

EMBEDDED

л.5

Цифрова комунікація

Палій Святослав



Що таке комунікація?

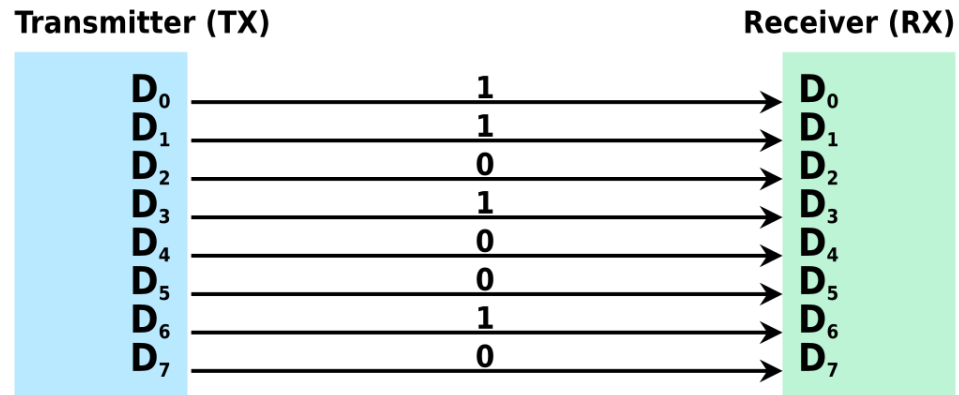
- **Інтерфейс (*interface*)** слід розуміти як **електронну схему, завдання якої полягає в передачі сигналів з однієї схеми до іншої**. Зазвичай він має здатність передавати чи/та приймати. Поняття інтерфейсу відноситься як до фізичного шару (наприклад, наявність певних виводів, або навіть конкретного роз'єму), так і до функціонального (можливість кодування/декодування повідомлень за певним стандартом). Якщо ми говоримо, що мікроконтролер оснащений інтерфейсом UART, це означає, що він має інтегровану схему, здатну передавати дані в цьому стандарті, і що ця схема може бути підключена до виводів інтегральної схеми.
- **Шина (*bus*)** відноситься до будови самої схеми. Йдеться про **фізичні шляхи, доріжки на платі або дроти, які використовуються для з'єднання компонентів**. Шина складається з шин, тобто ліній згрупованих за своєю функцією, наприклад, адресні лінії, шина даних тощо. На практиці основних електронних схем рідко зустрічаються шини, оскільки найчастіше кожна лінія шини має свою специфічну функцію. Відразу звернімо увагу, що хоча поняття інтерфейсу і шини схожі, вони відносяться до різних елементів системи комунікації: інтерфейс є електронною схемою, яка служить для підключення даної схеми до шини.
- Це залишає нам останнє важливе поняття: **протокол**. Це найбільш інтуїтивна з назв, її можна розуміти навіть буквально: протокол, тобто **набір правил, що складає метод комунікації**. Це щось трохи більше, ніж просто мова, якою передаються дані, оскільки він також описує всю процедуру обміну інформацією. Досить часто в мікропроцесорних системах, простих електронних пристроях тощо – інтерфейс, шина і протокол описані одним стандартом.

Послідовна чи паралельна?

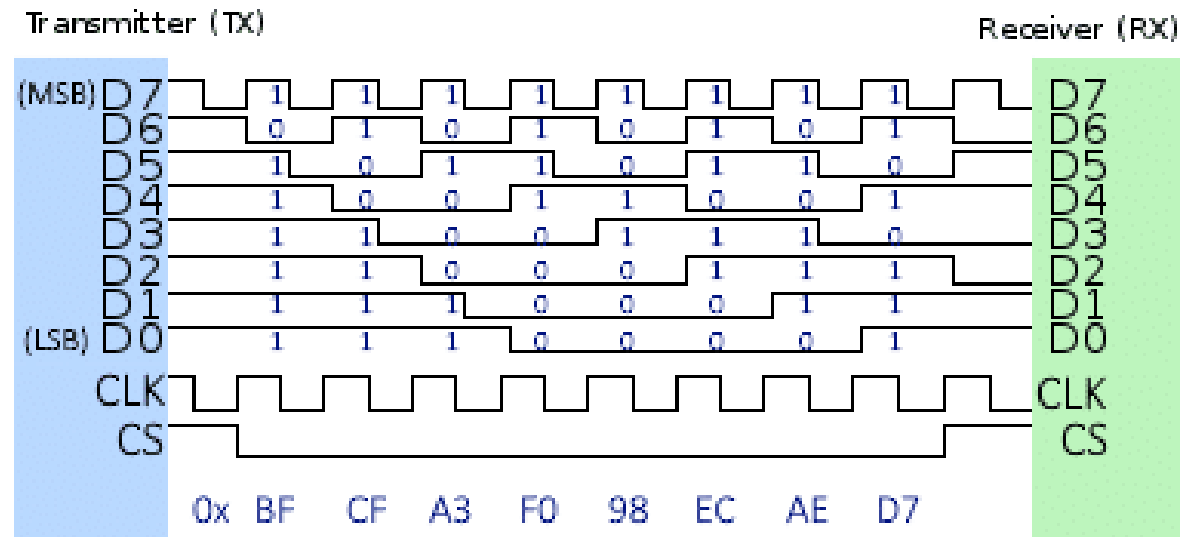
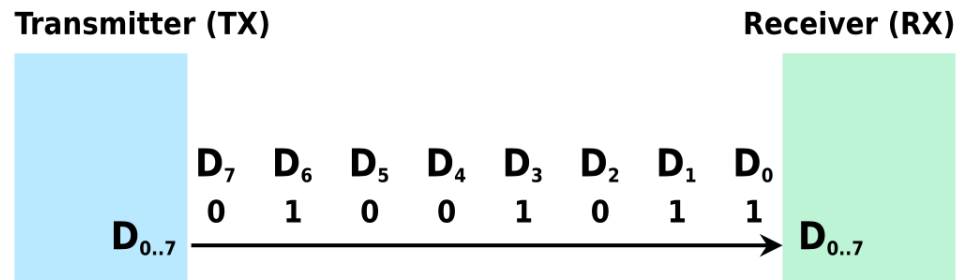


