

Arduino Communication

APPS Summer Camp 2021

APPLIED
SCIENCES
FACULTY.CS

Bohdan Hlovatskyi. Karyna Volokhatiuk



Pins for common communication protocols



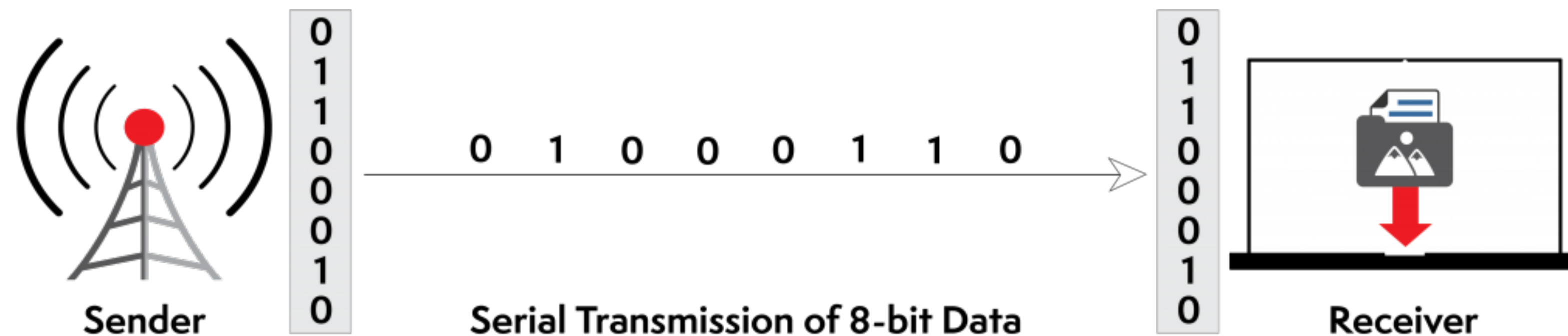
Паралельна передача даних

- Потребує кілька ліній для передачі даних
- Швидка передача даних
- Дорога
- Використовується на коротких відстанях



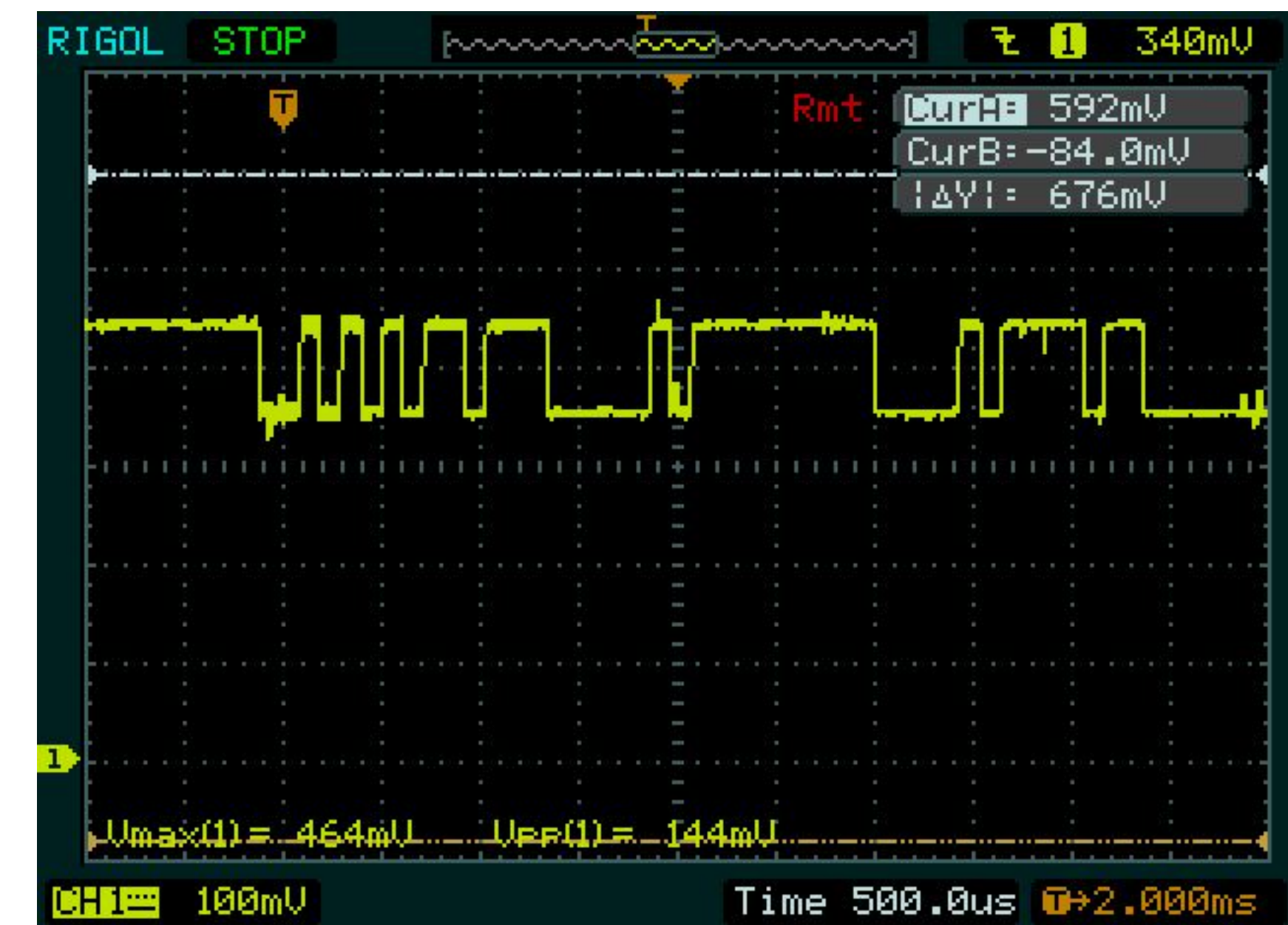
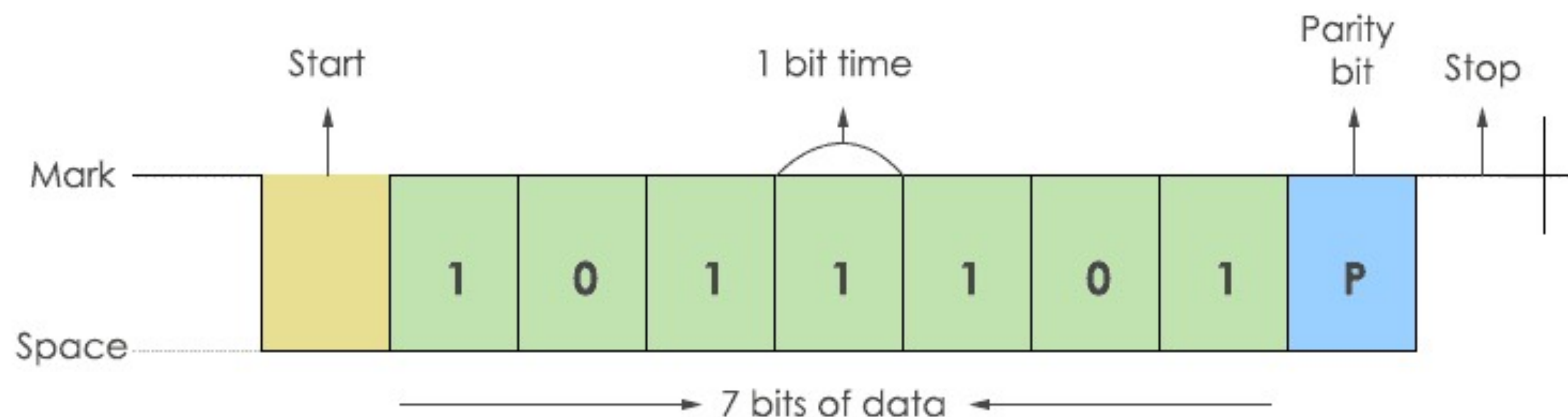
Serial передача

