интерфейса се характеризира със следните скорости на предаване:

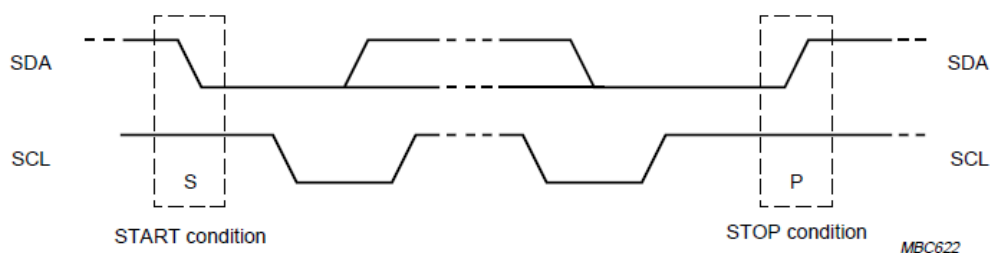
- Оригинална версия от 1982 – 100 kbit/s.
- Версия 1 (1992) – 400 kbit/s (Fast mode).
- Версия 2 (1998) – 3.4 Mbit/s (High-speed mode).
- Версия 3 (2007) – 1 Mbit/s (Fast mode plus).
- Версия 4 (2012) – 5 Mbit/s (Ultra Fast mode).

I2C интерфейса се включва в интегрални схеми с различни функции във вградените системи. АЦП, ЦАП, EEPROM, цифрови потенциометри, температурни датчици, хол датчици и т.н. са само малка част от приложението му. Въпреки че този интерфейс се използва главно за комуникация между ИС в рамките на една вградена система, то не липсват примери за приложения и в комуникацията между две отделни устройства. Например всеки персонален компютър използва I2C като част от HDMI интерфейса **фиг. 6.3**. С негова помощ се „опознават“ поддържаните видео формати от изобразяващото устройство (монитор, телевизор, проектор и други).



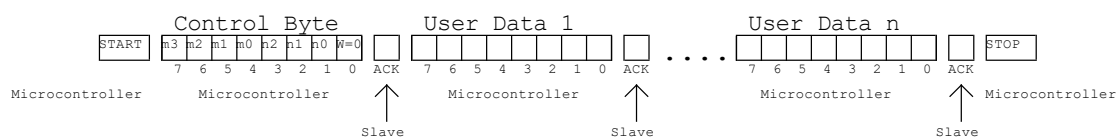
Фиг. 6.3. HDMI куплунг. I²C на извод 15 – SCL и извод 16 – SDA

I²C комуникацията се осъществява по стандартизиран протокол. Най-общо казано обменът на данни започва с условие СТАРТ, продължава с трансфер на данни и завършва с условие СТОП. Условието старт и стоп са специални събития, реализирани чрез комбинации на логически нива и фронтове на SCL и SDA линиите, които са уникални и се различават от данните. На **фиг. 6.4** са показани едно старт и стоп условие. Старт условие се генерира, когато SDA линията премине в ниско ниво, докато SCL линията е във високо. Стоп условие се генерира, когато SDA линията премине във високо ниво, докато SCL линията е във високо. Между старт и стоп условията се изпращат потребителските данни. Всеки бит от тях трябва да запази своето състояние, докато SCL е във високо ниво (в противен случай ще се генерира старт или стоп условие и трансферът ще се наруши), т.е. трансферът на данни се осъществява по ниво.



Фиг. 6.4. Състояние на линиите SDA и SCL при START и STOP условие

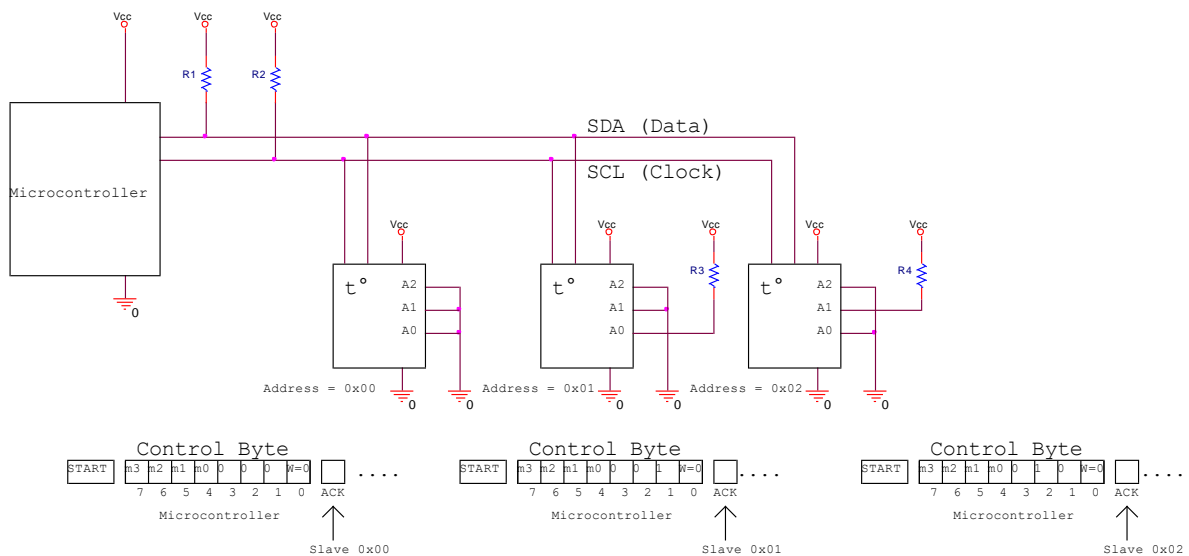
Форматът на данните при **запис** от микроконтролер в подчинена ИС е показан на **фиг. 6.5**.