- Метод **паралельної** комунікації полягає в одночасній передачі кількох бітів даних, тоді як послідовні інтерфейси зазвичай служать для послідовної передачі. Паралельні шини користувалися великою популярністю ще дуже донедавна, оскільки дозволяли передавати відносно велику кількість даних при низькій частоті сигналів. Щоб передати ту ж кількість бітів, послідовна передача потребує восьми імпульсів, а отже, такий інтерфейс працює з восьмиразово більшою частотою.
- Сьогодні комунікаційні схеми досягають частот порядку сотень гігагерців, що дозволяє широко використовувати послідовні інтерфейси – навіть там, де потрібна значна швидкість передачі. Прикладом в ПК можна побачити SATA якими під'єднуються жорсткі диски, які замінили паралельну технологію ATA.
- **Послідовний метод передачі має конструктивні та практичні переваги**, оскільки вимагає меншої кількості ліній, а отже, наприклад, використання тонших кабелів. Послідовні шини займають меншу площу на друкованих платах, оскільки вимагають прокладання лише 2-4 шляхів (замість кількох десятків), сприяючи мініатюризації електронних пристроїв.
- Прикладом цього останнього прогресу можуть служити одні з найпопулярніших дисплеїв: алфавітно-цифрові LCD, з контролером HD44780. Вони доступні в двох версіях: з паралельним інтерфейсом, де для управління потрібно 10 ліній; та керованих за допомогою послідовного інтерфейсу I²C використовуючи лише 2 проводів та ще двох для живлення!

Parallel interface example

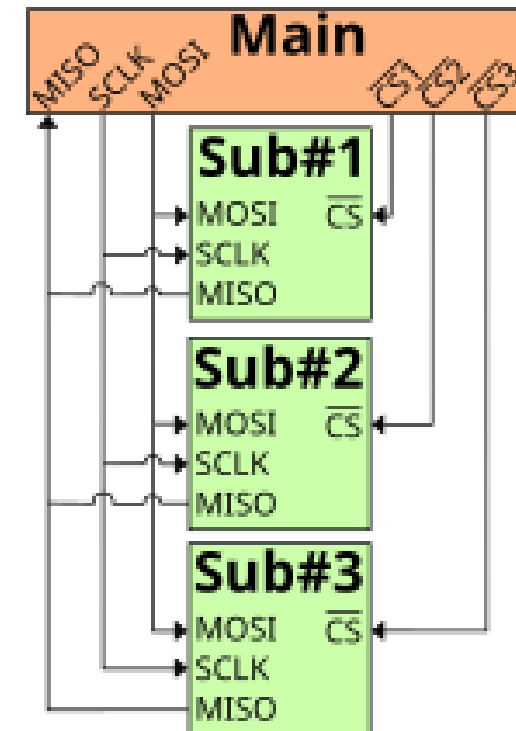
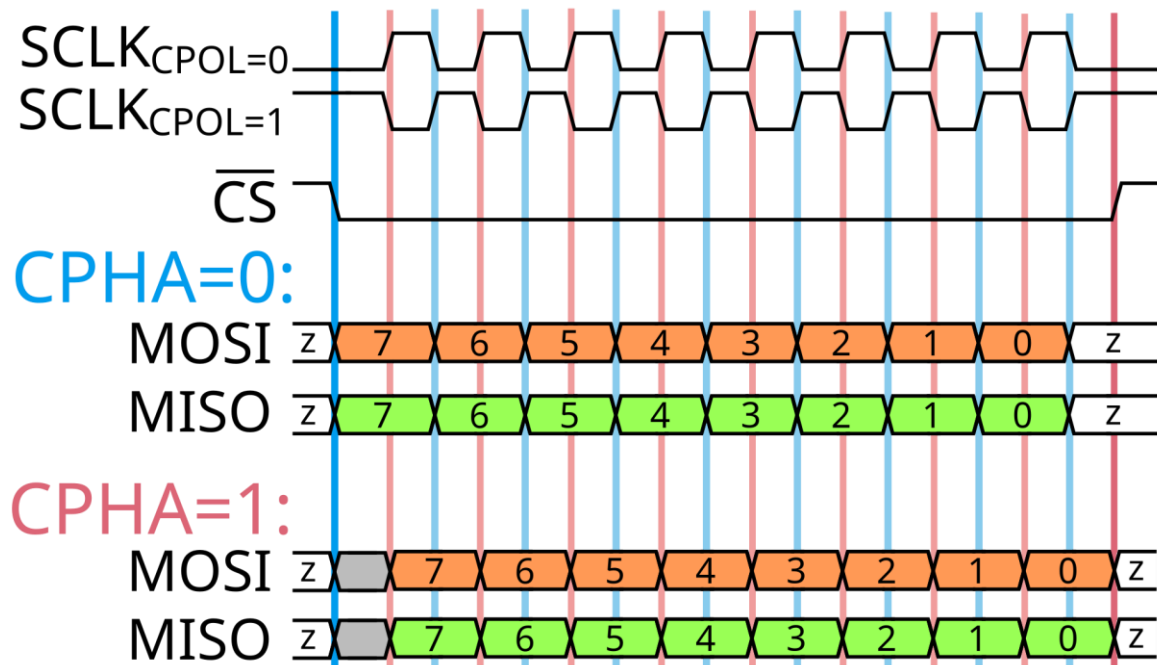
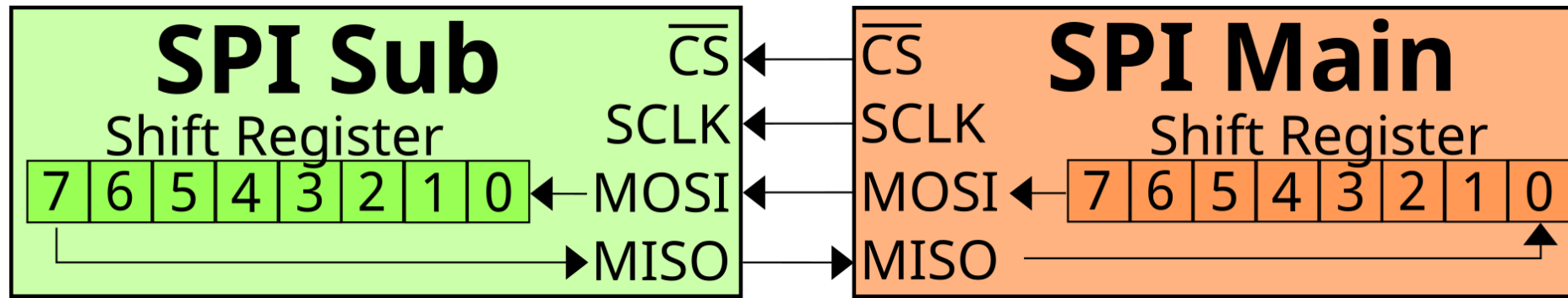


