

BAKALÁŘSKÁ PRÁCE

Praktické vyučování podporované počítačem

Practical teaching supported by computers

STUDIJNÍ PROGRAM

Specializace v pedagogice

STUDIJNÍ OBOR

Učitelství praktického vyučování a odborného výcviku

VEDOUCÍ PRÁCE

Prof. RNDr. Emanuel Svoboda, CSc.

ŽÁK

PAVEL

2019

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: Žák Jméno: Pavel Osobní číslo: 470484
Fakulta/ústav: Masarykův ústav vyšších studií (MÚVS)
Zadávací katedra/ústav: Oddělení pedagogických a psychologických studií
Studijní program: Specializace v pedagogice (B7507)
Studijní obor: Učitelství praktického vyučování a odborného výcviku (7507R056)

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Praktické vyučování podporované počítačem

Název bakalářské práce anglicky:

Practical teaching supported by computers

Pokyny pro vypracování:

Bakalářská práce bude obsahovat teoretickou a praktickou část.

V teoretické části se student zaměří na názorně demonstrační metody a zmapuje problematiku laboratoří se vzdáleným přístupem.

V praktické části student navrhne a zrealizuje vzdálenou laboratoř pro vhodný experiment. Vzdálená laboratoř bude založena na prototypovací platformě Arduino. Dále budou analyzovány výhody a nevýhody vzdálených laboratoří oproti reálným experimentům vyplývající ze zkušenosti s realizací. Součástí práce bude také porovnání finanční a časové náročnosti výroby vzdáleného experimentu oproti reálnému experimentu provedenému ve vyučovací hodině.

Seznam doporučené literatury:

VANĚČEK, David a kol. Didaktika technických odborných předmětů. Praha: ČVUT v Praze, 2016.

MAŇÁK, Josef a ŠVEC, Vlastimil. Výukové metody. Brno: Paido, 2003. 219 s. SKALKOVÁ, Jarmila. Obecná didaktika: vyučovací proces, učivo a jeho výběr, metody, organizační formy vyučování. Praha: Grada, 2007. Pedagogika (Grada). LUSTIG, František.

Distanční fyzikální laboratoře. In MECHLOVÁ, E. (ed.). ICTE 2003 - sborník příspěvků. Ostrava: PTF OU, 2003, s. 27-34.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

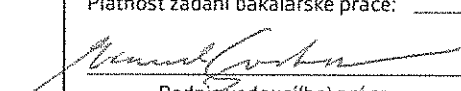
Prof. RNDr. Emanuel Svoboda, CSc.; Oddělení pedagogických a psychologických studií, MÚVS, ČVUT v Praze

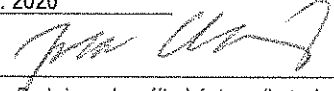
Jméno a pracoviště konzultanta(ky) bakalářské práce:

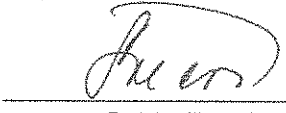
Doc. RNDr. František Lustig, Matematicko-fyzikální fakulta UK, Praha.

Datum zadání bakalářské práce: 13. 12. 2018 Termín odevzdání bakalářské práce: 2. 5. 2019

Platnost zadání bakalářské práce: 30. 9. 2020


Podpis vedoucí(ho) práce

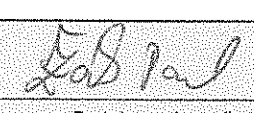

Podpis vedoucí(ho) ústavu/katedry


Podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

14. 2. 2019

Datum převzetí zadání



Podpis studenta(ky)

ŽÁK, Pavel. *Praktické vyučování podporované počítačem*. Praha: ČVUT 2019. Bakalářská práce. České vysoké učení technické v Praze, Masarykův ústav vyšších studií.



**MASARYKŮV ÚSTAV
VYŠŠÍCH STUDIÍ
ČVUT V PRAZE**

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně. Dále prohlašuji, že jsem všechny použité zdroje správně a úplně citoval a uvádím je v příloženém seznamu použité literatury.

Nemám závažný důvod proti zpřístupnění této závěrečné práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

V Praze dne: 02. 05. 2019

Podpis:

Poděkování

Mé upřímné poděkování je věnováno vedoucímu bakalářské práce, panu profesoru RNDr. Emanuelu Svobodovi, CSc., za jeho přátelský přístup, obětavost, trpělivost a nevyčerpatelnou víru v existenci lepších stránek nás, studentů.

Velice rád bych touto cestou poděkoval také doc. RNDr. Františku Lustigovi, CSc. za jeho vstřícnost, odborné rady, technickou podporu, a především za zapůjčení softwaru iSES Remduino Lab, potřebného pro tvorbu mé bakalářské práce.

Srdečné poděkování patří mé rodině. Především Hanině, „Malému betonáři“ a Kubovi za podporu i kritiku mé práce a za jejich dobré rady. Na poděkování mé Janě by mi nestačila slova ani stránky, přesto děkuji.

