Edushield – metodická příručka

# Úvod

EduShield byl vytvořen pro výuku základů práce s elektronikou a Internetem věcí. Jeho primárním cílem je přinést jednoduchý a cenově dostupný způsob, jakým mohou zájemci z věkové skupiny 12+ získat základní i rozšířenou sadu kompetencí v oblasti mikroelektroniky, programování elektronických zařízení a propojování věcí z reálného světa s digitální technologií.

EduShield je otevřená technologie, navržená tak, že spolupracuje s nejrozšířenější elektronickou platformou dneška, s vývojovým kitem Arduino. Výhodou tohoto kitu je nízká cena, příznivá křivka učení a obrovská dostupnost rozšiřujících zařízení, díky nimž je možné použít Arduino jako platformu pro sběr dat, pro řízení strojů, pro vývoj elektronických zařízení i pro výuku nejrůznějších technik a technologií.

EduShield byl navržen na základě zkušeností s kurzy základů elektroniky, kterými prošlo přes 400 lidí. Zohledňuje potřeby a cíle výuky jak pro základní seznámení, tak i pro pokročilejší techniky. Rozvíjí kompetence v oblastech práce s elektronickými zařízeními, od programování po jejich reálný návrh. Učí frekventanty, jak spojit reálný svět s digitálním, ukazuje způsoby měření fyzikálních veličin a techniky pro interakci s lidmi. Předpokladem pro úspěšné absolvování kurzu jsou znalosti základů elektroniky. Tyto znalosti jsou v průběhu kurzu upevněny a jsou posíleny o kompetence tzv. extrémního programování. Účastníci zároveň získávají důležité kompetence v oblasti inteligentních systémů řízení, Internetu věcí, automatizace a regulace, a v oblasti návrhu a testování elektronických obvodů.

# Prerekvizity

* Znalost základů programování v jazyce, který vychází z jazyka C (C, Java, C++, C#, JavaScript, PHP a další). Požadovaná úroveň: znalost konceptu proměnných, konstant, funkcí a základních konstrukcí programovacího jazyka (smyčky, podmínky)
* Fakultativní: znalost fyzikálních principů elektřiny (napětí, proud, funkce el. obvodů)

# Příprava pro studenty

* Instalace IDE. Arduino IDE je dostupné pro Linux, Mac a Windows <http://www.arduino.cc/en/Main/Software>.
* Zdrojové kódy ukázkových aplikací <https://github.com/bechynsky/ArduinoWorkshop>. Stačí vybrat *Download ZIP*. Součástí balíku jsou i knihovny pro použité rozšíření.
* Schéma shieldu a datasheet pro použité obvody

# Příprava pro lektora

**Vše si nejdříve vyzkoušejte, než začnete školit, nebo se sami tohoto školení zúčastněte.**

## Pro studenty

Pro každého studenta je potřeba mít připraveno:

1. Než začne seminář
   1. Poslat studentům odkaz, co si mají stáhnout a nainstalovat
      1. <https://www.arduino.cc/en/Main/Software>
      2. https://bitbucket.org/maly/edushield/src
2. Hardware
   1. Arduino Uno nebo klon. **U klonů pozor na nutnost instalace ovladačů!**
   2. EduShield
   3. USB flash disk ke každému hardware.
      1. Arduino IDE pro všechny platformy, kdyby někdo zapomněl.
      2. Zdrojové kódy z https://bitbucket.org/maly/edushield/src
      3. Všechny potřebné knihovny.
      4. Ovladače
   4. Vytisknout pro každého účastníka:
      1. Arduino Cheat Sheet <https://github.com/liffiton/Arduino-Cheat-Sheet>
      2. EduShield Cheat Sheet.docx

## Instalace ovladačů

Podle operačního systému a použité desky se může instalace ovladačů lišit. Systém musí desku Arduino rozpoznat jako sériový port. Pro převod USB na sériový port se používají dva typy čipů.