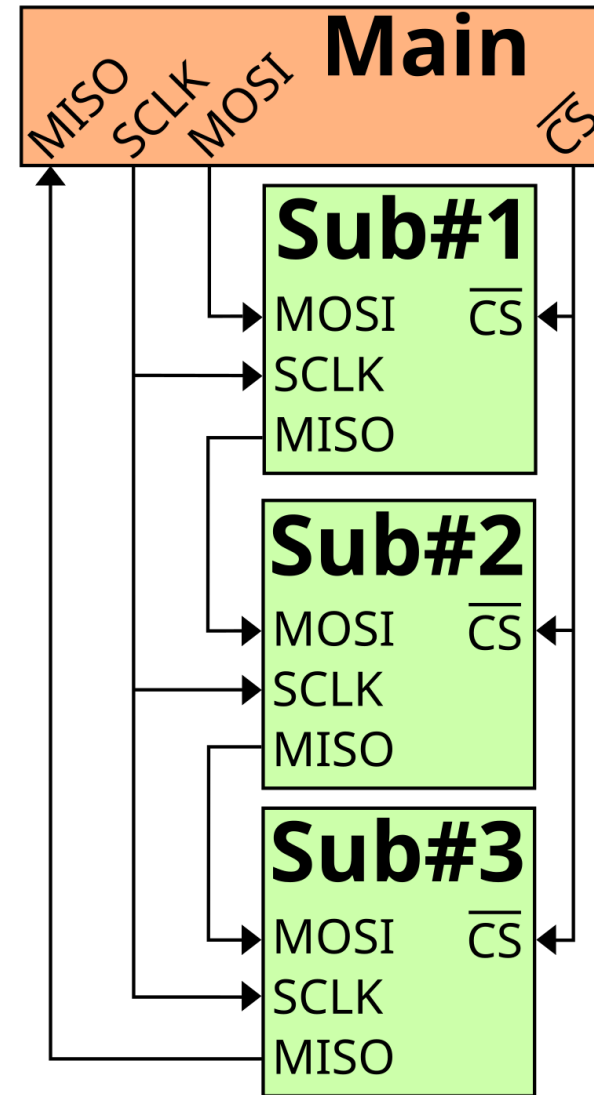
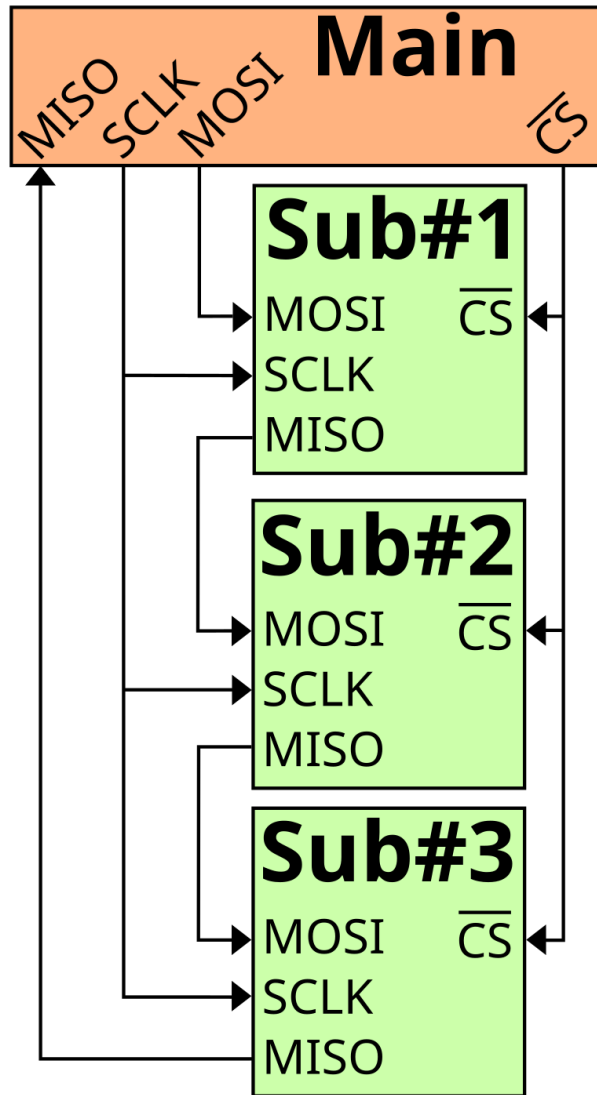
Serial interface example



SPI

- **Serial Peripheral Interface.** Це одна з найстаріших та найпростіших спроб створення універсального інтерфейсу комунікації в цифрових схемах (кінець 70-х років, Motorola). SPI залишається дуже популярним стандартом, що використовується, наприклад, для комунікації з картами пам'яті SD чи там де потрібні високі швидкості комунікації.
- SPI складається з чотирьох ліній: тактовий сигнал SCLK, дві лінії даних та Chip Select (CS). Кожна схема, що працює в режимі slave (згідно нової термінології sub), має власну лінію CS, яка служить для сигналізації, з яким sub елементом, підключеним до шини, ведеться комунікація. SPI передбачає дві лінії даних, оскільки передача може відбуватися одночасно в обох напрямках (full duplex!). Шина містить сигнал MOSI (~~Master~~ Main Out, ~~Slave~~ Sub In) та MISO (~~Master~~ Main In, ~~Slave~~ Sub Out), де перший служить для передачі даних від контролера до sub схем, тоді як другий працює в зворотному напрямку.
- Інтерфейс SPI простий у реалізації, завдяки строгим сигналам (push-pull) **дозволяє дуже швидко передачу даних**, що робить його широко використовуваним для обслуговування різноманітних комунікаційних модулів, пам'яті, регістрів та інших швидких периферійних пристроїв. Сучасно практично кожен мікроконтролер пропонує апаратну підтримку цього інтерфейсу, що означає, що його використання не навантажує ядро і дозволяє конструювати мініатюрні пристрої зі складною функціональністю.





