

Arduino: sériová komunikácia

Základy internetu vecí

ÚINF PF UPJŠ

3. týždeň

DigitalWrite a DigitalRead

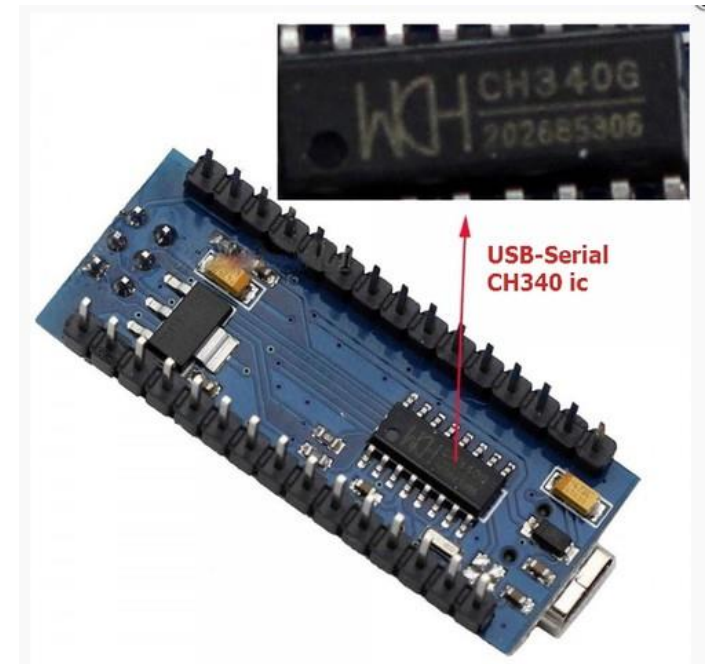
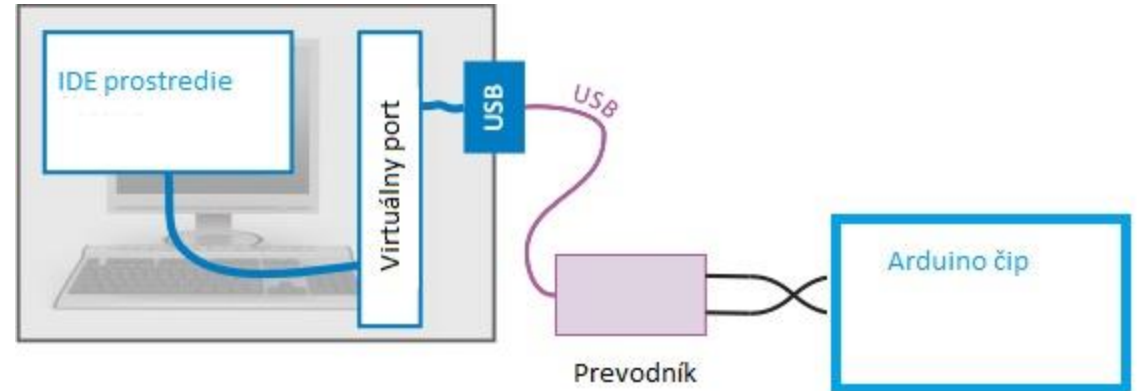
- nastavuje sa pinMode
- **DigitalWrite**
 - OUTPUT mode: zapisuje HIGH (Nano +5V) a LOW (0V) hodnoty
 - INPUT mode: zapína (HIGH) a vypína (LOW) pull-up rezistor
 - na všetkých pinoch okrem A6, A7 (Nano)
- **DigitalRead**
 - INPUT mode: meria hodnotu HIGH alebo LOW
 - na všetkých pinoch okrem A6, A7 (Nano)

AnalogWrite a AnalogRead

- **AnalogWrite**
 - PWM, min. 0 (vždy vypnuté), max. 255 (vždy zapnuté)
 - INPUT mode: zapína (HIGH) a vypína (LOW) pull-up rezistor
 - Nano: digitálne piny s PWM: 3, 9, 10, 11 (490Hz) a 5, 6 (980Hz)
- **AnalogRead**
 - Nano: hodnota z rozsahu 0 – 1023 (rozlíšenie 10 bitov)
 - Analógové piny A1 – A7