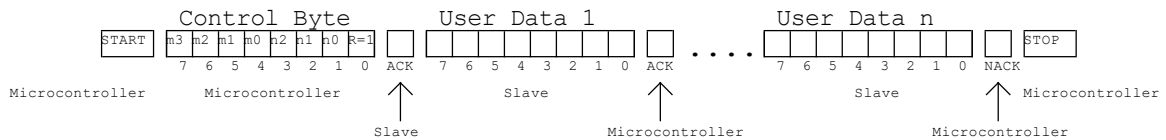
Фиг. 6.5. Форматът на данните при запис от микроконтролер в подчинена ИС

Микроконтролерът генерира СТАРТ условие, след което изпраща контролен байт. Този байт достига до всички подчинени устройства. Старшата тетрада (битове $m0 \div m3$) съдържа контролен код, част от 7-битов адрес, който е различен за различните подчинени чипове – трябва да се провери datasheet-а им. Следват три бита, които указват младшата част на 7-битовия адрес на подчиненото устройство (битове $n0 \div n2$). Благодарение на тях към една I²C шина могат да се свържат повече от един чипове от един и същи вид (на **фиг. 6.6** е показано свързване на три еднакви температурни датчика, чиито адреси са зададени хардуерно с изводи 1, 2 и 3). Последният бит от контролния байт е четене/запис (R / !W) и при запис трябва да е 0. Ако на I²C шината има устройство със зададения адрес (битове $m0 \div m3$ и $n0 \div n2$), то трябва да отговори с изпращане на един бит, наречен ACK (acknowledge) или още - потвърждение. Неговото ниво е логическа 0. След това се изпращат потребителските данни User Data 1 ... User Data n и най-накрая се генерира условие стоп, с което трансферът приключва.

Трансферът на данните от подчинена ИС към микроконтролера се определя като **четене** и форматът е показан на **фиг. 6.7**. Да се обърне внимание на предавателя и приемника под всеки байт на **фиг. 6.5** и **фиг. 6.7**! Ако микроконтролерът иска да спре приемането на данни, той трябва да генерира NACK бит (No Acknowledge, логическа 1) и след това условие СТОП.

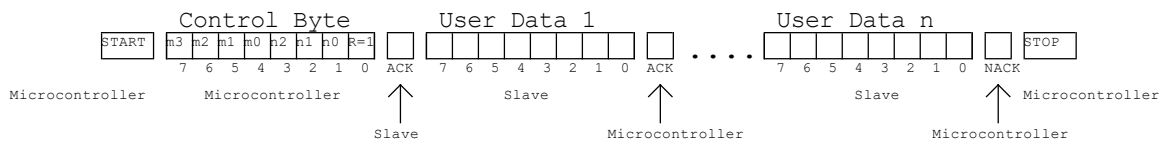


Фиг. 6.6. Свързване на три еднакви температурни датчика, чиито адреси са зададени хардуерно с изводи 1, 2 и 3



Фиг. 6.7. Форматът на данните при четене от подчинена ИС (трансфер от подчинената ИС към микроконтролера)

Отделените седем бита $m0 \div m3$ и $n0 \div n2$ позволяват на една I²C шина да се свържат до $2^7 = 128$ чипа (минус един бродкаст адрес и няколко за бъдещо ползване). Версия 1 дефинира някои леки промени в протокола, които позволяват адресиране на $2^{10} = 1024$ (т.е. 10-битов адрес) чипа. Те са показани на **фиг. 6.8**. Новото тук е фиксираното число 11110 в контролния байт. То указва, че ще се използва 10-битово адресиране. Следват битове $z9 \div z0$, които са 10-битовия адрес на подчиненото устройство. Да се обърне внимание на двойното изпращане на условие СТАРТ и редуването на $W=0$ и $R=1$ при четене ($MCU \leftarrow SLAVE$).



Фиг. 6.8. Протокола на четене и запис на данни при версия 1 на интерфейса

ЗАДАЧИ ЗА ИЗПЪЛНЕНИЕ

1. Да се създаде нов проект с име **Lab_6_1** в папка **/Desktop/MSHT_GR_XX_N/Lab_6_1**, да се копира програмата на C, чрез която се четат данни от температурния датчик TMP102 посредством I²C интерфейс. Да се наблюдават SCL и SDA линиите с осцилоскоп и да се свалят осцилограмите.

Datasheet на температурния сензор: <https://www.ti.com/lit/gpn/tmp102>

2. Да се създаде нов проект с име **Lab_6_2** в папка **/Desktop/MSHT_GR_XX_N/Lab_6_2** и да се въведе програмата на C, чрез която се реализира запис по I²C интерфейса. За целта да се запише произволно число в един от регистрите T_{LOW} или T_{HIGH} на температурния датчик TMP102. Да се снимат осцилограмите. Да се сравнят с осцилограмите от предишната задача. Да се обяснят разликите.