* FTDI (<http://www.ftdichip.com>) – nejčastěji u dražších Arduino. Některé klony Arduino mohou obsahovat čip, který je kopií čipu od společnosti FTDI. S těmito čipy mohou mít oficiální ovladače problémy.
* CH340 – levná alternativa čipů FTDI. Většinou najdete ve velmi levných klonech Arduino. Viz https://iotta.cz/ovladace-pro-ch340g/

### Mac OS

Pokud není zařízení rozpoznáno, je třeba nainstalovat ovladač z https://iotta.cz/ovladace-pro-ch340g/

Další alternativy: <http://www.wch.cn/download/CH341SER_MAC_ZIP.html>

<http://0xcf.com/2015/03/13/chinese-arduinos-with-ch340-ch341-serial-usb-chip-on-os-x-yosemite/>

### Linux

Arduino by mělo být rozpoznáno automaticky s čipem FTDI i CH340. Uživatel musí být ve skupině *dialup*, aby měl přístup k sériovému portu a nemusel Arduino IDE spouštět přes *sudo*.

#### Testované distribuce

* Ubuntu 14.04 LTS
* Ubuntu 15.04
* Ubuntu 16.04

### Windows 10

Instalace ovladačů proběhla automaticky. Vyžaduje připojení k internetu.

### Windows 8.1, 8, 7

Pokud se ovladače nenainstalují automaticky, tak je třeba ovladač nainstalovat ručně. Umístění ovladačů:

* Ovladače pro FTDI čip jsou součástí Arduino IDE.
* Ovladač pro CH340 čip: https://iotta.cz/ovladace-pro-ch340g/

### Instalace nepodepsaných ovladačů

Při instalaci ovladačů na 64-bit verze Windows může výjimečně dojít k problému s digitálním podpisem ovladače. Nové verze ovladačů pro čip FTDI jsou podepsané. Může být problém s ovladačem pro CH340, ale neměl by být. Pokud systém odmítne instalaci ovladače kvůli digitálnímu podpisu, tak postupujte podle tohoto návodu <http://robodoupe.cz/wp-content/uploads/2013/04/Arduino-a-PICAXE-na-Windows-8-64.pdf>

## Arduino.cc vs. Arduino.org

V nedávné době došlo k rozkolu mezi zakladateli projektu Arduino (<http://hackaday.com/tag/arduino-vs-arduino/>). V tuto chvíli existují dvě Arduino IDE. Jedno ze stránek Arduino.cc a druhé za stránek Arduino.org. Každé má trochu jiné vlastnosti a v základu podporuje jinou sadu desek Arduino. V příkladech budeme používat Arduino IDE z Arduino.cc.

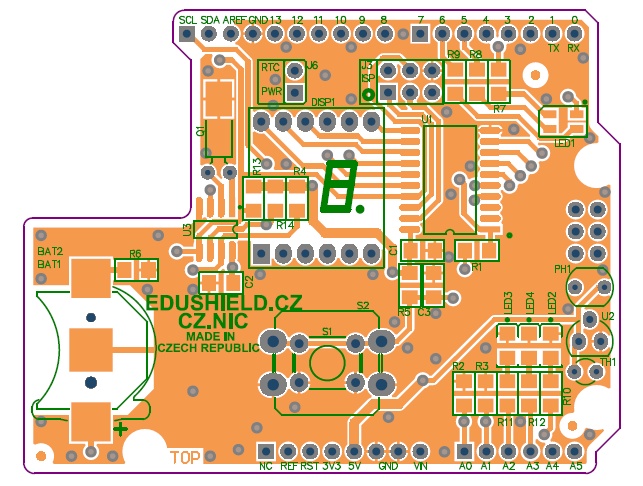
# S čím budeme učit

## Hardware

### Arduino UNO

Arduino je jednoduchá vývojová deska s mikrořadičem AVR společnosti ATMEL. Na této desce najdete vše potřebné pro vytváření jednoduchého hardware pro domácí automatizaci, robotiku, interaktivní instalace a mnoho dalšího.

### EduShield



## Software

### Arduino IDE

Arduino IDE je vývojové prostředí, pomocí kterého budete psát aplikace pro vývojové desky Arduino a zároveň umožňuje aplikaci na vývojovou desku Arduino nahrát.

# Rozvrh

## 4 hodiny

?

## 8 hodin

?

# Lekce

## Arduino IDE a práce s HW Arduino

### Probraná teorie

* Instalace IDE
* Instalace ovladače
* Výběr desky v nastavení
* Nastavení sériového portu
* Nahrání aplikace do desky
* Připojení desky k PC, spojení se shieldem

### Praktické ukázky

* Step-by-step demonstrace probíraných témat na učitelském PC a projektoru

## Hello World!

*Blink.ino*

Blikání LED.

Doporučujeme nejprve ukázat kód, nechat účastníky, aby jej našli v příkladech, zkompilovali, nahráli a viděli, co se děje. Teprve pak vysvětlit strukturu kódu a účel.