SPI mode	Clock polarity (CPOL)	Clock phase (CPHA)	Data is shifted out on	Data is sampled on
0	0	0	falling SCLK, and when \overline{CS} activates	rising SCLK
1	0	1	rising SCLK	falling SCLK
2	1	0	rising SCLK, and when \overline{CS} activates	falling SCLK
3	1	1	falling SCLK	rising SCLK

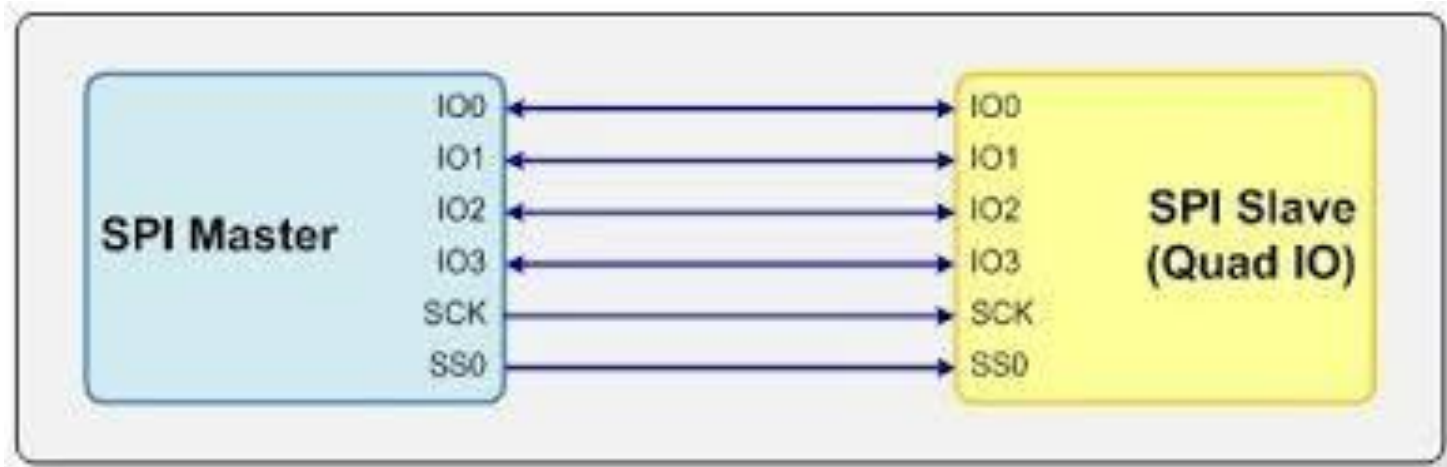
- **Послідовний**
- **Дуже швидкий**
- **Синхронний**
- **Main-sub**
- **Довжина слова не лімітована**

- **Відсутнє підтвердження прийому**
- **Відсутній flow-control**
- **Відсутній контроль помилок**

Подібні:

- **μ Wire**
- **JTAG**

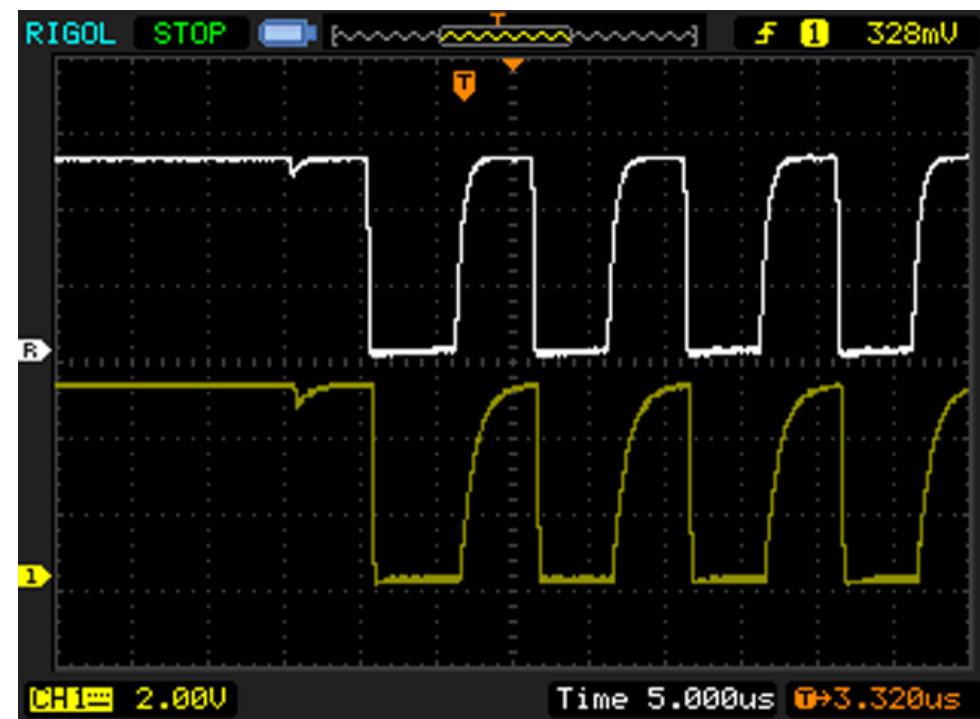
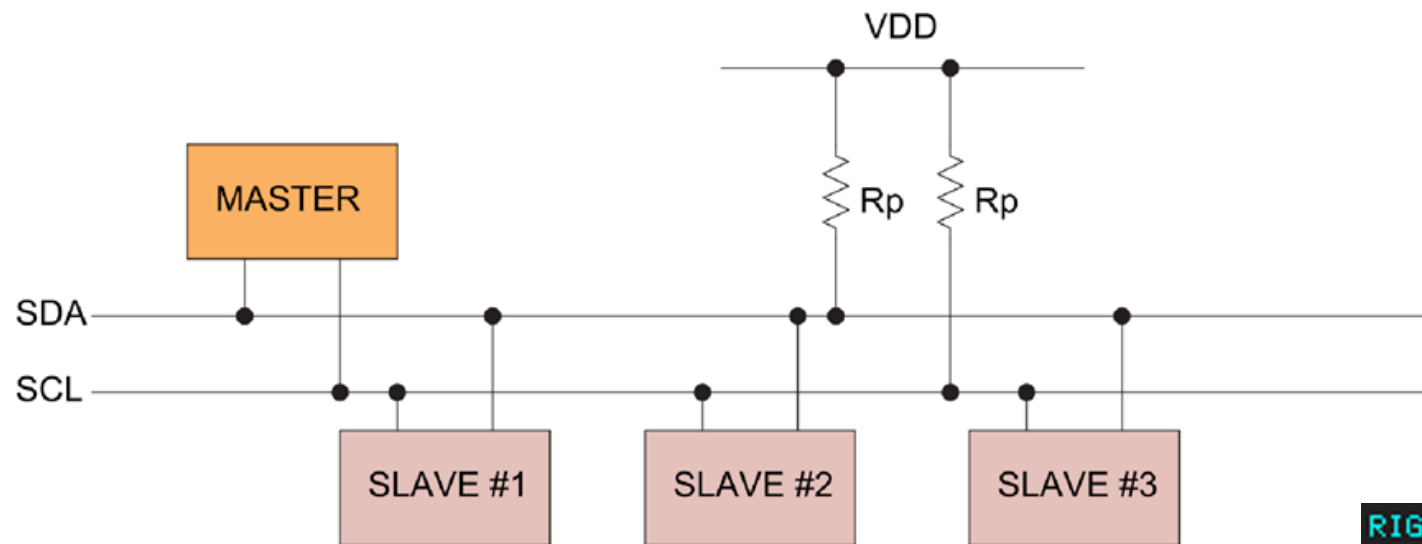
...але • **Quad SPI.**