- Може використовуватись на довгих відстанях
- Дозволяє комунікацію один-до-багатьох тощо.
- Дешева та ефективна



Asynchronous Serial Transmission. UART

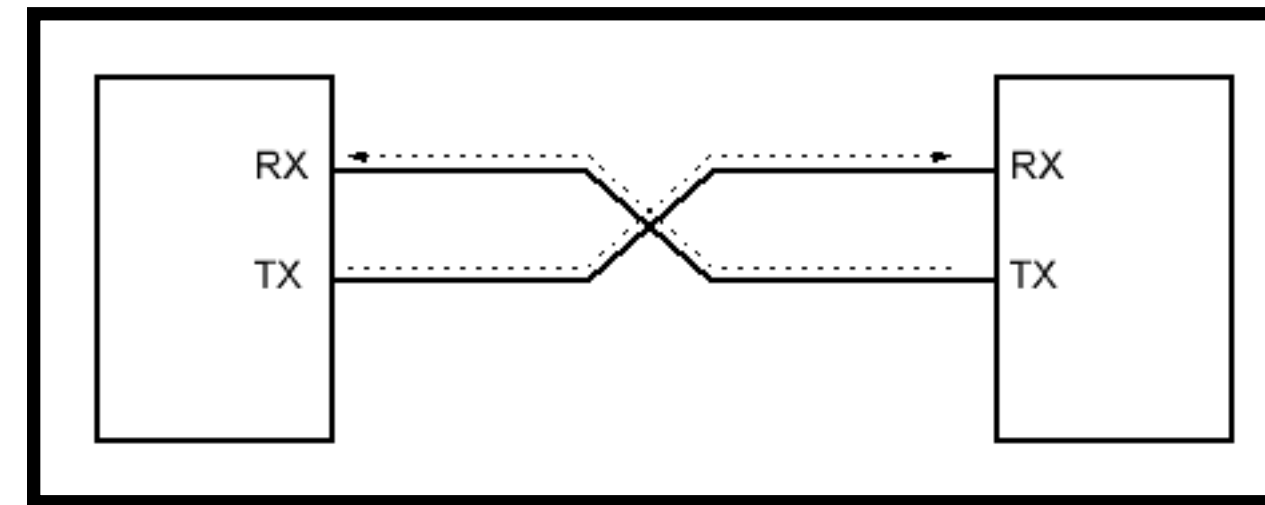
- кожна послідовність складається із **стартового біта**, за яким слідує вісім біт даних (bits 0-7), і один **стоп біт**.
- щоб знати, який проміжок між двома сигналами, два пристрої повинні домовитись про **baud rate** (В контексті Arduino, 9600 baud rate, означає, що система зможе переслати щонайбільше 9600 бітів).



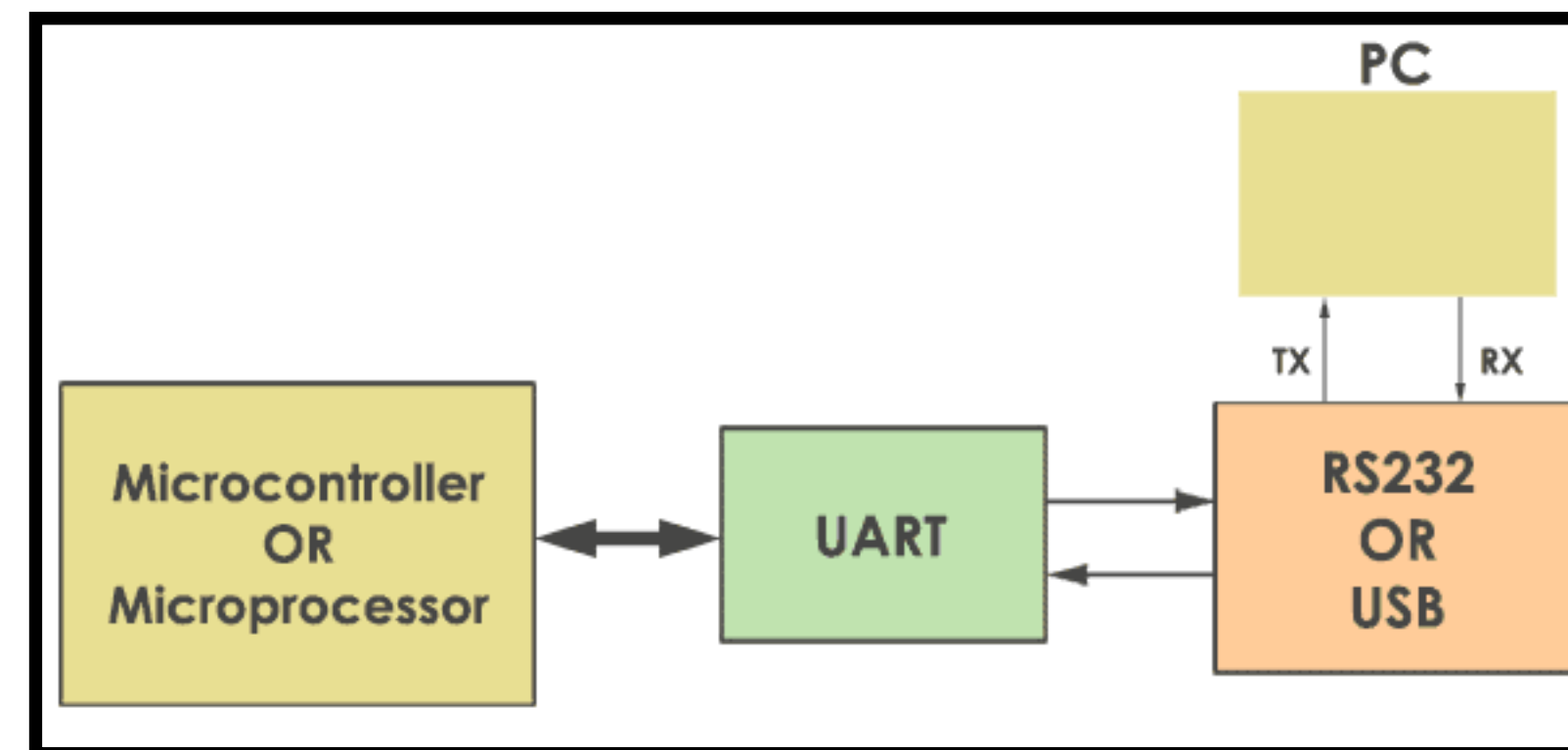
- Найбільш поширений та простий тип зв'язку між двома девайсами