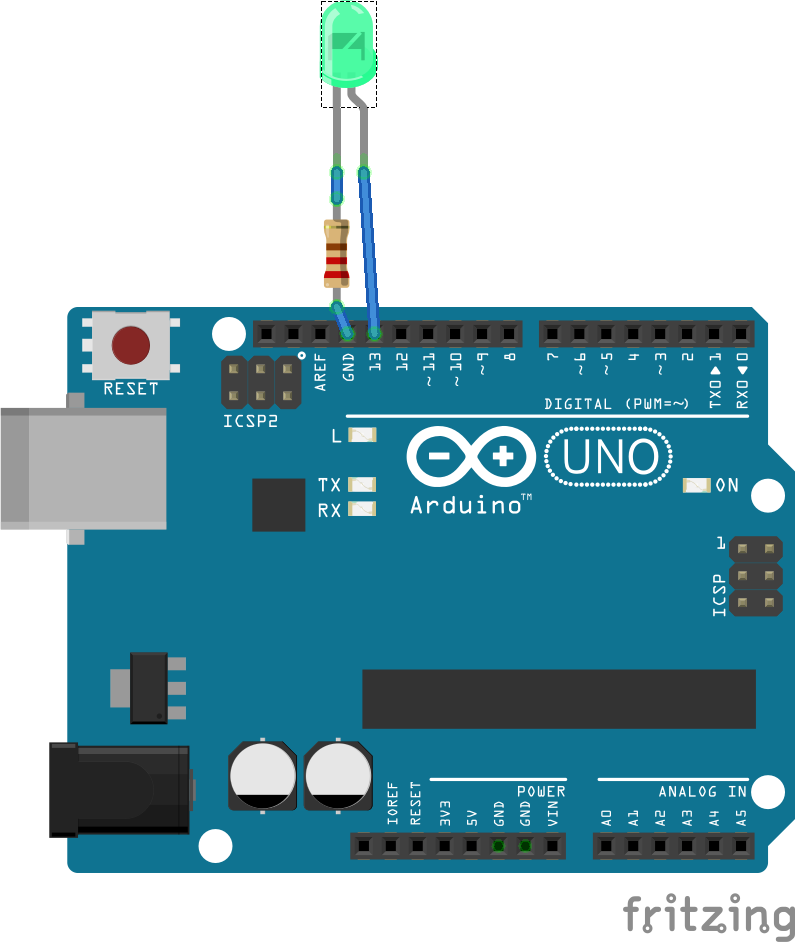
### Teorie

* Logické úrovně
* Fungování LED
* Předřazený rezistor
* Práce s IDE: otevřít soubor, přeložit, nahrát

### Funkce

* pinMode
* digitalWrite
* delay

### Zapojení



### Cvičení

* Změna rychlosti blikání
* Použití dalších LED
* Morseovka <https://cs.wikipedia.org/wiki/Morseova_abeceda>

### Pro lektora

1. Vysvětlit studentům, že delay blokuje vykonávání kódu.

## RGB LED

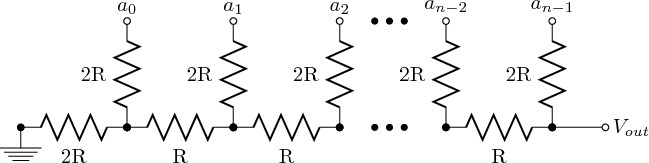
Ukázat, jak se pracuje s RGB LED – PWM

### Teorie

* RGB – míchání barevných světel. Každá barva se skládá ze tří základních světel
* Pro pokročilejší: zmínit rozdíl mezi „mícháním světel“ a „mícháním barevných pigmentů“
* PWM <https://cs.wikipedia.org/wiki/Pulzn%C4%9B_%C5%A1%C3%AD%C5%99kov%C3%A1_modulace>