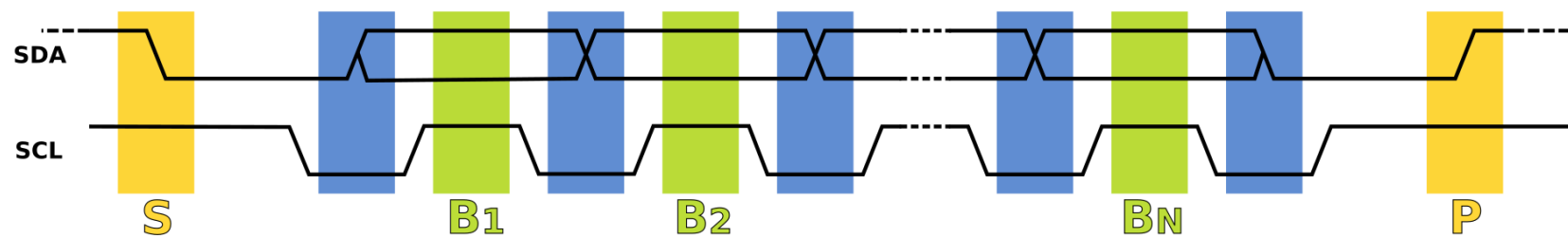
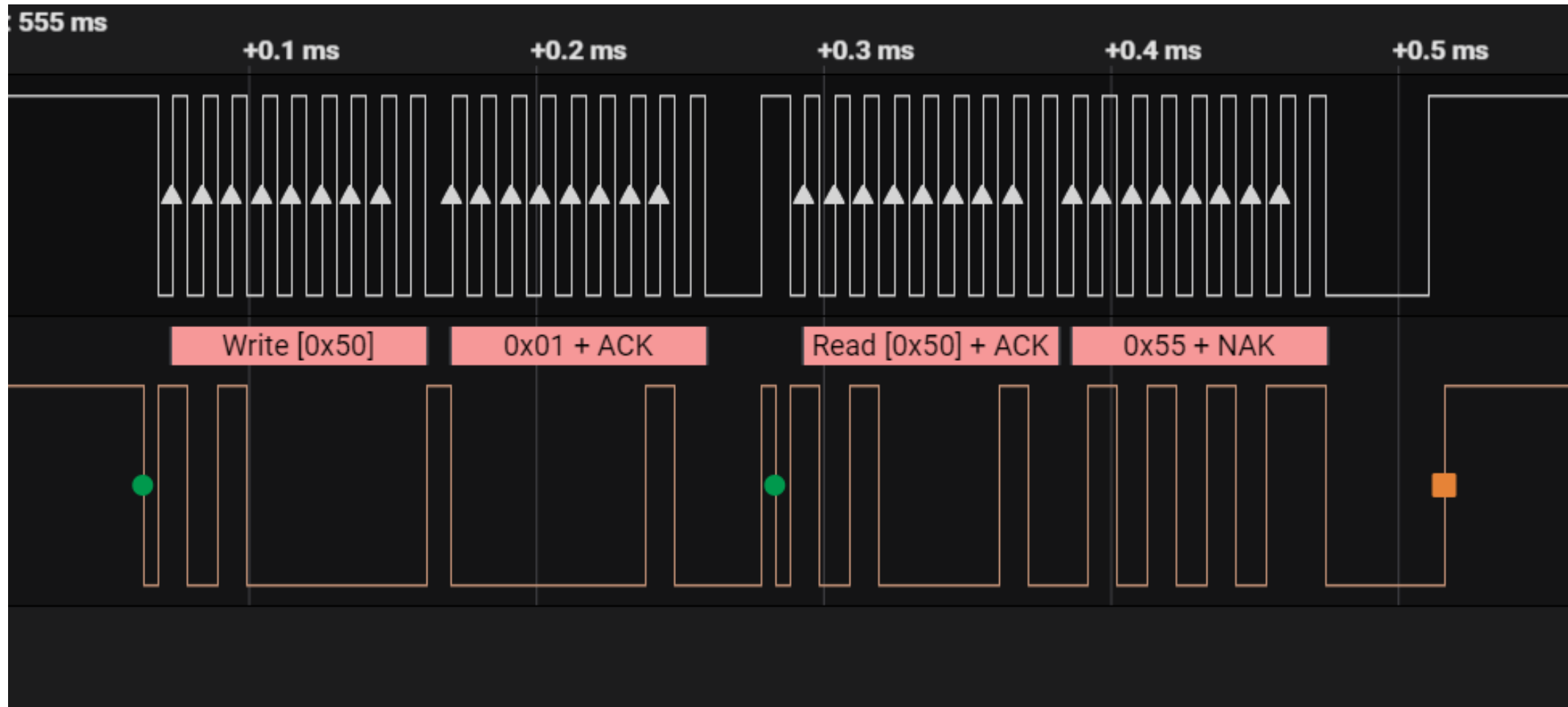