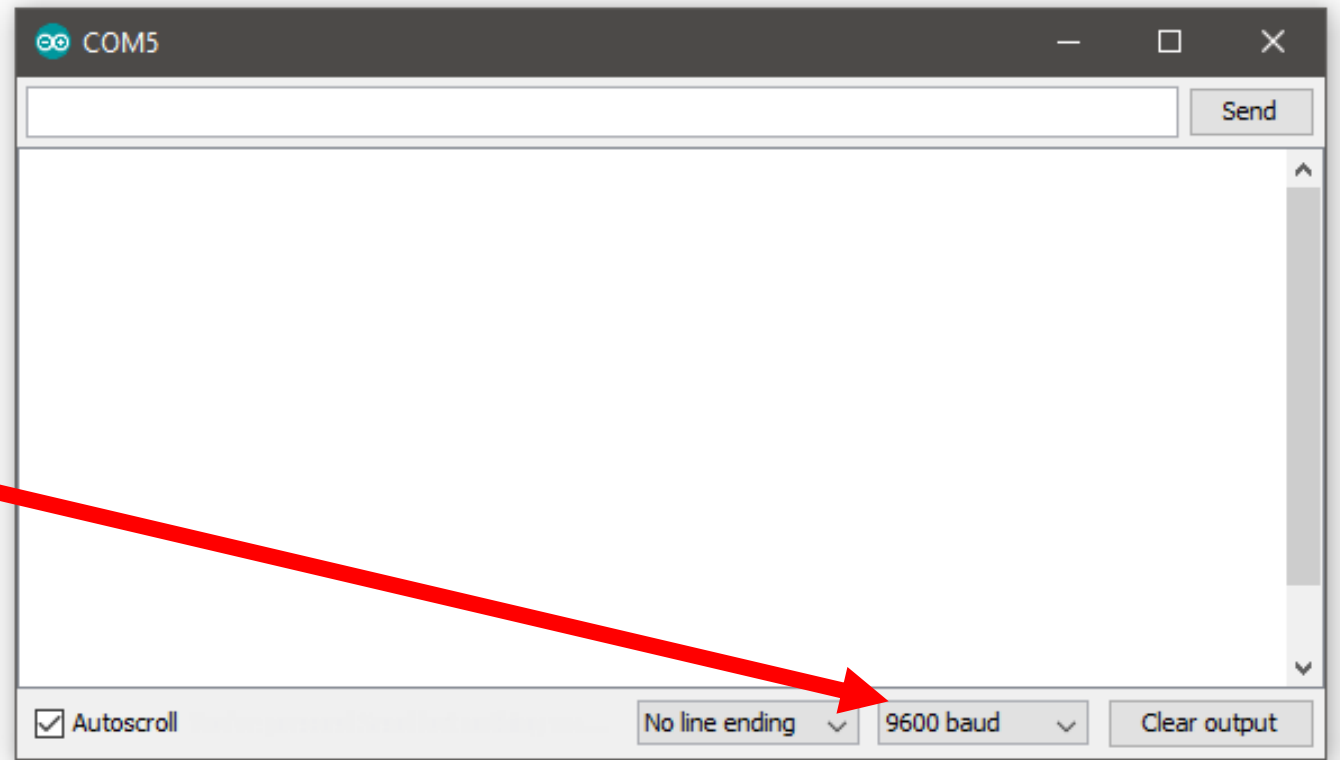
Seriová komunikácia

- Seriová komunikácia (UART)
- Piny **0** (RX) a **1** (TX)
 - nepracujte s nimi, ak neviete, čo robíte...
- 1 hardvérový seriový port (Arduino Mega má 4 porty)
- **Komunikačná rýchlosť** (baudrate) musí byť na oboch koncoch rovnaká
 - zvyčajne násobok 9600