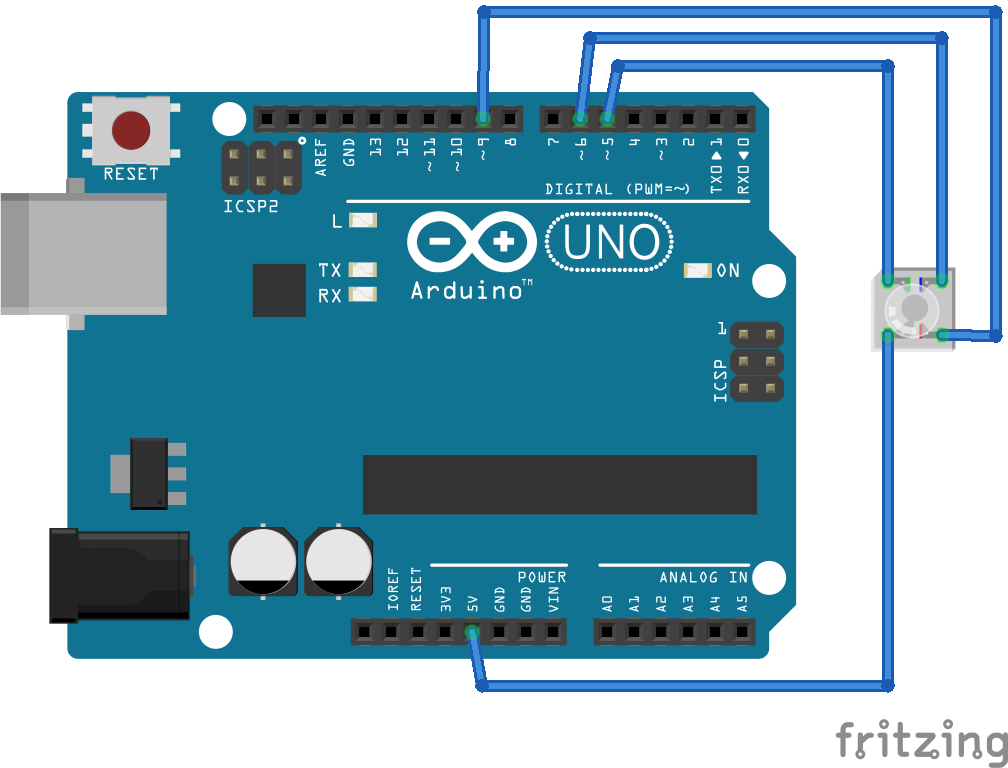
* D/A převodník pomocí rezistorů <https://en.wikipedia.org/wiki/Resistor_ladder>



### Funkce

* analogWrite

### Zapojení



### Praktické cvičení

* Ukázka plynulých změn barvy světla
* Změnit přírůstky/úbytky pro PWM. (Student by si měl všimnout, že od určité hranice svítí LED plně. Vysvětlit důvod: minimální napětí nutné pro rozsvícení)

## Sériová komunikace

Využití sériové komunikace pro ladění programu.

### Teorie

* RS232 <https://cs.wikipedia.org/wiki/RS-232>
* Arduino IDE Serial Monitor
* Převodník USB-Serial
  + 5 V vs 3,3 V
* AT příkazy
  + Bluetooth modul
* HW serial port na Arduino
  + Arduno Mega
* SW serial port na Arduino

### Funkce

* Serial <https://www.arduino.cc/en/Reference/Serial>

### Cvičení

* Ovládejte LED pomocí příkazů poslaných přes sériovou linku.
* Vyzkoušejte, co se stane, když nejsou na obou stranách nastaveny stejné rychlosti.

### Pro lektora

1. Místo readString() použijte jen read()

## Semafor

*Semafor-\*.ino*

Krok 1: Využijte vědomosti pro sestrojení jednoduchého semaforu. Posloupnost barev: Č – Č+O – Z – O – Č