- Повільний

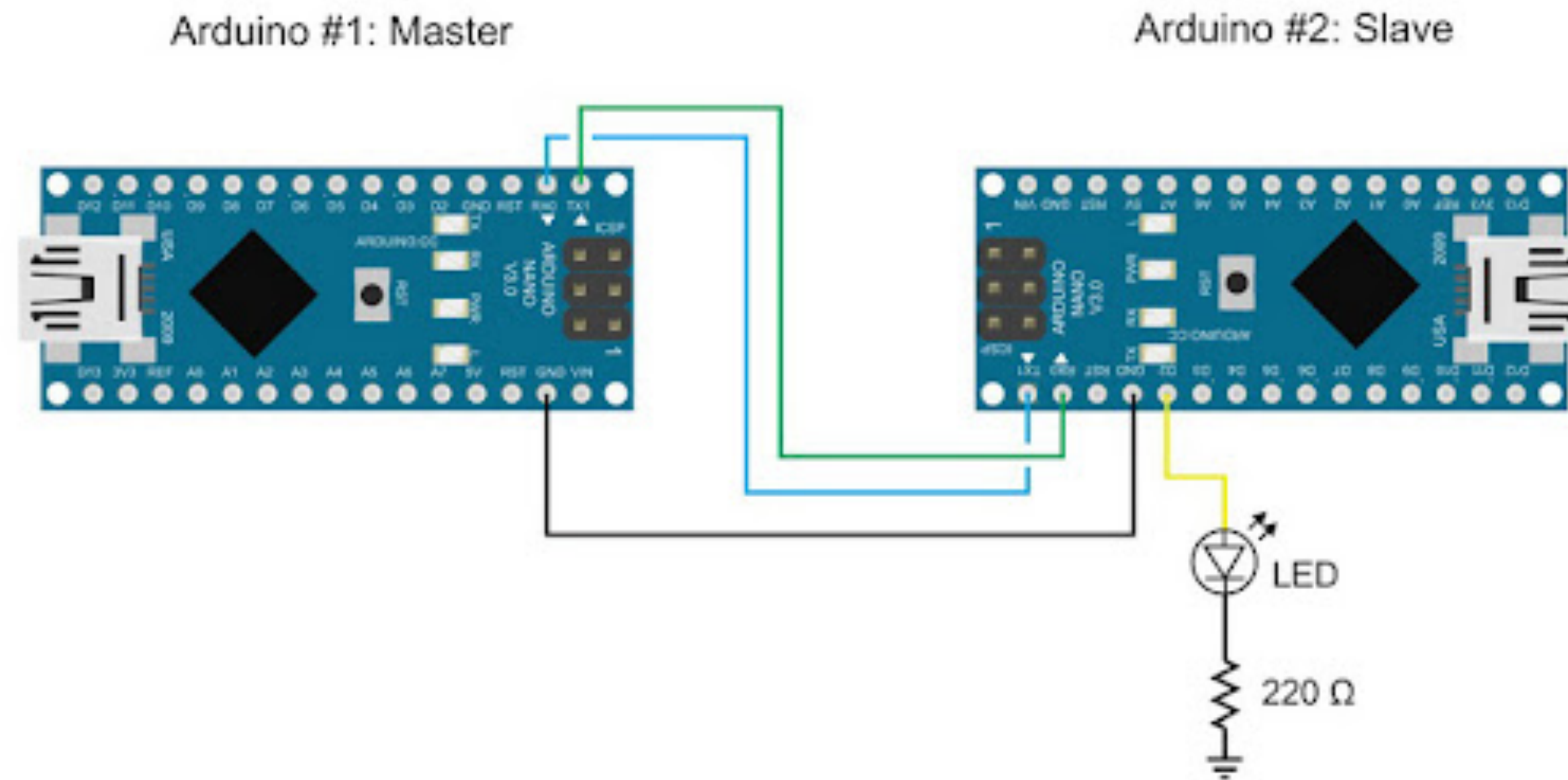
- Схема підключення:



- ! Використовується зокрема при комунікації комп'ютера з мікроконтролером*



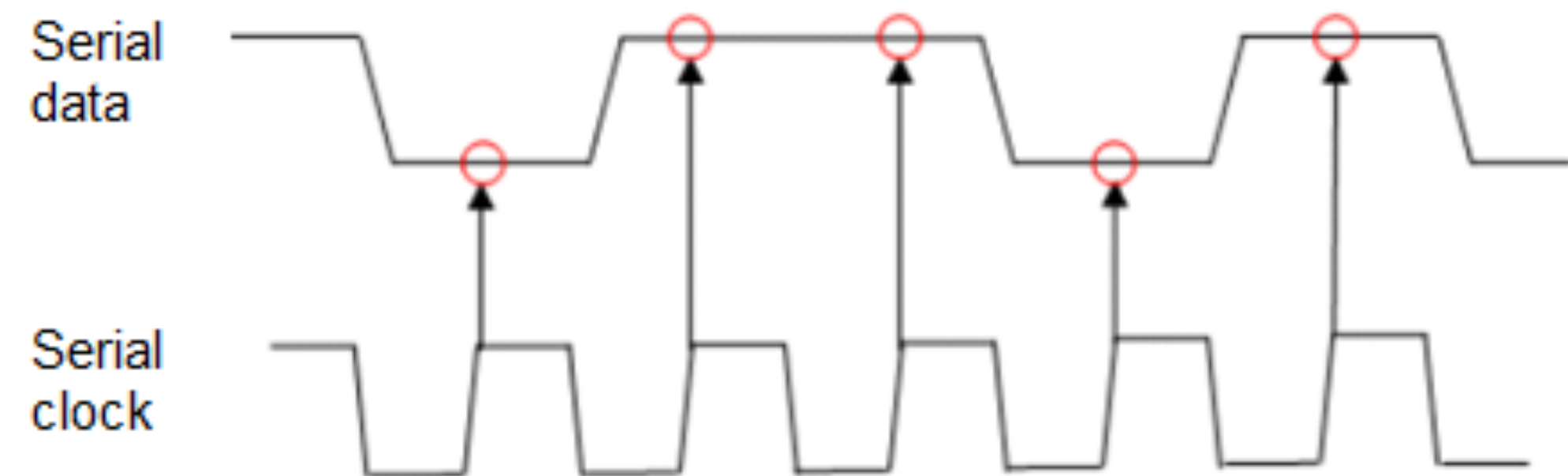
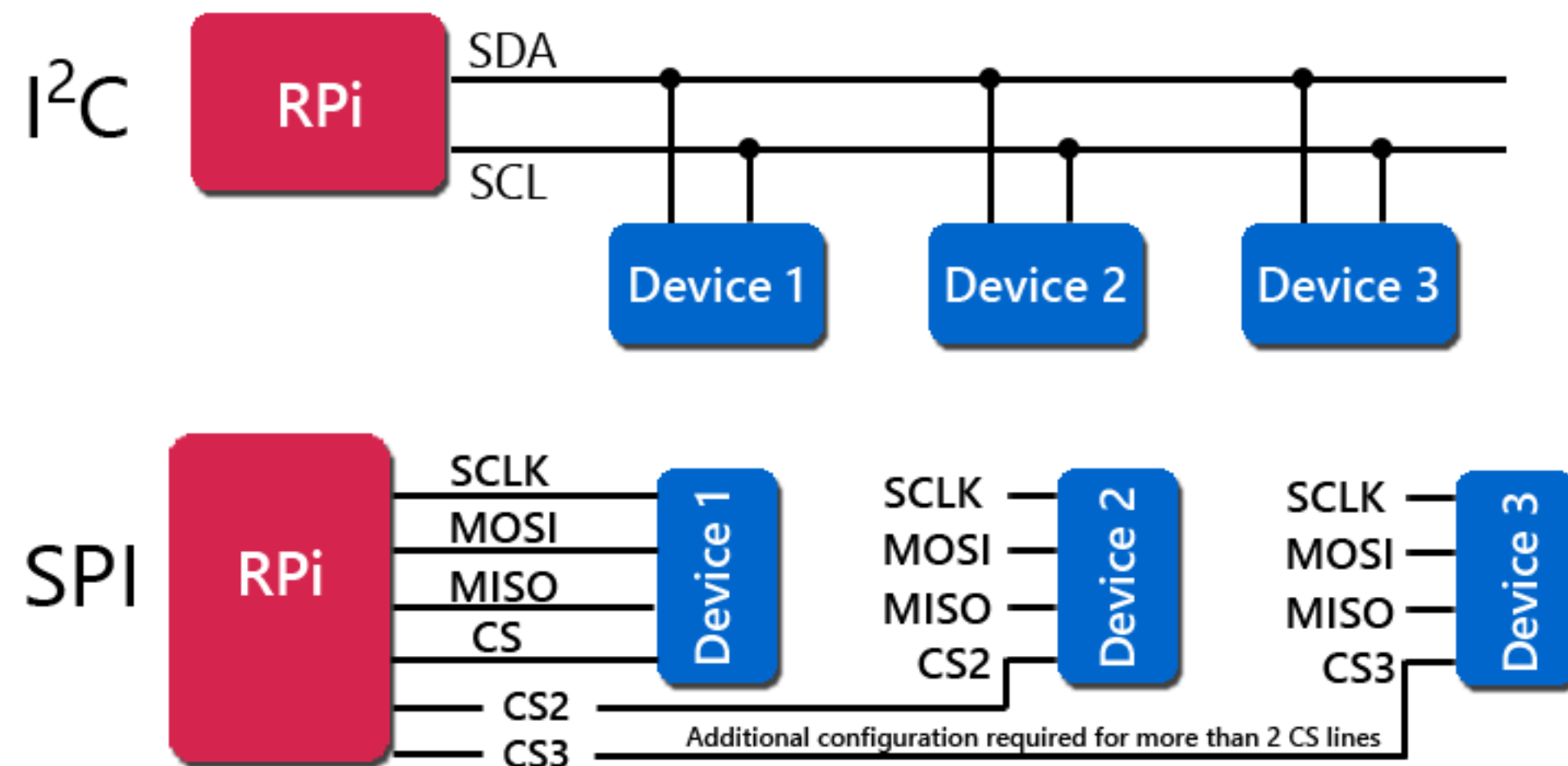
Приклад використання UART*



Дещо навчальний, втім в реальному житті таке часто потрібне: для прикладу, використати ще один контролер з доступом до інтернету, на яких будуть приходити дані та які він відсилатиме на веб-сервер.