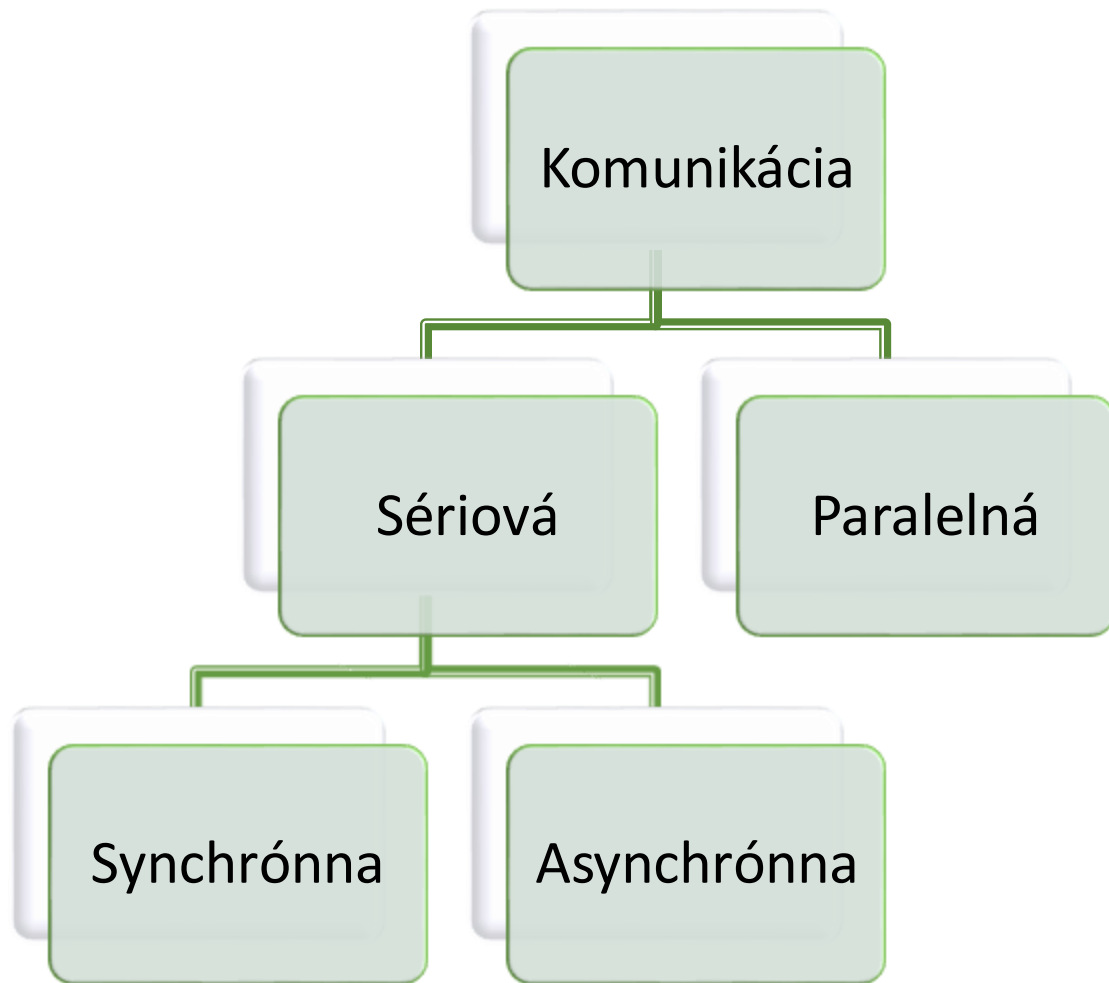


Seriová komunikácia – inicializácia a výstup

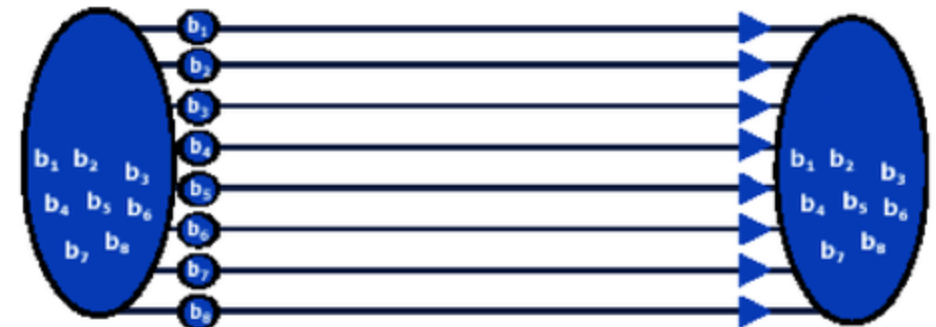
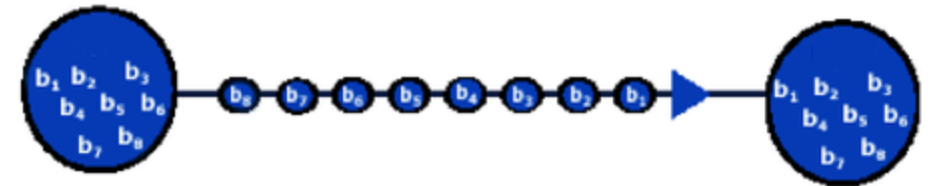
```
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    int c = 100;  
    Serial.println(c);  
    Serial.println("Zaklady IoT");  
}
```



Spôsoby komunikácie

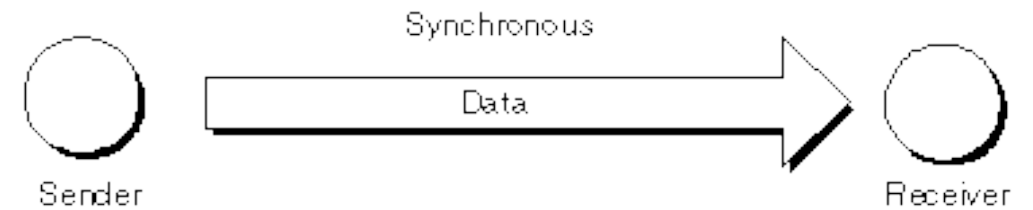
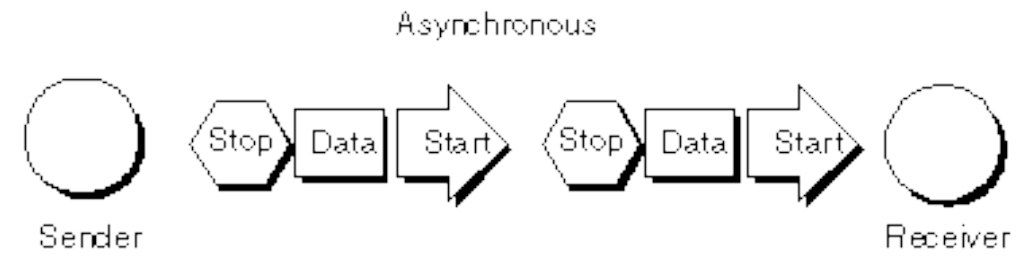
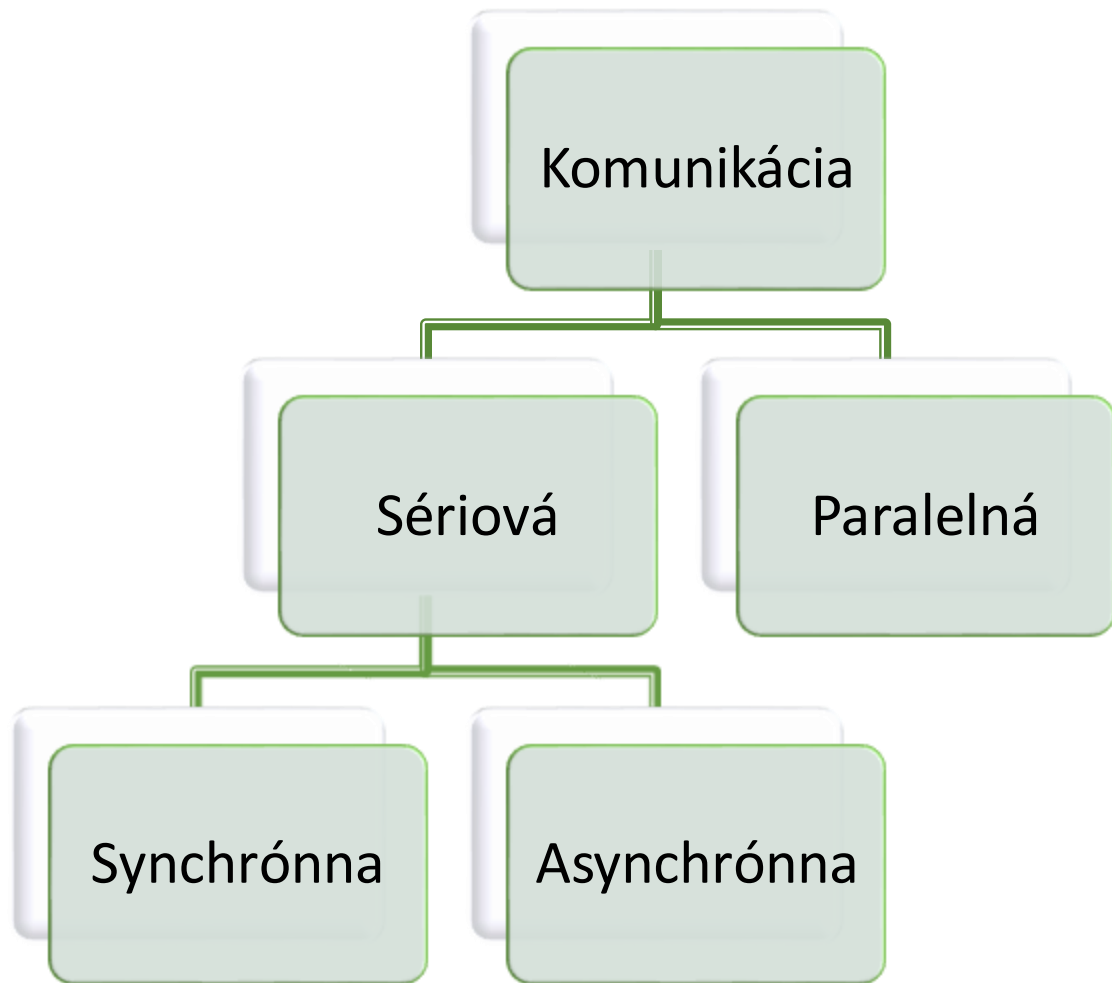