Krok 2: Další krok na stisknutí tlačítka

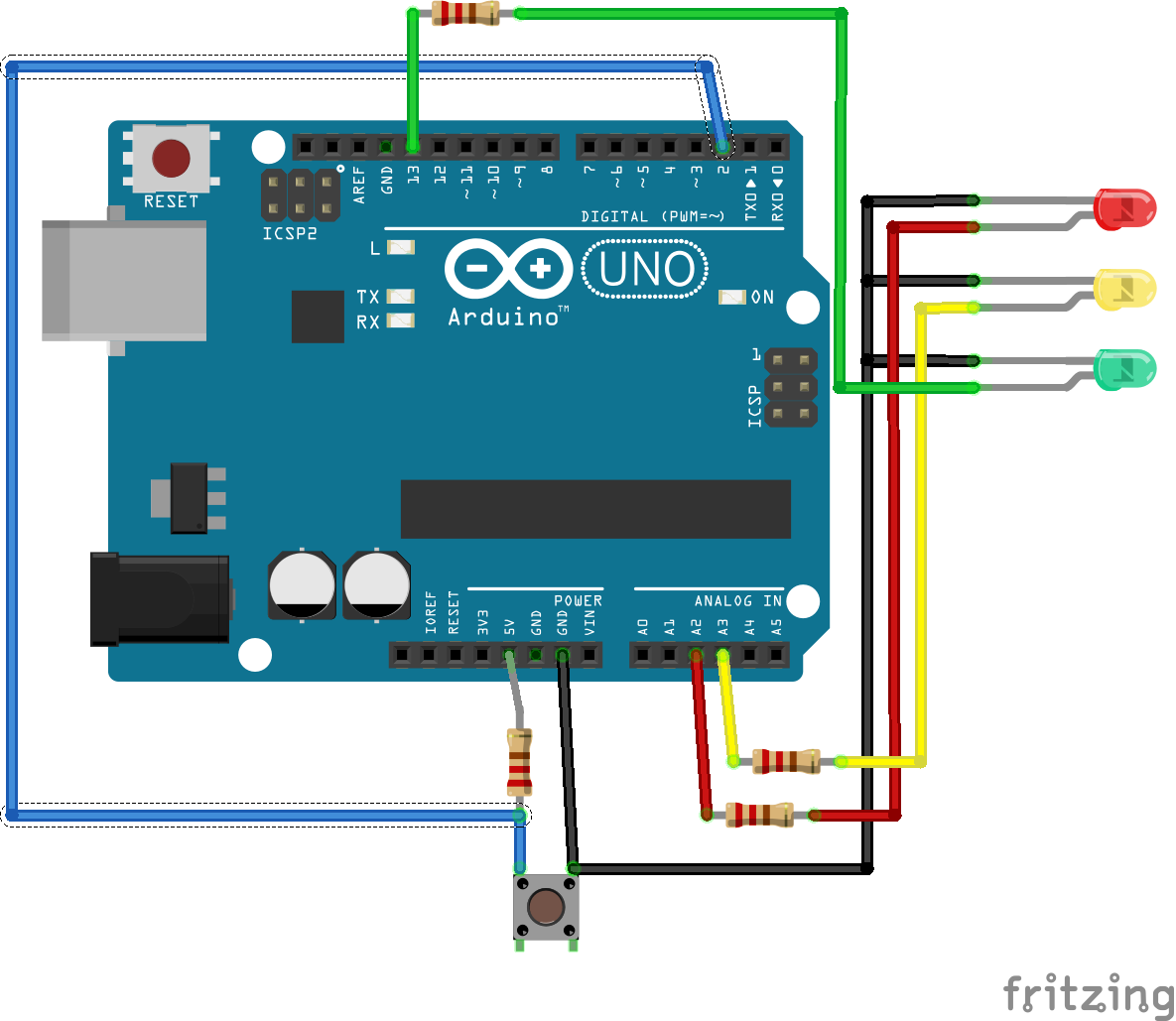
### Teorie

* Stavový automat vs vodopád
* Ukázka nejednoznačnosti tlačítka (zákmity atd.)
* Prázdná smyčka (čekání na něco)
* Tlačítka
* Pull-up rezistor <https://cs.wikipedia.org/wiki/Pull_up_rezistor>

### Funkce

* digitalRead
* INPUT\_PULLUP

### Zapojení



### Cvičení

1. Vyzkoušet chování bez pull-up rezistoru.
2. Upravte příklad tak, aby svítila červená, a na stisknutí tlačítka proběhl celý semaforový cyklus zpátky k červené (spouštěný automat)

## Hello world

*hello.ino*

Zobrazení nápisu "AHOJ" na 7seg displeji

### Teorie

* Nalezení správného postupu v knihovně
* Implementace vlastního postupu
* Ovládání hardware napřímo, pokud knihovna nenabízí požadované

### Cvičení

* Přidat tabulku nejčastějších znaků
* Otestovat změnu jasu

### Pro lektora

* Je třeba prozkoumat knihovnu a nalézt místo, kde se posílají data do displeje. Všimněte si položky, která by mohla obsahovat jas. Experimentálně zjistěte rozložení segmentů (rozsvěcení jednoho po druhém).