I²C



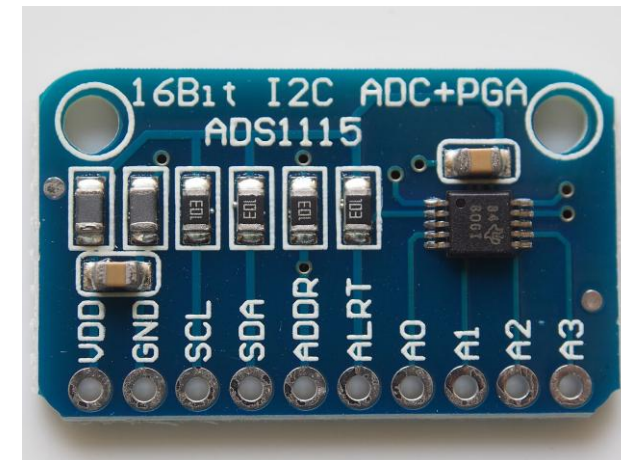
- **Inter-Integrated Circuit, I²C, I2C** (початок 80-х, Philips) метод двосторонньої комунікації між інтегральними схемами в електронних схемах. Послідовна шина I2C складається з двох ліній – тактової SCL (кожен імпульс - передача біта) та лінії даних SDA.
- Одна схема, підключена до шини, виконує роль контролера (~~master~~ controller), генерує тактовий сигнал і керує комунікацією (через передачу команд ініціації/завершення передачі, відправлення команд до окремих схем), тоді як інші компоненти працюють у режимі ~~slave~~ target, тобто їх поведінка визначається інструкціями, що надходять від контролера. У деяких випадках target є лише приймачем (наприклад, дисплей), в інших (наприклад, аналого-цифрових перетворювачах) ці одиниці передають дані через шину, але тільки у відповідь на певну команду центральної схеми і відповідно до власного протоколу (точно визначаючи кількість доступних бітів), який виробники детально описують у документації.
- Кожен target, підключений до шини, має унікальну 7-бітову адресу, яка дозволяє йому розпізнавати, чи команди, що надсилаються контролером, адресовані йому. Отже, такий метод комунікації дозволяє підключити понад 100 схем. Часто користувачеві залишають певний вибір щодо адреси, вибирають її шляхом з'єднання певних виводів з масою або лінією живлення схеми. Завдяки цьому до однієї шини можна підключити кілька, а навіть кілька десятків елементів одного типу, але деколи це ускладнено.
- Кількість елементів на шині обмежується загальною ємністю 400 pF, для стандартної швидкості 100kbit/sec, та 100 pf для швидкості 3.4 Mbit/sec
- Існує ультрашвидкий 5Mbit/sec режим але він трохи відрізняється від інших і не є цілком сумісним.





Адресація I2C

- Кожен пристрій, підключений до шини, програмно адресований за унікальною адресою. Для вибору приймача повідомлення передавач використовує адресу у форматі посилки.
- Використовується 7-бітна адресація.
- Процедура адресації на шині I²C полягає в тому, що перший байт після сигналу СТАРТ визначає адресу для проведення циклу обміну.
- Перші сім бітів першого байта утворюють адресу веденого. **Восьмий, молодший біт, визначає напрям пересилання даних.** «0» означає, що controller буде записувати інформацію до обраного target. «1» означає, що controller буде зчитувати інформацію з target.



Програмна реалізація I2C є тут:
<https://en.wikipedia.org/wiki/I2C>

Подібні протоколи

- **SMBus**
- **I3C**
- **I2S** Схожа, але несумісна з I²C шина I²S, яка розроблено для передачі стереофонічного звуку (*Sound*) у безвтратному форматі PCM. Це метод, що використовується в багатьох мультимедійних пристроях, від смартфонів до HiFi підсилювачів. Він базується на трьох лініях (одна з них визначає, до якого каналу адресовані дані) та опціонального додаткового тактового сигналу, що служить для підтримання правильної частоти відтворення зразків.

- **Послідовний**
- **Присутнє підтвердження прийому**
- **Присутній flow-control**
- **Синхронний**
- **Controller-target, так само як Main-sub**
- **Довжина слова 8-біт**
- **Адресована, через це не потрібно CS.**