Sériová komunikácia =
v každom okamihu najviac 1 bit



Paralelná komunikácia =
viacero prenášaných bitov
v tom istom okamihu

Spôsoby komunikácie

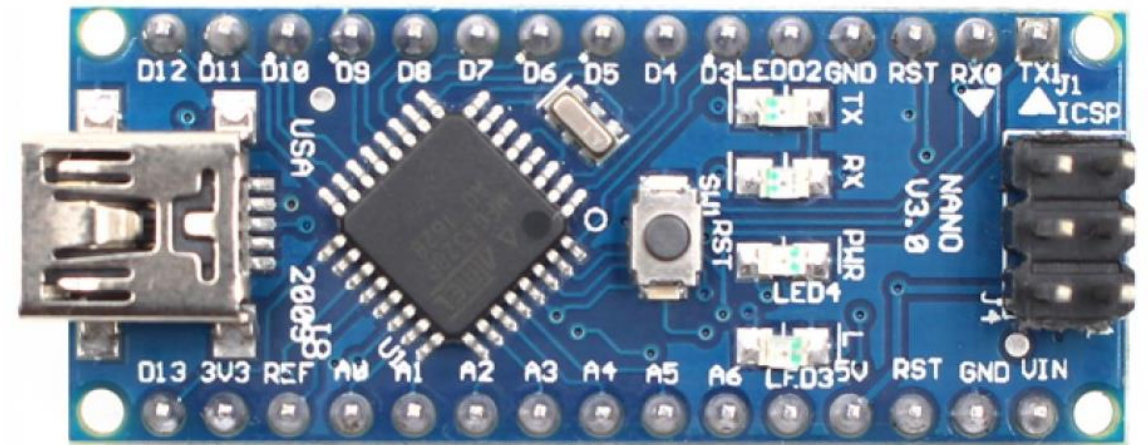
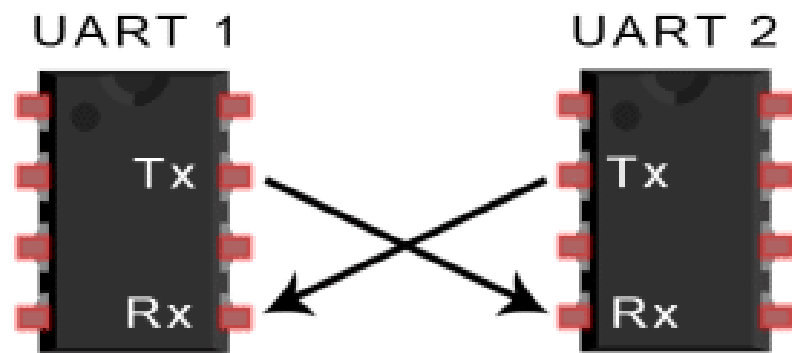
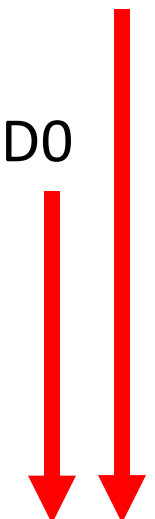


Asynchrónna komunikácia =
bez externého signálu zo spoločných hodín

UART

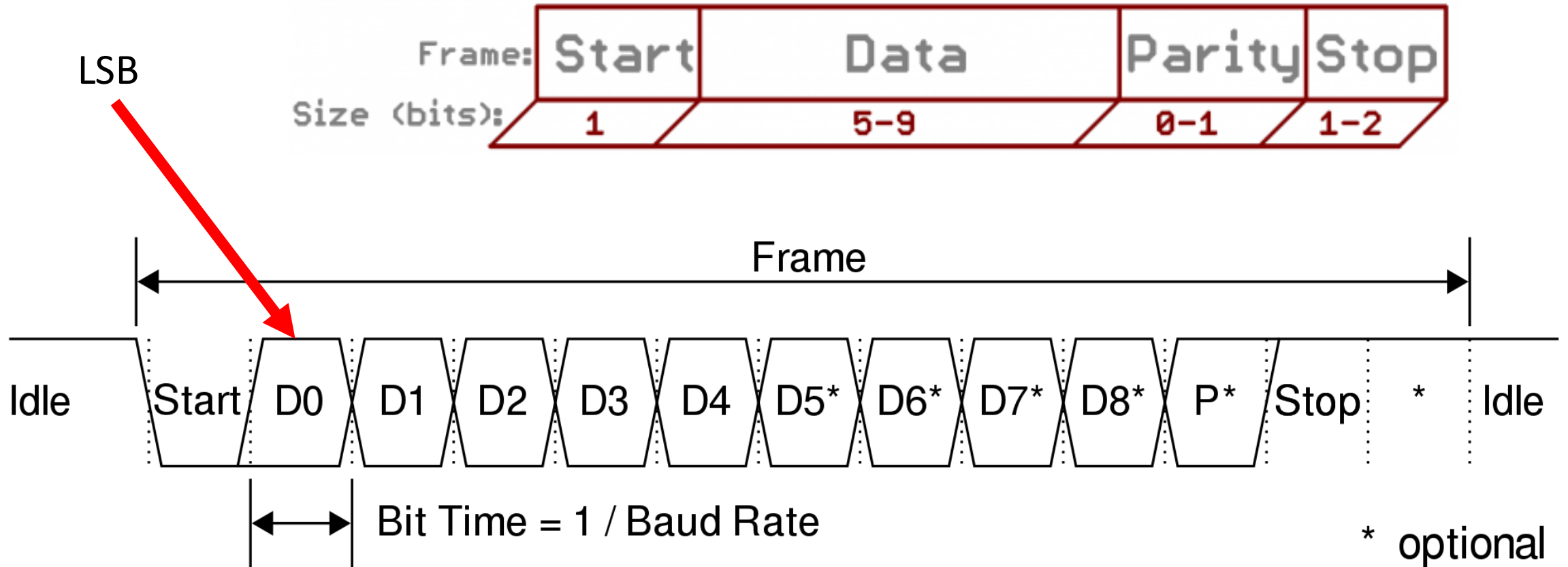
- UART = **U**niversal **A**synchronous **R**eceiver-**T**ransmitter
 - **hardvérové zariadenie**
 - konfigurovateľný dátový formát a prenosová rýchlosť
 - asynchrónna sériová komunikácia medzi 2 zariadeniami
 - **full-duplex** (možná komunikácia naraz v oboch smeroch)
 - RX, TX piny (+spoločná GND)