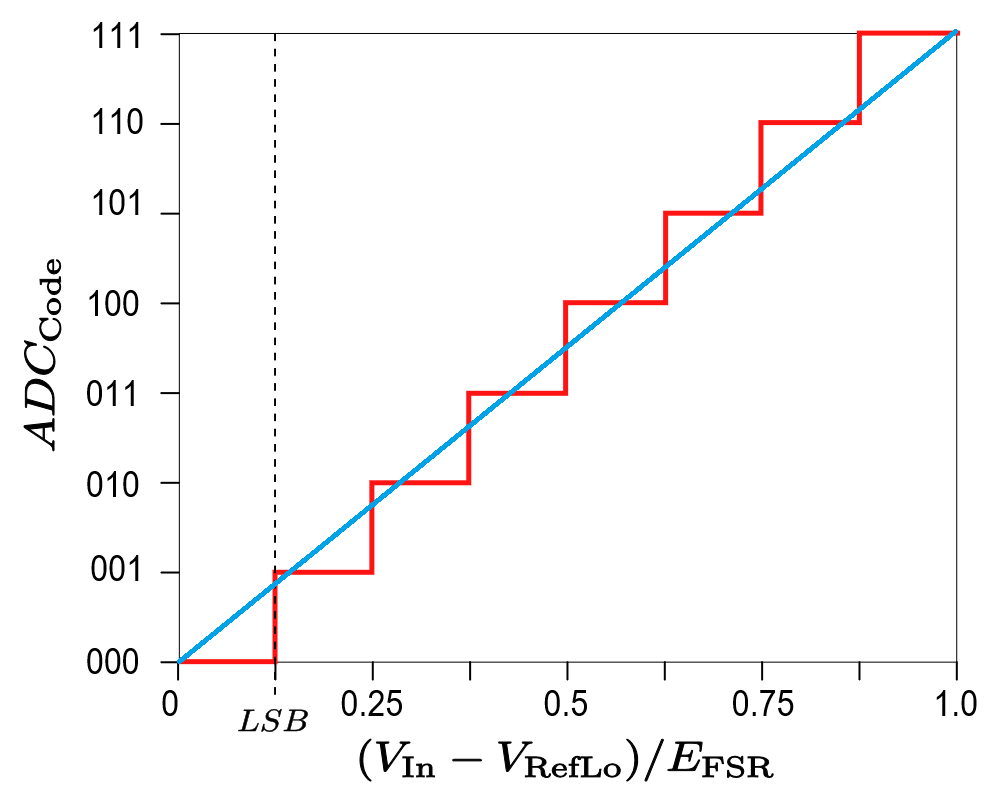
## Měření Světla

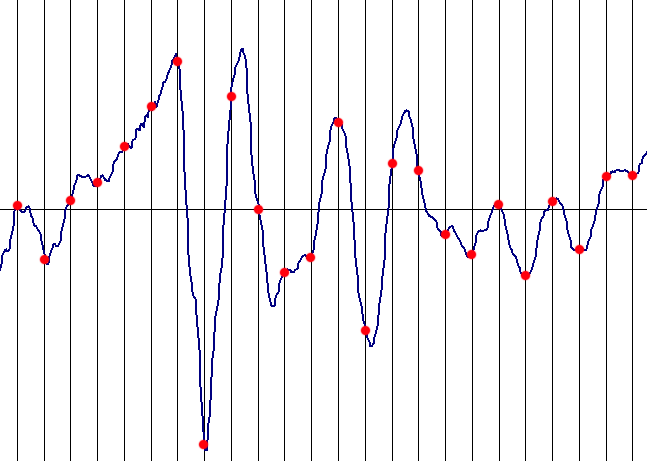
*LightScale.ino*

Intenzita světla se převede na LED, které se budou chovat jako škála. Hodnota načtená z analogového senzoru se vypíše na sériovou konzoli.