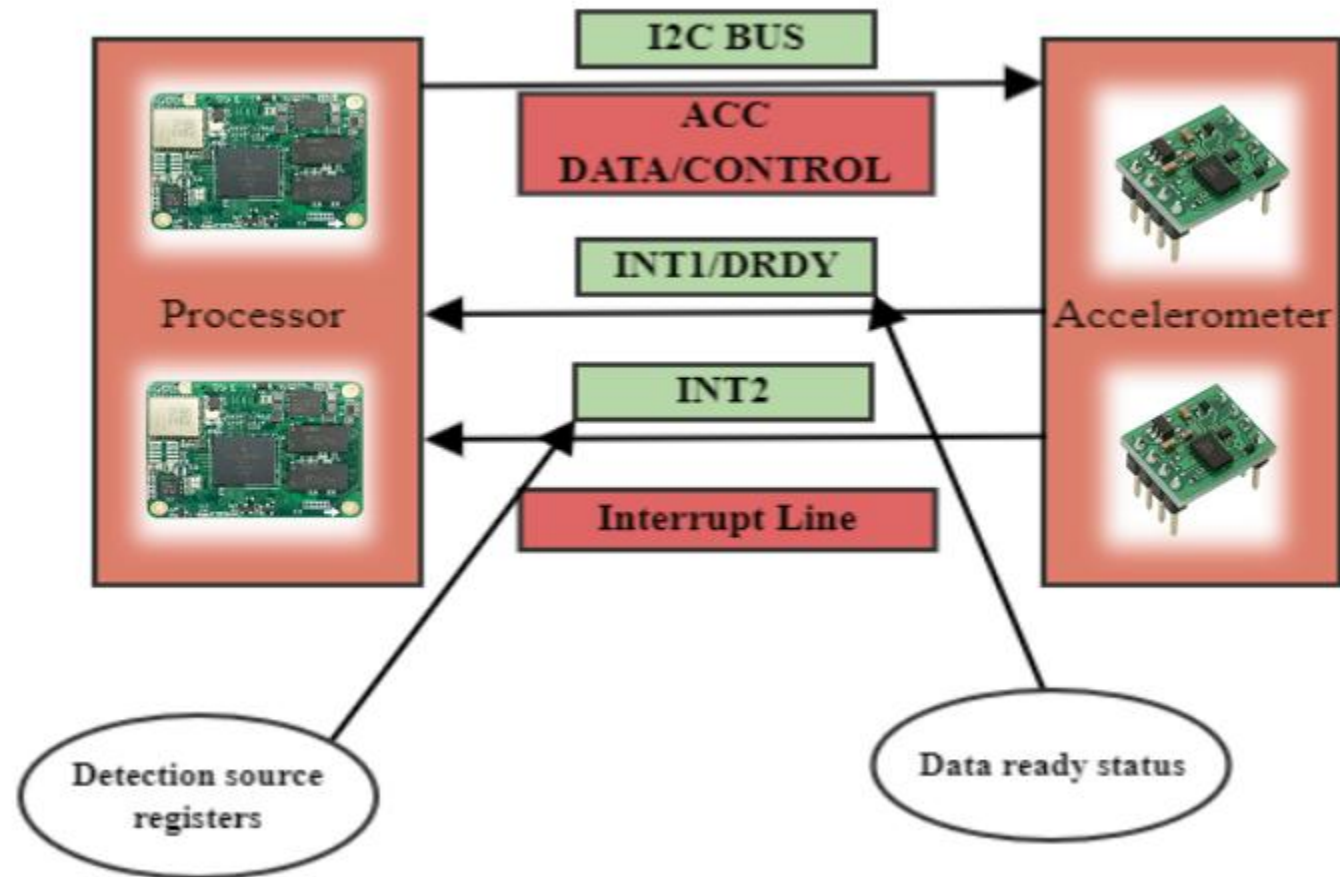
- **Повільніша за SPI, через pull-up “1”**
- **Відсутній контроль помилок**
- **Деколи бракує адрес для однотипних пристроїв на шині**

Що робити якщо sub/target хоче
про щось повідомити
main/controller ?

В такому випадку використовуються додаткові лінії котрі спричиняють до переривання і таким чином сигналізують main/controller про те що щось відбулося.

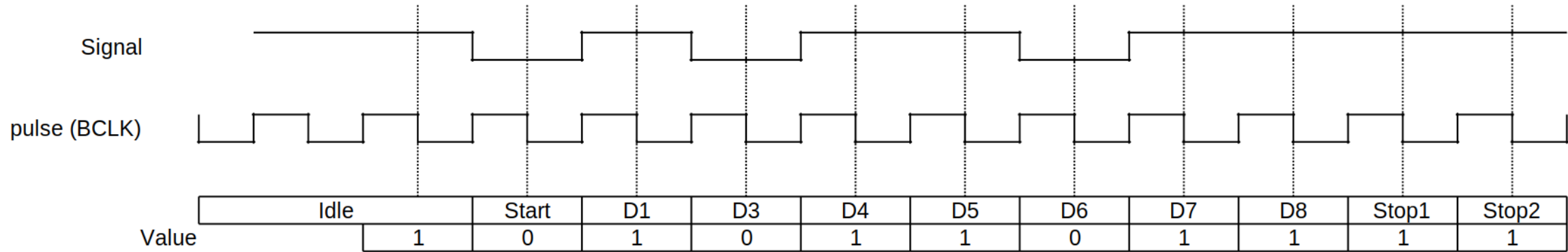
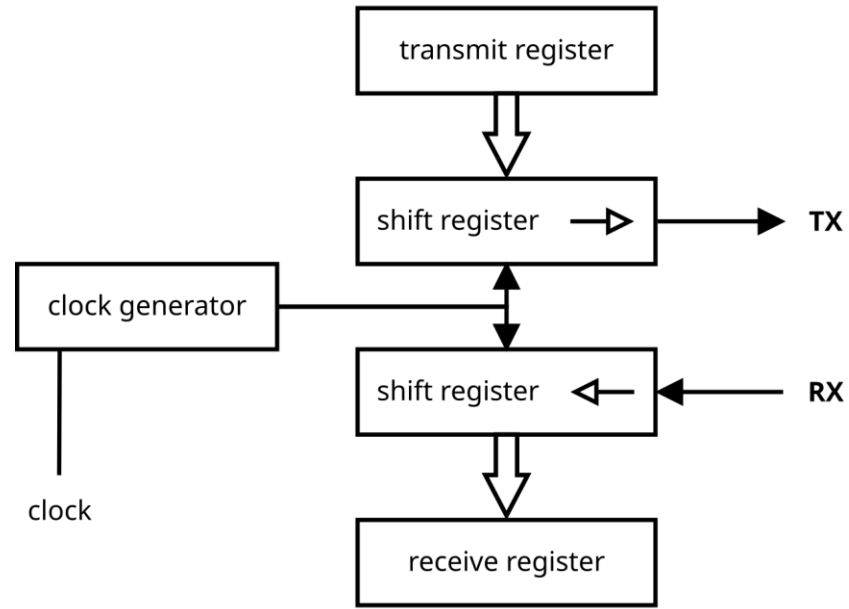
Додаткові лінії interrupt ніяк не промотуються або лімітуються стандартами I2C / SPI.

Додаткові лінії переривань часто присутні в пристроях котрі є джерелом даних для мікроконтролера.