TX = D1
RX = D0



UART - protokol

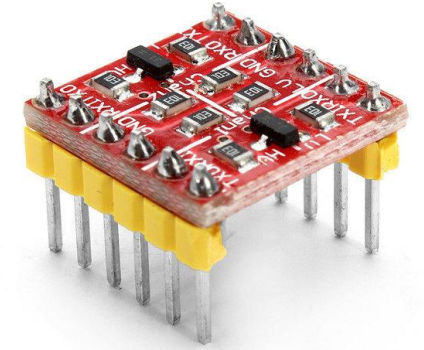
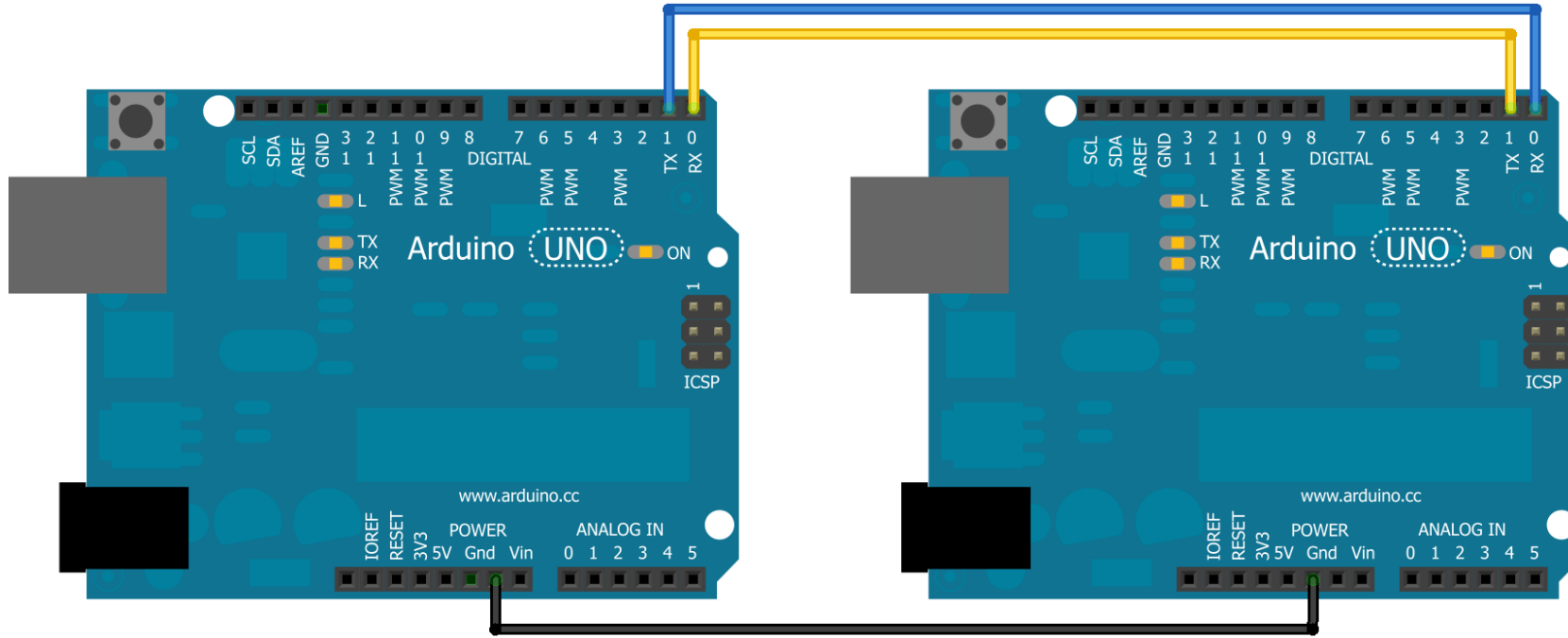
- Komunikácia prebieha v „rámcoch“ (frame-och)



UART - protokol

- Komunikácia prebieha v „rámcoch“ (frame-och)
- Komunikujúce strany **sa musia dohodnúť** na voliteľných parametroch protokolu:
 - **prenosová rýchlosť** (zvyčajne násobok 9600, 9600-115200)
 - **počet dátových bitov** (zvyčajne 8)
 - **použitie parity bitu** – párny vs. nepárny súčet (zvyčajne 0)
 - **počet stop bitov** (zvyčajne 1)
 - Linux: `stty`

Komunikácia Arduino-Arduino (UART – UART)