Abstrakt

Bakalářská práce obsahuje teoretickou a praktickou část. V teoretické části jsou zmapovány výukové metody včetně názorně demonstračních. Je zde popsán vznik a vývoj laboratoří se vzdáleným přístupem a klasifikace experimentů. V praktické části je podrobně popsán návrh a realizace vzdálené laboratoře s pomocí systému iSES Remduino Lab a prototypovací platformy Arduino UNO. Při tvorbě vzdáleného experimentu byla zachycena časová i finanční náročnost sloužící k porovnání s reálným experimentem ve vyučovací hodině. Byly také analyzovány výhody i nevýhody vzdáleného experimentu.

Klíčová slova

Výukové metody, metoda demonstrační, vzdálená laboratoř, vzdálený experiment, reálný experiment, vzdálený přístup, systém iSES, iSES Remduino Lab, Arduino UNO, internet

Abstract

This bachelor thesis is divided into two parts – theoretical and the practical part. Theoretical part is focused on teaching methods including especially demonstration methods. Formation of the remote-labs is also described there. In the practical part there is described in the detail design and realization of the remote-lab using iSES Remduino Lab and prototyping platform Arduino UNO. Financial and time difficulty was described in the creation of the experiment. Remote experiment should be compared with the real experiment shown during the teaching lesson. Advantages and disadvantages of the remote-lab were also analysed.

Keywords

Teaching methods, demonstration method, remote Lab, remote experiment, realistic experiment, remote access, iSES system, iSES Remduino Lab, Arduino UNO, internet

Obsah

ÚVOD.....	5
TEORETICKÁ ČÁST.....	6
1 METODY VÝUKY.....	7
2 VZDÁLENÉ LABORATOŘE.....	15
2.1 Historie internetu	15
2.2 Školský pokus – experiment ve výuce přírodních věd.....	17
2.3 Klasifikace experimentu s podporou počítače.....	19
2.4 Požadavky na vytvoření jednoho pracoviště	21
PRAKTICKÁ ČÁST.....	22
3 NÁVRH A REALIZACE VZDÁLENÉ LABORATOŘE	23
3.1 iSES – Internet School Experimental System	24
3.1.1 Základní princip fungování iSES Remduino Lab	25
3.1.2 Spuštění vzorového příkladu.....	26
3.1.3 Základní funkce v iSES Remduino Lab.....	30
3.1.4 Potřebné znalosti	31
3.1.5 Arduino IDE	32
3.1.6 Webové skripty.....	41
3.2 Experiment s hradly	48
3.2.1 Použitá zařízení a součástky	49
3.2.2 Skripty	54
3.2.3 Výroba.....	58
3.3 Vzdálený přístup k experimentu.....	60
3.4 Porovnání vzdáleného a reálného experimentu	62
3.4.1 Porovnání finanční náročnosti.....	63
3.4.2 Porovnání časové náročnosti.....	65
ZÁVĚR	66
SEZNAM POUŽITÉ LITERATURY	68
SEZNAM OBRÁZKŮ	71
SEZNAM TABULEK.....	73
PŘÍLOHA.....	74

Úvod

Konečné rozhodnutí při výběru tématu bakalářské práce značně ovlivnila moje soukromá exkurze v kabinetu docenta RNDr. Františka Lustiga, CSc. na Matematicko-fyzikální fakultě Univerzity Karlovy v Praze, ve kterém se nachází řada funkčních otevřených vzdálených experimentů. Tyto experimenty jsou všem přístupné z každého koutu zeměkoule, kde je dostupná síť internetu. Na pár čtverečních metrech se tu potkává historie se současností návrhu a tvorby vzdálených experimentů. Lze říci, že doc. RNDr. František Lustig, CSc. se svým týmem v oblasti vzdálených laboratoří především zpočátku dalece předběhl dobu. V současnosti je vytvoření laboratoře a její umístění na internet díky technickým možnostem snazší i pro běžné uživatele. Mohl jsem si prohlédnout všechny reálné konstrukce vzdálených experimentů včetně hardware i software systému iSES Remduino Lab. Doc. RNDr. František Lustig, CSc. ve svých článcích ukazuje nejen učitelům, že navržení a sestavení vzdáleného experimentu je v možnostech každého člověka, jen to občas může stát nějaké úsilí. Je vždy ochoten poradit zájemcům s vytvořením podmínek pro novou vzdálenou laboratoř.

V teoretické části bakalářské práce jsou zmapovány metody výuky včetně metod názorně-demonstračních, vzdálené laboratoře a školní pokusy. Z diagramů, jako je např. pyramida učení, či Daleho kužele, je jasné patrné, že oproti klasické výuce je mnohem vhodnější použití audiovizuálních a názorně-demonstračních metod oproti čtení textu či poslechu přednášky, protože se zvyšuje úspěšnost zapamatování nových informací. Modelování nebo simulace reálné zkušenosti spolu s přímou cílevědomou zkušeností tvoří základnu Daleho kužele. Proto se při výuce nejen v technických předmětech osvědčilo zařazení reálných experimentů, popřípadě využití laboratoří se vzdáleným přístupem či nově definovaných blízkých vzdálených experimentů.

