

SCÉNÁŘ

WSN

Lukáš Němec

23. března 2015

# Obsah

<b>1</b>	<b>Arduino všeobecně</b>	<b>2</b>
1.1	Arduino IDE . . . . .	2
1.2	Základy syntaxe . . . . .	3
<b>2</b>	<b>WSN uzel, JeeLib</b>	<b>4</b>
2.1	JeeNode hardware . . . . .	4
2.2	JeeLib . . . . .	4
<b>3</b>	<b>WSN síť</b>	<b>5</b>
3.1	Ukázkové aplikace . . . . .	5
3.2	Síť . . . . .	5

# 1 Arduino všeobecně

Arduino je open-source platforma a zároveň fenomén několika posledních let. Poskytuje jednak samotný hardware a jednak software pro programování mikrokontrolerů ATmega. V současné době už existuje nejenom velké množství oficiálních Arduino desek, ale zároveň se objevuje velké množství jejich klonů. Tyto jsou buď motivovány snahou vytvořit vlastní a levnější variantu svého Arduina, nebo naopak některou z oficiálních desek rozšířit o funkcionalitu navíc. Mezi první kategorii typicky patří čínské klony typu Funduino a další, zatímco druhá kategorie obsahuje například specializované desky, jako jsou námi používané JeeLink a JeeNode.

Jelikož se budeme zabývat ad-hoc sítěmi a senzorovými sítěmi, tak z celé platformy využijeme pouze vývojové prostředí, zatímco hardware použijeme specializovaný pro naše potřeby, především už bude obsahovat rádiový modul. Začneme však od úplných základů práce s Arduinem. Více všeobecných informací je možné najít na oficiálním webu [www.arduino.cc](http://www.arduino.cc).

## 1.1 Arduino IDE

Arduino IDE je velmi jednoduché prostředí pro vyvíjení programů, z praktického hlediska se jedná spíše o upravený textový editor s podporou pro Arduino. Výhodou je podpora pro většinu běžných operačních systémů, tedy je možné vyvíjet jak pod linuxem, tak windows či OS X. Může fungovat buď bez instalace, nicméně tato varianta je vhodná pouze pro samotné programování v případě, že pouze potřebujete programovat a komunikace s Arduinem samotným Vás nezajímá. Pokud Vás však například zajímá komunikace s některou z desek přes sériové rozhraní, vyplatí se prostředí nainstalovat.

Prostředí můžete využít i pokud se rozhodnete programovat ve svém oblíbeném textovém editoru a v Arduino IDE pouze kompilovat, či používat jiné nástroje které poskytuje.

**Instalace** Instalace samotná není složitý proces, spíše naopak. Spousta linuxových distribucí Vám práci usnadní, jelikož Arduino IDE se nachází v repositářích a jediné, co je potřeba pohlídat, je verze, která by měla být ideálně nejnovější dostupná, ideálně verze 1.5 a starší.

Pokud potřebujete instalovat jiným způsobem, pak veškeré potřebné informace naleznete na oficiálních stránkách <http://arduino.cc/en/Main/Software>.

### Nástroje

**Verifikátor** Z nástrojů, které prostředí nabízí je tím nejdůležitějším verifikátor, který slouží k ověření syntaxe napsaného kódu. Tuto možnost bohužel prostředí nenabízí průběžně, tedy je vhodné relativně pravidelně syntaxi kontrolovat.

**Kompilátor** Druhým podstatným nástrojem je kompilátor spojený s nahráváním na desku. Nejdříve je potřeba zvolit konkrétní model Arduina, případně model nejvíce podobný pro veškeré odvozené modely a klony. Dále je třeba zkontrolovat nastavení portu, na který je konkrétní deska připojena. V případě, že je připojena pouze jedna většinou prostředí port vybere správně, nicméně kontrola je vhodná. Zároveň je možné mít připojeno více desek a poté je potřeba zvolit jednu konkrétní.

V případě problémů s nalezením desky je vhodné restartovat celé prostředí, případně desku připojovat před spuštěním prostředí.