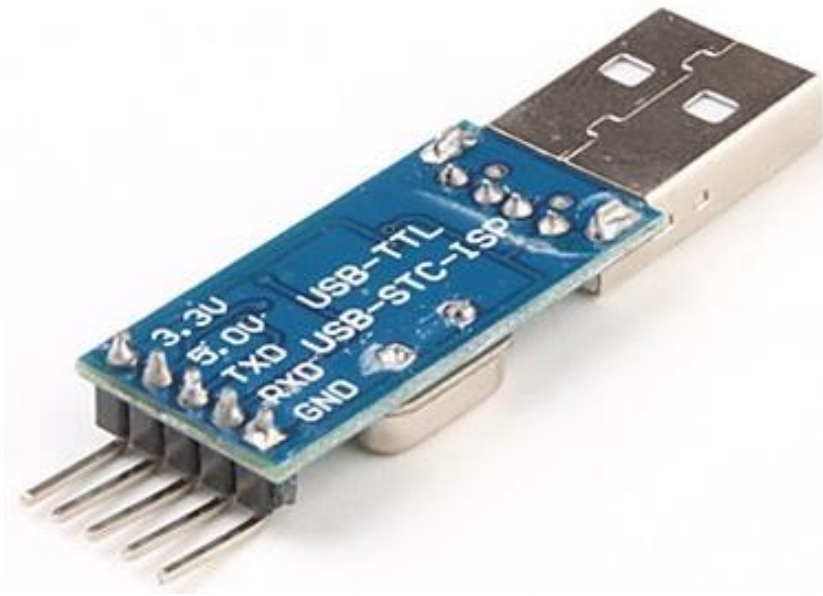
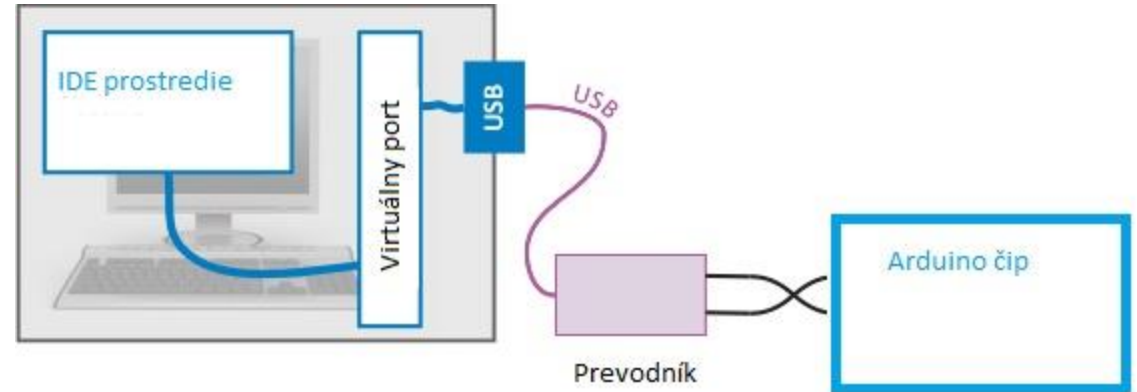


Konvertor 3.3V na 5V

Prepojené Arduina, resp.
mikrokontroléry musia mať rovnaké
operačné napätie!

Komunikácia Arduino (UART) – PC (USB)

- Väčšina Arduino zariadení má USB-Serial prevodník
- Inak USB-TTL prevodník:

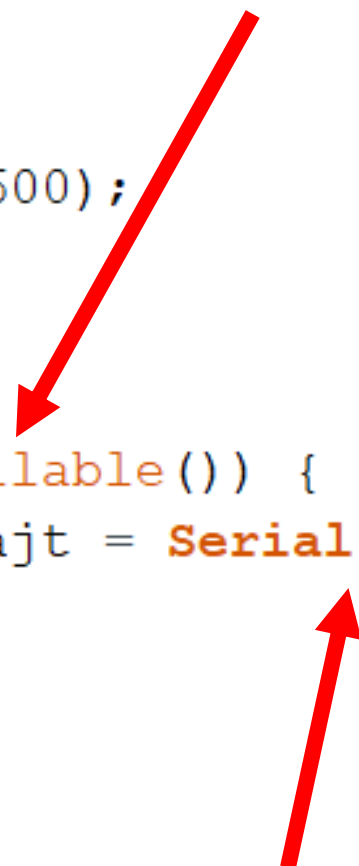


Seriová komunikácia - Arduino

```
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    int c = 100;  
    Serial.println(c);  
    Serial.println("Zaklady IoT");  
}
```

Sú prijaté nejaké dáta?

```
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    if (Serial.available()) {  
        int prijatyBajt = Serial.read();  
    }  
}
```



Prečítaj jeden bajt

Seriová komunikácia - Arduino

- **Serial.flush()** – čaká, kým nie je ukončené odosielanie odoslanie všetkých dát
- **Serial.available()** – vráti počet prijatých, no neprečítaných bajtov uložených **v prijímacom bufferi** (default 64 bajtov)
 - ak je prijímací buffer plný, novoprijaté bajty sa zahadzujú
- **Serial.availableForWrite()** – vráti počet bajtov, ktoré môžu byť zapísané do **odosielacieho buffera** bez toho, aby došlo k zablokovaniu
- **Serial.parseInt()/Serial.parseFloat()** – prečíta zo vstupu celé číslo, resp. číslo s desatinnou čiarkou
 - za ďalším bajtom čaká maximálne zadaný počet milisekúnd nastavený cez **Serial.setTimeout()**

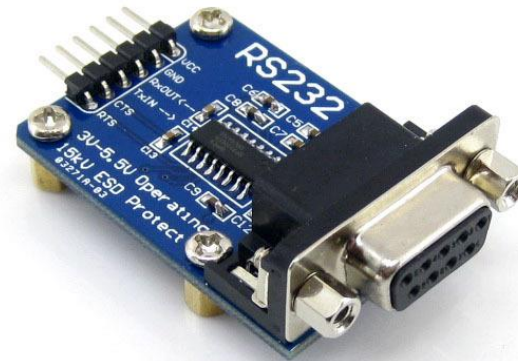
Seriová komunikácia - Java

- Viacero knižníc:
 - JavaComm, RxTx, jSerialComm, jSSC ...
- jSSC:

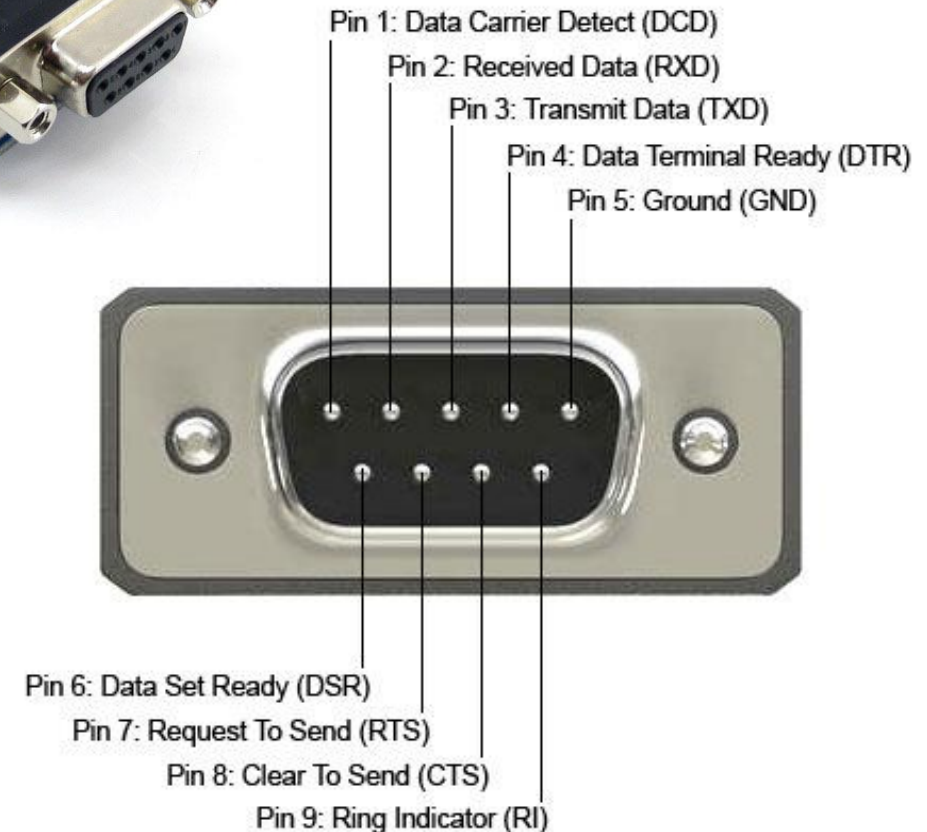
```
<dependency>  
  <groupId>org.scream3r</groupId>  
  <artifactId>jssc</artifactId>  
  <version>2.8.0</version>  
</dependency>
```

RS-232

- RS-232 definuje úrovne **napätia** použité na prenos UART rámcov, konektory, ...
- Počítač (DTE) ↔ Modem (DCE)
- **Logická 0 = 3V až 15V**
- **Logická 1 = -15V až -3V**
- RXD, TXD (data), GND + kontrolné signály
- 15m pri 9600 bps
- V minulosti bol používaný ako **servisný port** TV, SAT-tunerov, atď.