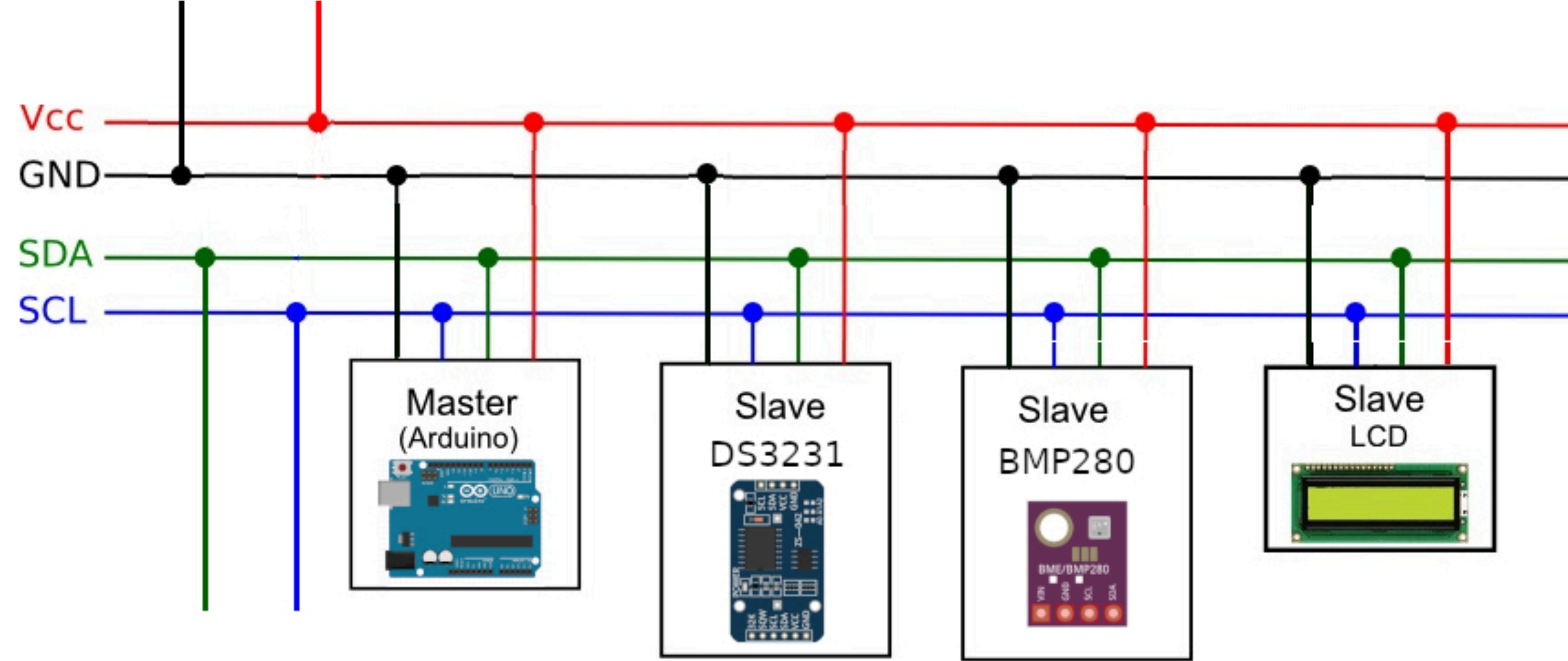
Synchronous Serial Transmission.

Безперервний потік сигналів даних супроводжується сигналами синхронізації (генерованими clock-ом для забезпечення того, щоб передавач і приймач були синхронізовані).