**Serial monitor** Serial monitor je posledním důležitým nástrojem, který je vhodné znát. Jeho funkcionalitu sice lze nahradit pomocí jakéhokoliv programu, který umožňuje komunikovat přes sériový port (například PySerial), na druhou stranu je Serial Monitor tou nejjednodušší variantou.

Při spuštění je třeba nastavit frekvenci komunikace (na stejnou hodnotu, jako ve zdrojovém kódu programu) a následně je možné číst informace zaslané z desky, nebo případně desce posílat instrukce.

Pokud ze zobrazují znaky, které však nedávají smysl, bude pravděpodobně problém s různě nastavenou frekvencí desky a Serial Monitoru.

## 1.2 Základy syntaxe

Arduino je možné brát jako speciální případ jazyka C. Zůstává téměř vše a mění se pouze podoba funkce `main`, která je rozdělená na dvě části, `setup` a `loop`. Tedy je možné psát `for` cykly jako v jazyce C, stejně tak podmínky, funkce a téměř cokoliv dalšího, co budete potřebovat. Navíc Arduino přidává některé vlastní funkce a velké množství knihoven, které především slouží k příjemnější manipulaci s veškerým možným příslušenstvím. Nicméně můžeme nalézt i různé jiné knihovny, které především usnadňují psaní kódu.

**setup** Funkce sloužící k úvodnímu nastavení proměnných, počátečnímu spuštění a všem dalším úvodním nastavením. Při samotném spuštění se tato funkce zavolá jako první a právě jednou. Ideální je například ke spuštění komunikace po sériovém portu, nastavení pinů a inicializaci knihoven.

**loop** Funkce která provádí samotný běh programu, jak napovídá její název, tuto funkci můžeme vnímat jako nekonečný cyklus. Zavolá se až po funkci `setup`, tedy již můžeme počítat s inicializovaným prostředím a bude se volat neustále až do restartu desky nebo ztráty napájení. Zdrojový kód v této funkci je vhodné psát s ohledem na neustálé opakování a zároveň v případě, že potřebujeme něco pravidelně kontrolovat (stisknutí tlačítka) je vhodné mít buď celý kód dostatečně rychlý tak, aby kontrola probíhala v krátkých intervalech, nebo kontrolu provést několikrát za jeden průběh cyklu.<sup>1</sup>

---

<sup>1</sup>některé desky umožňují využít speciální piny k přerušení funkce `loop` a okamžitou reakci na událost. K přerušení je možné připojit k speciální vlastní funkci, která vykoná jeho obsluhu

**Serial** Jedna z nejdůležitějších knihoven slouží k ovládání sériového portu na straně desky. Na začátku je třeba ve funkci `setup` inicializovat společně s nastavením frekvence komunikace. Následně je možné jej začít používat, klidně již ve funkci `setup`. Všechny funkce této knihovny je třeba volat vždy jako `Serial.read()` nikoliv pouze `read()`.

**print** Slouží k výpisu hodnot a ladících informací přes sériový vstup. Je možné tisknout hodnotu samotnou, případně ji i formátovat (vhodné například pro hexadecimální hodnoty). Taktéž je možné použít funkci `println`, která navíc pošle i ukončení řádku. Při používání je třeba dát pozor, jelikož funkce je asynchronní, tedy vrátí návratovou hodnotu dříve, než začne odesílat znaky.

Pro obsluhu výstupu na sériovém portu je možné využít i funkce `write` (zápis binárních dat) nebo `flush` (počká na dokončení zápisu).

**read** Slouží ke čtení ze sériového portu, přečtenou hodnotu vrací jako návratovou hodnotu. Vhodné je její spojení s funkcí `available`, která vrací počet bytů dostupných k přečtení. Ideální použití je buď k ovládání desky pomocí příkazů z počítače, nebo k počátečnímu nastavení, které není možné nastavit v kódu napevno.

## 2 WSN uzel, JeeLib

### 2.1 JeeNode hardware

Všeobecná specifikace

Rádio

Piny

### 2.2 JeeLib

Přidání do IDE

Formát hlavičky

Poslání zprávy

Přijetí zprávy

Síť ze dvou uzlů

## **3 WSN síť**

### **3.1 Ukázkové aplikace**

Alive

Sniffer

### **3.2 Síť**

Všeobecné parametry

Fixní routování