### Teorie

* Analogový senzor
* ADC <https://cs.wikipedia.org/wiki/A/D_p%C5%99evodn%C3%ADk>

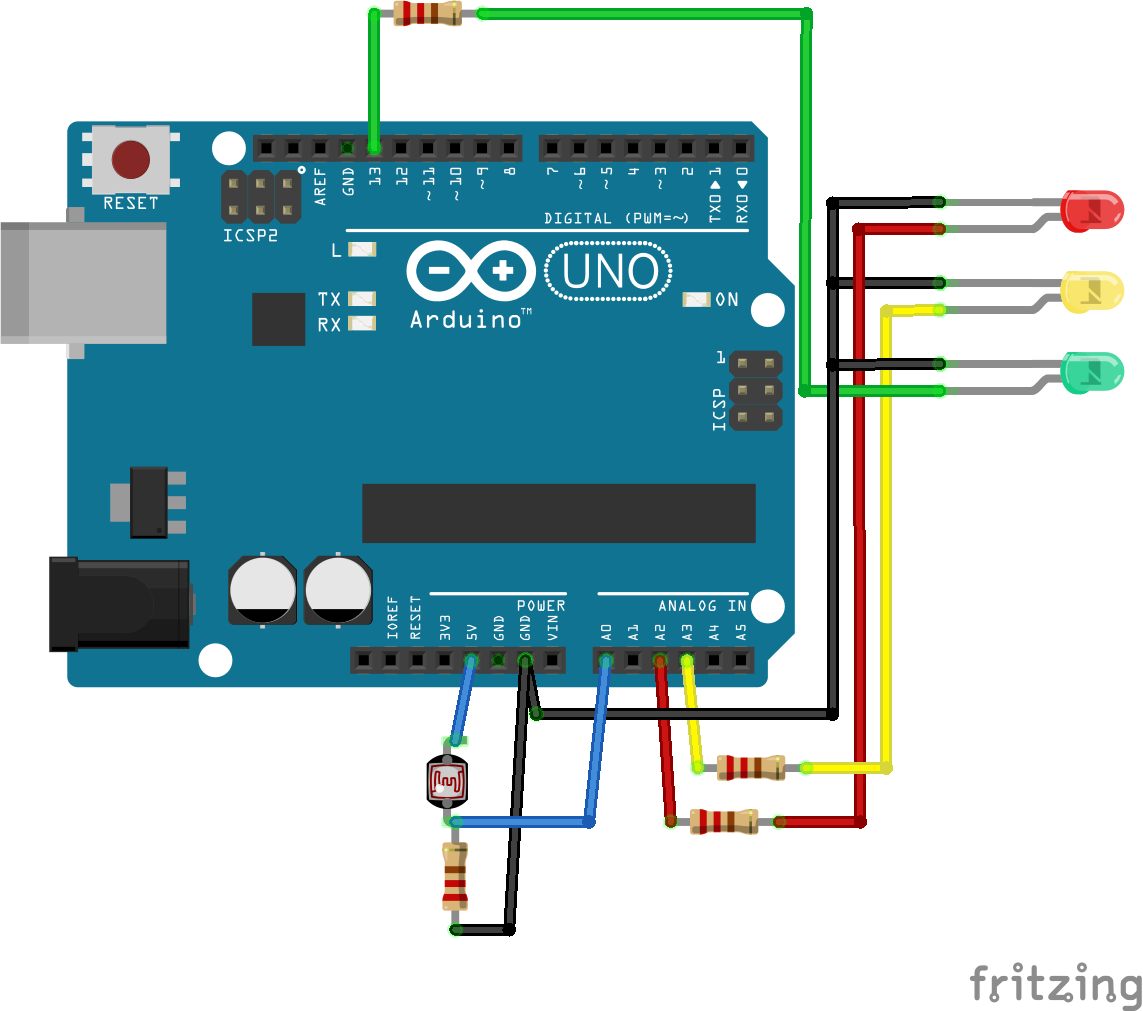


* + 10bit – 0-1024 → 0 V-5 (3,3) V
* Vzorkovací frekvence <https://cs.wikipedia.org/wiki/Vzorkov%C3%A1n%C3%AD> 
* Dělič napětí <https://cs.wikipedia.org/wiki/D%C4%9Bli%C4%8D_nap%C4%9Bt%C3%AD>

### Funkce

* analogRead
* map

### Zapojení



### Cvičení

1. Výpočet napětí na senzoru.
2. Změna citlivosti posunem rozsahu ve funkci *map*.

### Pro lektora

1. Referenční napětí je 5 V, převodník je 10-bit, tedy 0-1023.
2. Díky napěťovému děliči určité hranice nemusejí mít vliv. Citlivost obecně závisí na použitém rezistoru připojeného na zem.
3. Je třeba najít mez citlivosti.

## Teploměr

*MeasureTemperature.ino*

1. Nahradit teploměr za fotorezistor z předchozího příkladu a vypisovat hodnoty na sériové lince

2. Teplota se zobrazí na displeji.

### Teorie

* Instalace knihovny v IDE a ručně.
* Elektrotechnické schéma a značky.