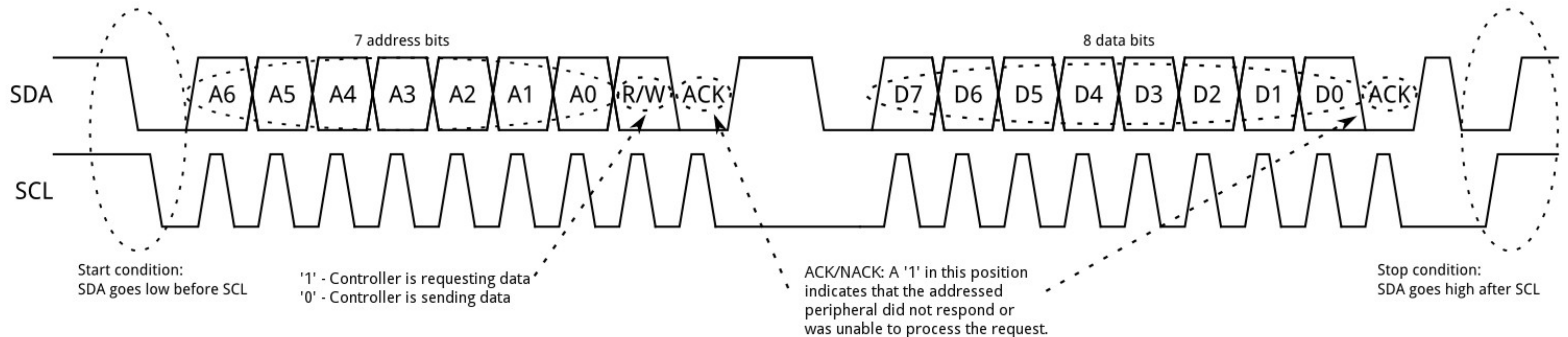


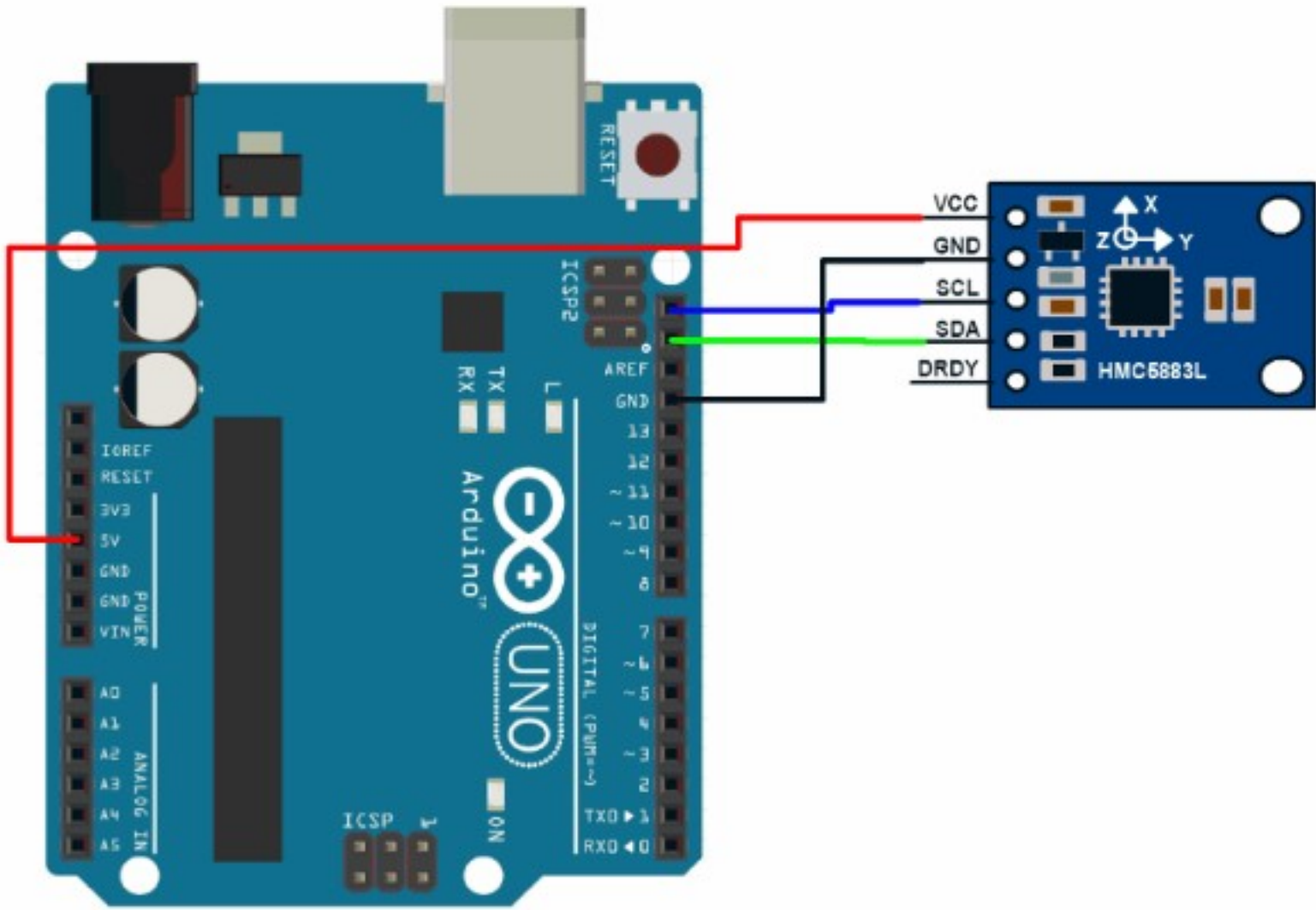
I2C

- складається з двох сигналів: **SCL** і **SDA**. SCL - тактовий сигнал, а SDA - сигнал даних.
- SCL завжди генерується Master-ом.
- Master також завжди ініціалізовує обмін.



- На лінії даних SDA він виставляє **адресу пристрою**, з яким необхідно встановити зв'язок (може підключити до **127** пристроїв з різними! адресами). Наступний біт послідовності - це **код операції** (читання або запис) і ще один біт - **біт підтвердження** (ACK)





Address Location	Name	Access
00	Configuration Register A	Read/Write
01	Configuration Register B	Read/Write
02	Mode Register	Read/Write
03	Data Output X MSB Register	Read
04	Data Output X LSB Register	Read
05	Data Output Z MSB Register	Read
06	Data Output Z LSB Register	Read
07	Data Output Y MSB Register	Read
08	Data Output Y LSB Register	Read
09	Status Register	Read
10	Identification Register A	Read
11	Identification Register B	Read
12	Identification Register C	Read

```