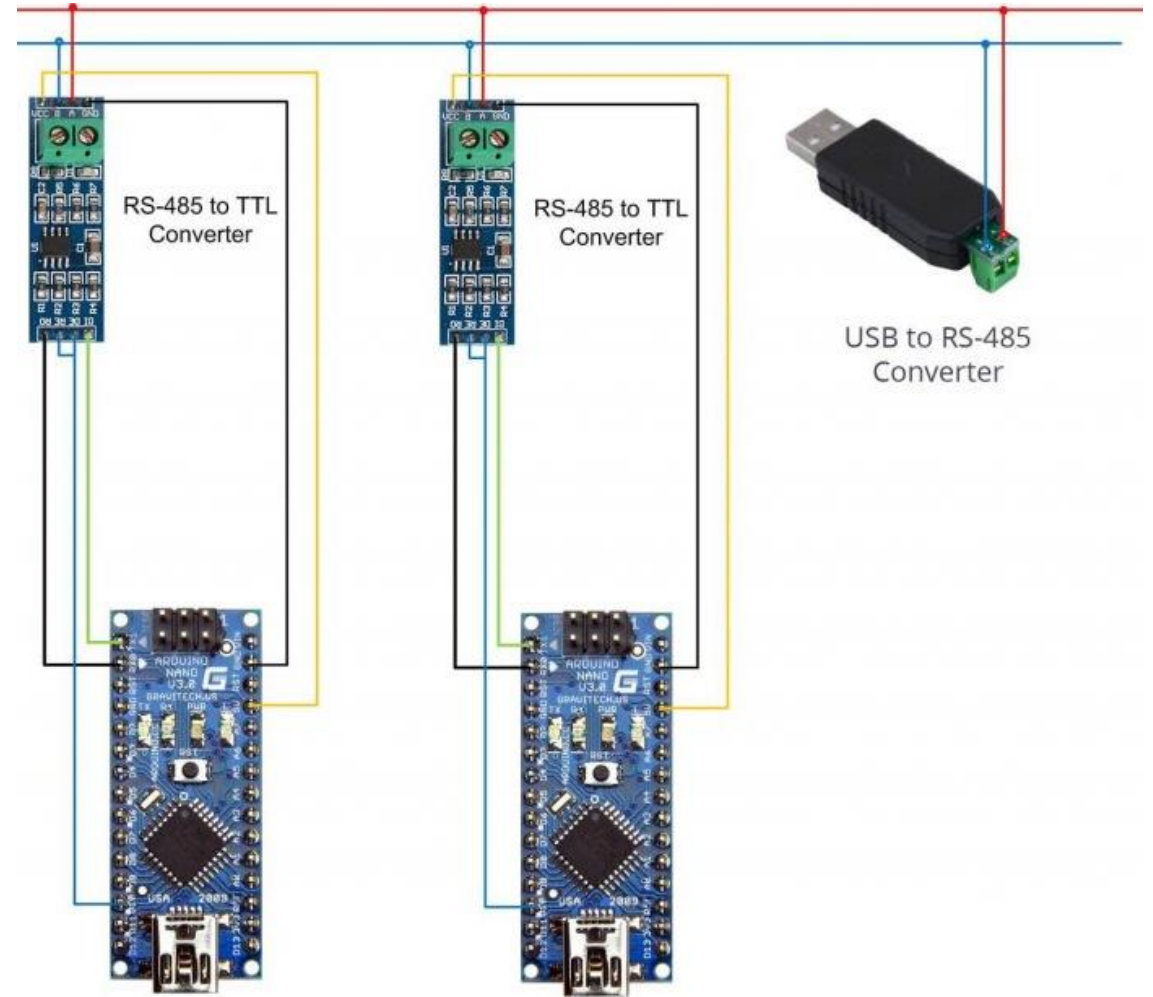


RS232 Pinout



RS-485

- 2 vodiče: A, B
- Bit je určený rozdielom napätia medzi A a B
- **Logická 1:** $A < B$
- **Logická 0:** $A > B$
- Odolnejší voči rušeniu
- 1500m pri 10 Mbps
- Half-duplex
- Možnosť viacero zariadení na RS-485 zbernici, zvyčajne master-slave

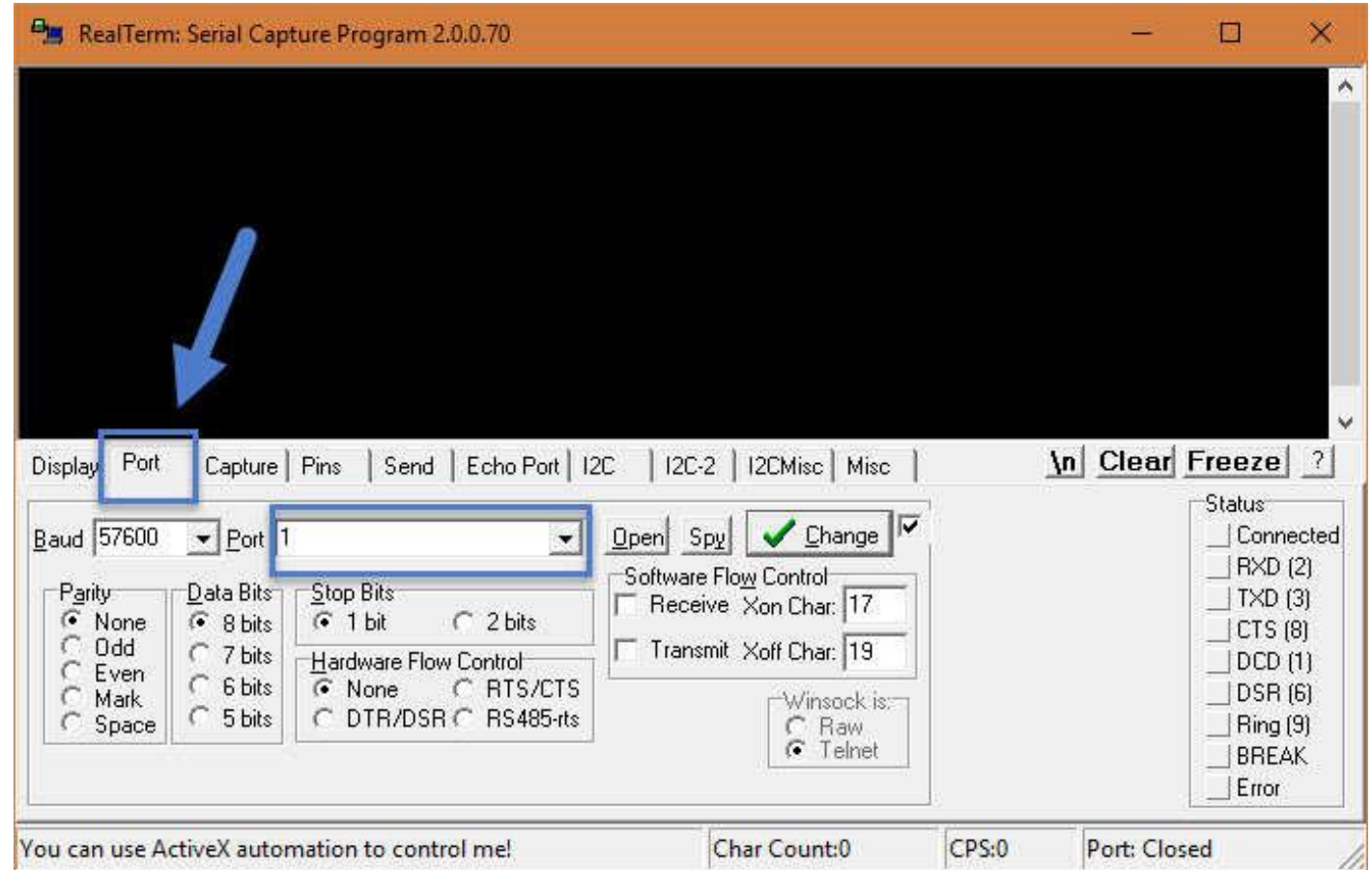


SoftwareSerial/Bit banging

- Bit banging
 - realizácia sériovej komunikácie **softvérovým spôsobom** namiesto špecializovaného hardvéru
 - **softvér je zodpovedný** za nastavovanie pinov, pravidelné samplovanie stavu pinov, časovanie, ...
 - náročný na CPU
- SoftwareSerial
 - softvérová implementácia sériovej komunikácie na Arduine
 - obmedzenia, ktoré piny je možné použiť

Užitočný softvér

- Monitor sériovej komunikácie:
RealTerm
- **Arduino IDE**
 - Serial monitor
 - Serial plotter



Ďakujem za pozornosť