UART

- UART є *full duplex* інтерфейсом, однак використовує лише дві лінії: передавальну Tx і приймальну Rx. UART не передбачає сигналу тактування. Це випливає з фундаментальних засад UART, тобто *Universal Asynchronous Receiver-Transmitter*. Тут з'являється питання асинхронної передачі.
- Як I²C, так і SPI є інтерфейсами **синхронними**, тобто передбачають наявність тактового сигналу, що синхронізує **комунікацію**, визначаючи її швидкість і поведінку підключених до схеми пристроїв. У випадку асинхронної передачі **передача ведеться з використанням повідомлень ініціації та закінчення**, а її швидкість визначається вже при ініціації: у коді програми, що виконується мікроконтролером, або в налаштуваннях апаратного пристрою.
- UART є найстарішим з представлених рішень, а його засади до сьогодні використовуються в цифровій електроніці – вони також є в основі таких стандартів, як USB та Ethernet тощо.
- Наявність інтерфейсу UART не передбачає використання конкретного протоколу. В той же час багато протоколів є по суті протоколами вищого рівня з нижчим логічним рівнем передачі UART. Наприклад DMX512, RS485 чи MODBUS.
- Також UART є рівноправним з обох сторін з точки зору ініціації початку комунікації. Таким чином немає потреби повідомляти main/controller про те що пристрій потрібно «вчитати»

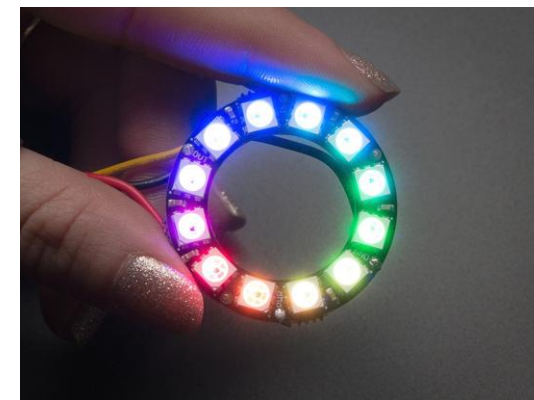
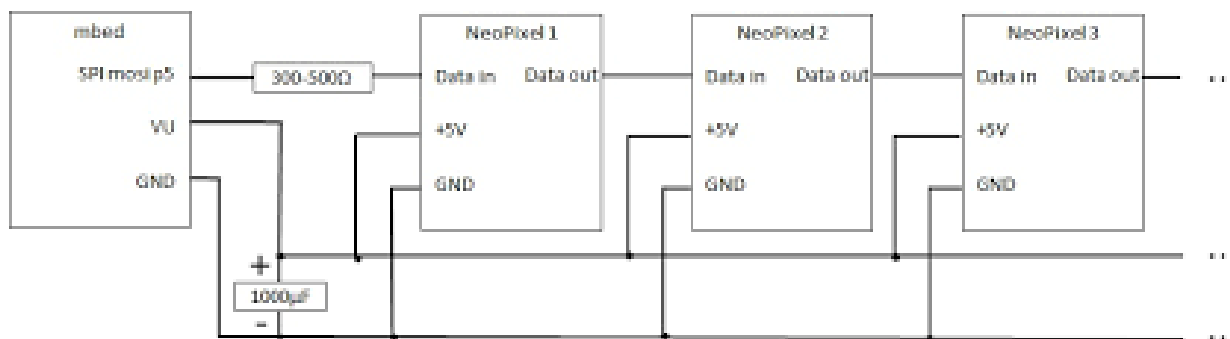


- **Послідовний**
- **Присутня мінімальна детекція помилок**
- **Асинхронний**
- **Симетричний**
- **Довжина слова 8-біт**

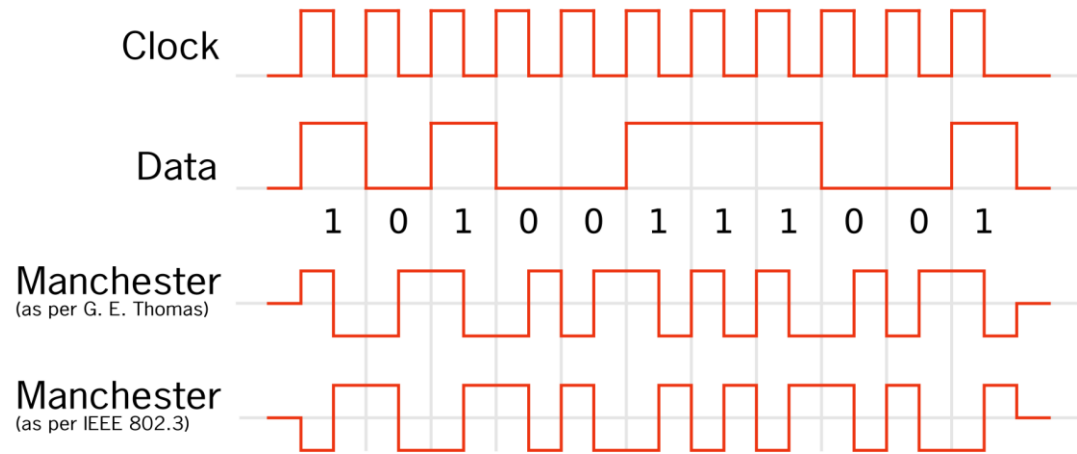
- **Flow control** можливий на рівні додаткових сигналів
- В базовому вигляді не передбачає більш одного «співрозмовника»

Чи це весь список - ні звичайно 😊

NeoPixel

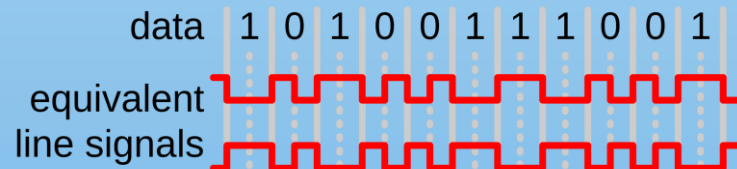


Манчестерський КОД



- **Манчестерський код** це двійковий код без постійної складової, в якому значення кожного переданого біта визначається напрямком зміни логічного рівня в середині обумовленого заздалегідь часового інтервалу.
- При манчестерському кодуванні кожен такт ділиться на дві частини. Інформація кодується перепадами потенціалу в середині кожного такту. Одиниця кодується перепадом від низького рівня сигналу до високого, а нуль — зворотним перепадом (за стандартом IEEE 802.3, хоча за Д. Є. Томасом кодування відбувається навпаки). На початку кожного такту може відбуватися службовий перепад сигналу, якщо потрібно представити кілька одиниць або нулів підряд. Оскільки сигнал змінюється принаймні один раз за такт передачі одного біта даних, то манчестерський код має здатність до самосинхронізації.

biphase mark code





Щиро дякую!