#include <Wire.h> //I2C Arduino Library

#define address 0x1E //I2C 7bit address of HMC5883

void setup(){
    //Initialize Serial and I2C communications
    Serial.begin(115200);
    Wire.begin();

    //Put the HMC5883 IC into the correct operating mode
    Wire.beginTransmission(address);
    Wire.write(0x02); //select mode register
    Wire.write(0x00); //continuous measurement mode
    Wire.endTransmission();
}

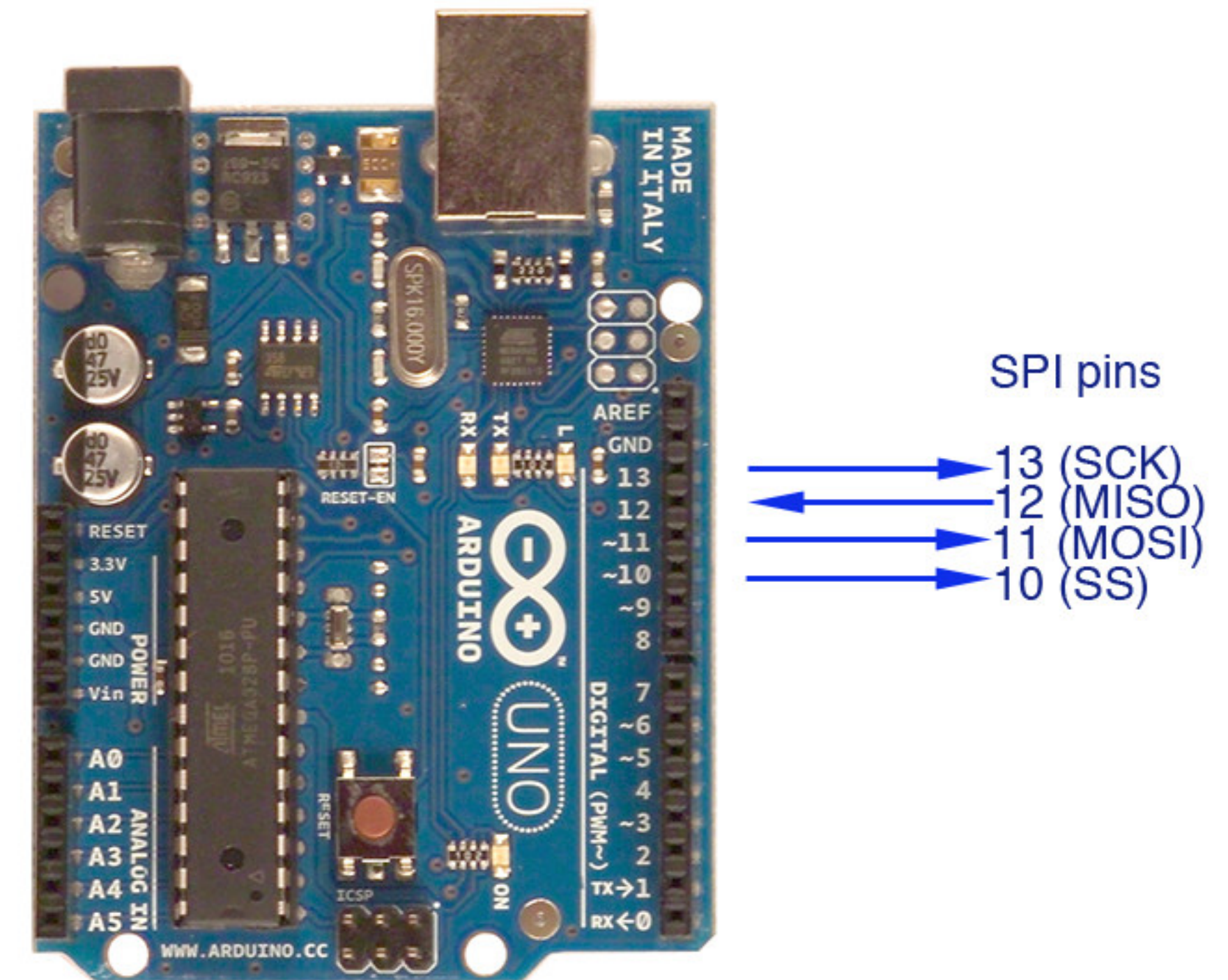
void loop(){
    int x,y,z; //triple axis data

    //Tell the HMC5883 where to begin reading data
    Wire.beginTransmission(address);
    Wire.write(0x03); //select register 3, X MSB register
    Wire.endTransmission();

    //Read data from each axis, 2 registers per axis
    Wire.requestFrom(address, 6);
    if(6<=Wire.available()){
        x = Wire.read()<<8; //X msb
        x |= Wire.read(); //X lsb
        z = Wire.read()<<8; //Z msb
        z |= Wire.read(); //Z lsb
        y = Wire.read()<<8; //Y msb
        y |= Wire.read(); //Y lsb
    }
}
```