### Cvičení

1. Při dosažení zvolené teploty rozsvítit LED.
2. Teplotu zobrazit v °F <https://cs.wikipedia.org/wiki/Stupe%C5%88_Fahrenheita>.
3. Vytvořit počítadlo průchodů lidí nebo předmětů pomocí světelného senzoru.

### Pro lektora

1. Není třeba přepočítávat na reálnou teplotu. Stačí surová hodnota ze senzoru, ušetří se výpočetní výkon, paměť, …
2. Účastníci si musí najít, jak zobrazit F. Je třeba je nasměrovat na zdrojové soubory knihovny TM1636.
3. Nastavením citlivosti senzoru na světlo vytvořit hranici, kdy zastínění bude odpovídat průchodu člověka nebo předmětu.

## Hodiny

*Clock.ino*

Zobrazení času na displeji s blikající dvojtečkou.

### Teorie

* Instalace knihovny pomocí správce knihoven.

### Cvičení

1. Zobrazit minuty a vteřiny.

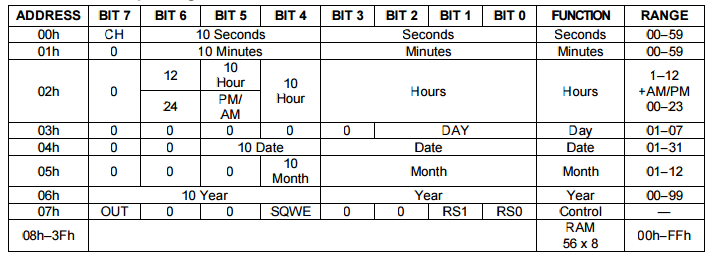
### Hodiny bez knihovny

*RealTimeClock.ino*

RTC bez hotové knihovny na úrovni sběrnice I2C. Výpis hodnot do počítače přes UART.

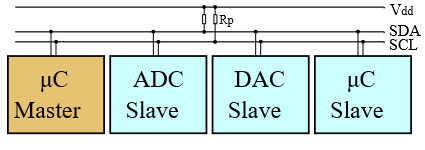
### Teorie

* Práce s katalogovým listem.
* BCD kódování <https://cs.wikipedia.org/wiki/BCD>.



<http://datasheets.maximintegrated.com/en/ds/DS1307.pdf>

* Specifikace <http://www.nxp.com/documents/user_manual/UM10204.pdf>



### Funkce

* Knihovna Wire

### Cvičení

1. Zobrazit kompletní datum a čas na sériovou konzoli.

## Processing

*TemperatureChart.pde, TemperatureChartDataSource.ino* (složka *Processing*)

Zobrazení naměřených hodnot ze senzoru teploty v grafu.

### Teorie

* Základy prostředí Processing <https://processing.org/tutorials/overview/>
* Práce se sériovým portem v prostředí Processing <https://processing.org/reference/libraries/serial/index.html>

### Cvičení

1. Přidat graf pro senzor světla.

### Pro lektora

* Je třeba správně přepočítávat hodnotu ze senzoru, aby odpovídala měřítku. Data lze předávat najednou a oddělit je např. středníkem (funkce *split* pro zpracování).

# Náměty na samostatnou práci

## Sledování teploty v lednici

1. Aplikace uchovává maximální a minimální teplotu.
2. Stiskem tlačítek volíme zobrazení minimální, maximální nebo aktuální teploty na displeji.
3. Typ zobrazené teploty je indikován LED.
4. Pokud je lednice otevřená déle než 10 sekund, tak začne bzučák pískat.

## Generátor náhodných čísel

* <https://cs.wikipedia.org/wiki/Gener%C3%A1tor_n%C3%A1hodn%C3%BDch_%C4%8D%C3%ADsel>
* <http://www.root.cz/clanky/hardwarovy-generator-nahodnych-cisel-aneb-nahoda-z-atomu/>

1. Čteme nejnižší bit A/D převodníku, ke kterému je připojený zdroj šumu (dioda, nebo i ten fotorezistor / termistor).
2. Sebrat vzorek 1000 / 10000 / 100 000 náhodných čísel, podívat se, jak náhodná jsou (rovnoměrné rozdělení atd.)
3. Cvičení z pilnosti: zakreslit např. průběh šumu do grafu; podívat se, jestli šum roste s teplem (u diody ano, stačilo cvaknout zapalovačem) ;)