SPI

- Послідовний синхронний інтерфейс передачі даних.
- Як головний, так і підлеглий можуть надсилати дані одночасно
- Пристрої SPI підтримують набагато вищі тактові частоти порівняно з інтерфейсами I2C

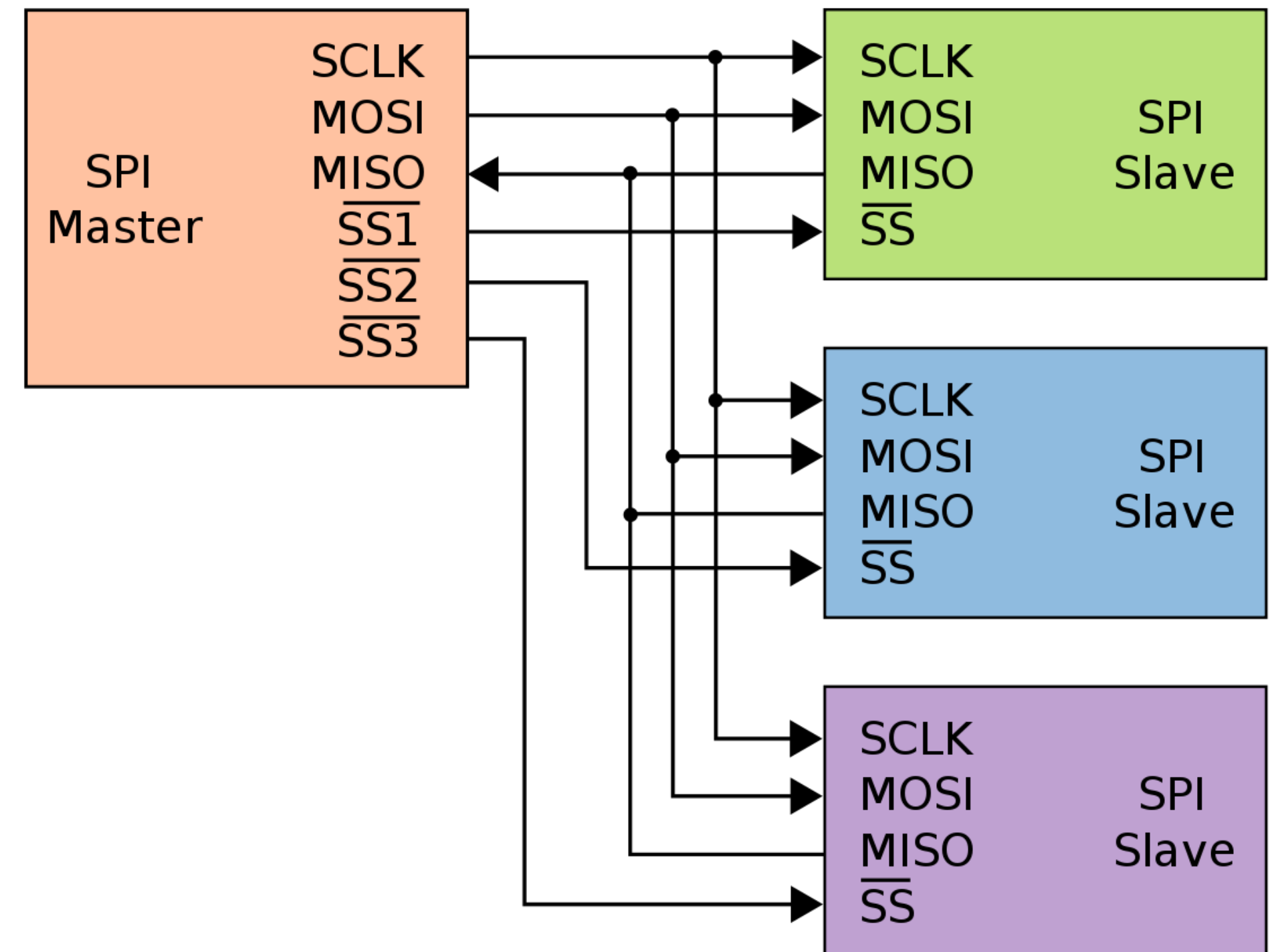


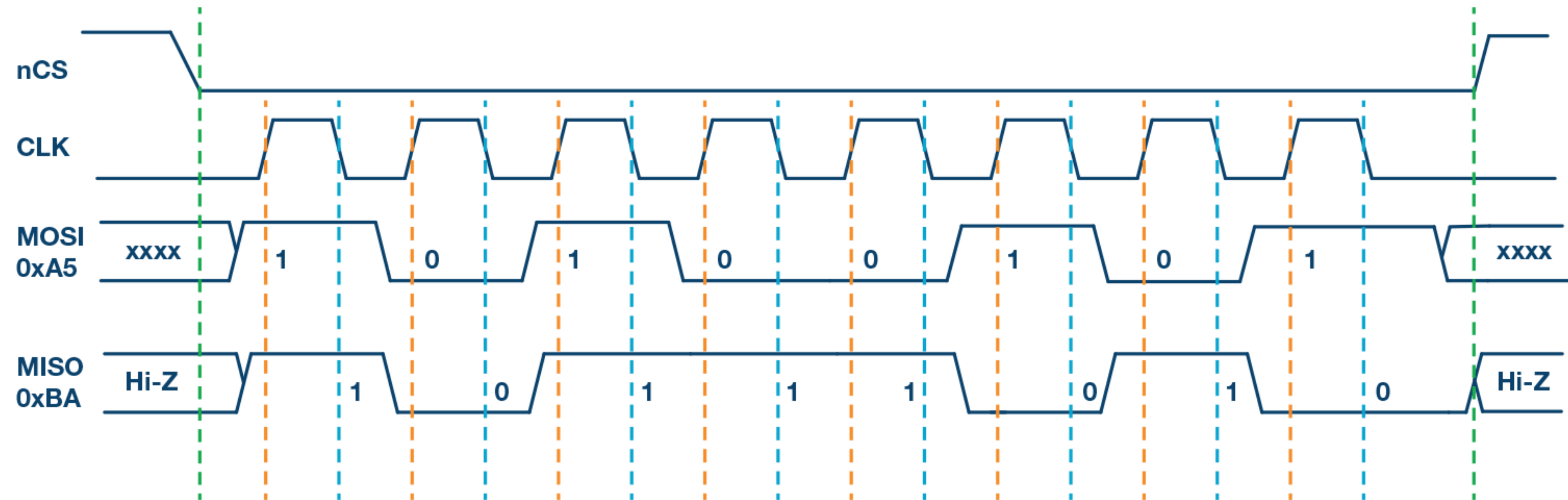
MISO (Master In Slave Out) - надсилання даних ведучому (Master мікроконтролеру)

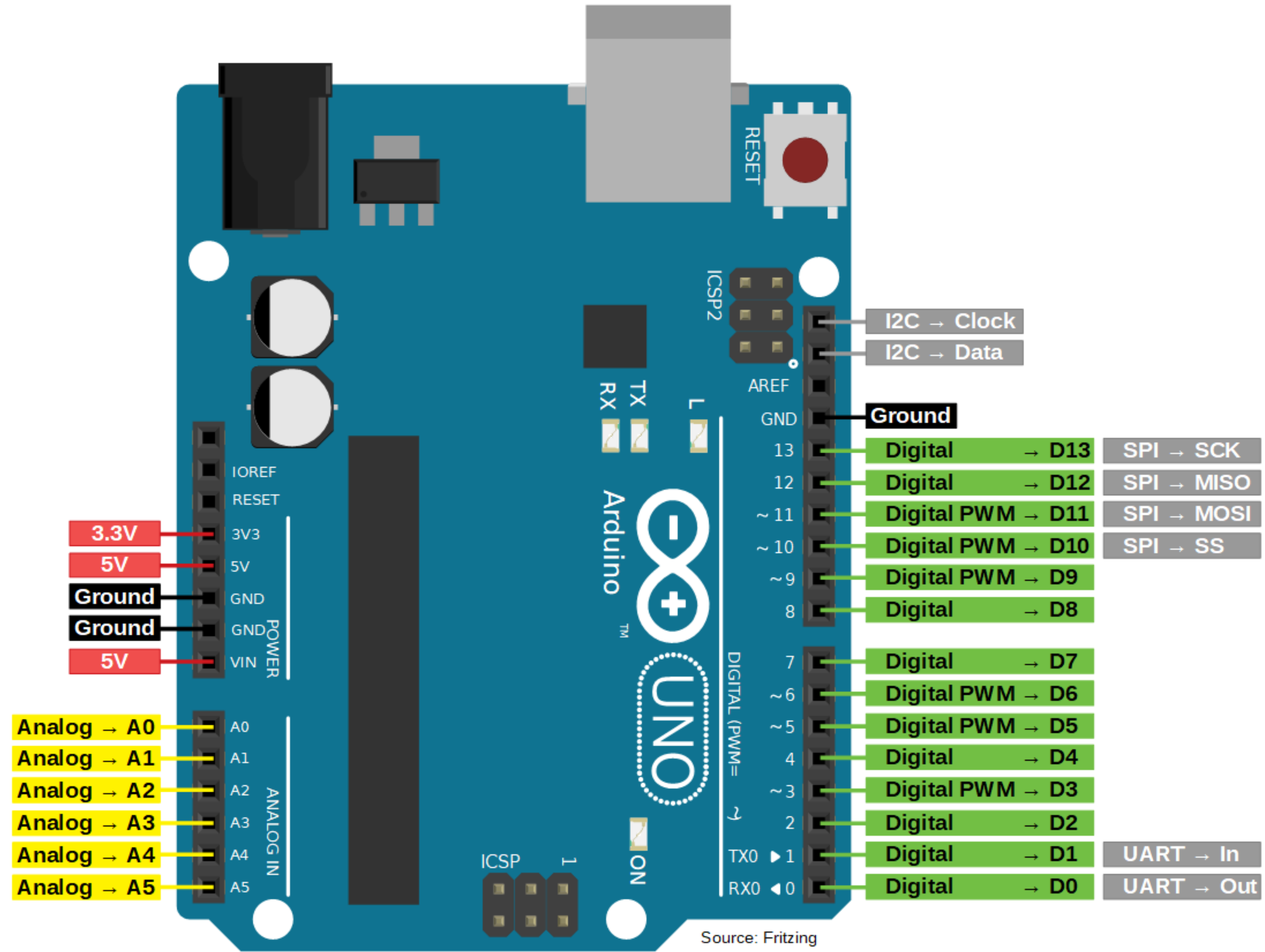
MOSI (Master Out Slave In) - головна лінія для відправки даних на периферію (сенсорам)

SCK (послідовний годинник) - тактові імпульси, які синхронізують передачу даних, що генерується ведучим

SS (Slave Select) - по одному для кожного Slave; Використовуються Master-ом для вибору пристрою, з яким він хоче взаємодіяти.







HW

- Підключити матрицю по I2C, вивести сердечко на ній
- Об'єднатись в команди та змусити дві ардуїнки “заговорити”
- Вітається креатив! Приклад: одна з ардуїнок працюватиме з пристроями керування: потенціометр, кнопка тощо; інша виводить щось на матрицю; взаємодія між ардуїнками - UART