Cílem praktické části je ověření reálných možností obyčejného člověka bez větších technických zkušeností k vytvoření vlastní vzdálené laboratoře v domácím prostředí. K realizaci byl díky ochotě doc. Lustiga využit systém iSES Remduino Lab a prototypovací platforma Arduino UNO. Pilířem návrhu laboratoře se stalo 5 úloh s hradly, jinak nazývanými též logické členy. Pochopení logických členů, potažmo výrokové logiky, je dobré pro rozvoj kreativního technického myšlení. Tyto úlohy mají za cíl ověření základních funkcí. Úlohy jdou od nejjednodušší funkce logických členů a postupně gradují ve své náročnosti. K hardwarové realizaci úloh byly použity integrované obvody zapojené pomocí nepájivých polí.

Zkušenosti s realizací vzdálené laboratoře byly použity pro potřebu porovnání výhod i nevýhod reálného experimentu a laboratoře se vzdáleným přístupem. Srovnávány byly faktory po stránce časové a finanční náročnosti, dále pak byly nastíněny předpokládané dopady experimentu na studenty.

Při řešení zadání bakalářské práce byly postupně použity následující metody: Teoretické studium odborné i didaktické literatury, analýza výukových metod především z pohledu názorně demonstračních metod, syntéza získaných poznatků, vlastní tvorba vzdálené laboratoře a ověření její funkčnosti.

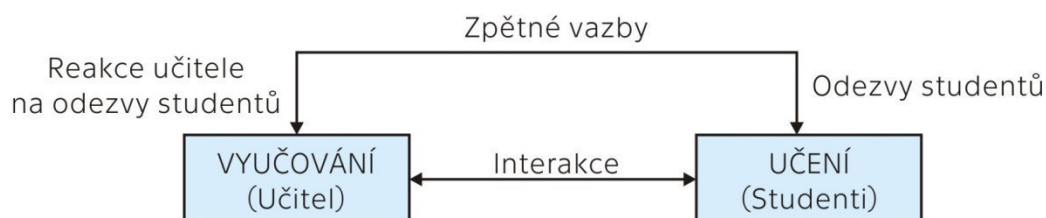
TEORETICKÁ ČÁST

1 METODY VÝUKY

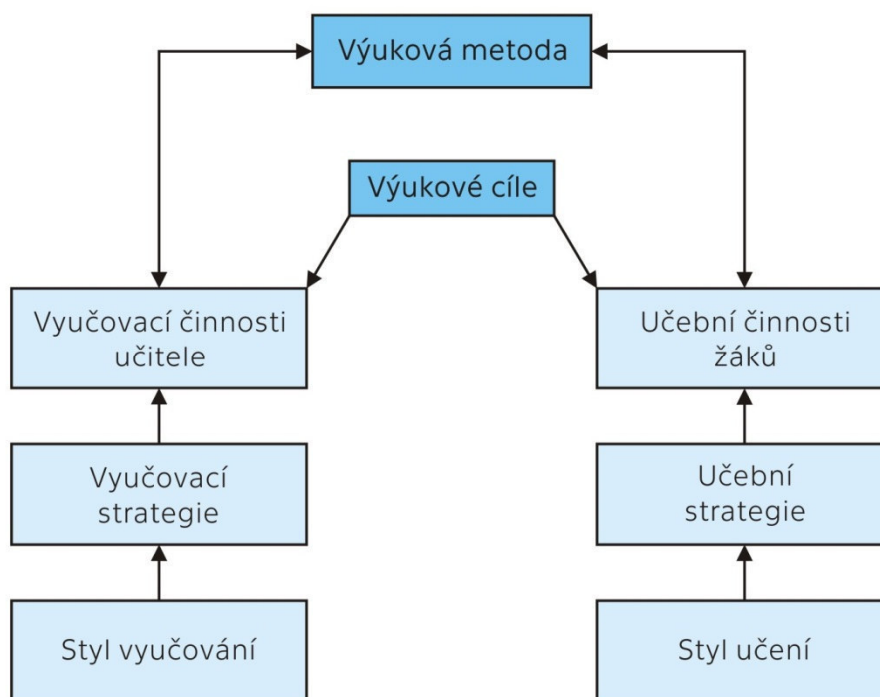
Prostřednictvím výuky probíhá komunikace a interakce mezi učitelem a studentem / studenty (obrázek 1). Cílem výuky je, aby vhodnou kombinací metod bylo dosaženo poznávacích, operačních a hodnotových cílů. Ve výuce je možno rozlišit vyučování, které je činností učitele a učení, které je činností studenta. Tyto dva procesy tvoří základ pedagogické komunikace ve škole (obrázek 2).

Pojmem výuková metoda je v moderní pedagogické literatuře označována současně činnost učitele i činnost studenta, přestože ve starší literatuře je možno se setkat i s pojmem vyučovací metoda.

S metodami výuky úzce souvisí metodický postup, kterým je označována část metody výuky. Různé metodické postupy aplikované na konkrétní učivo přináší z hlediska účinnosti různé výsledky, přestože by byla použita stejná výuková metoda. Právě metodický postup výrazně odlišuje výklad jednotlivých vyučujících. [1, 2, 3]



Obrázek 1 Vztah mezi vyučováním a učením [2]



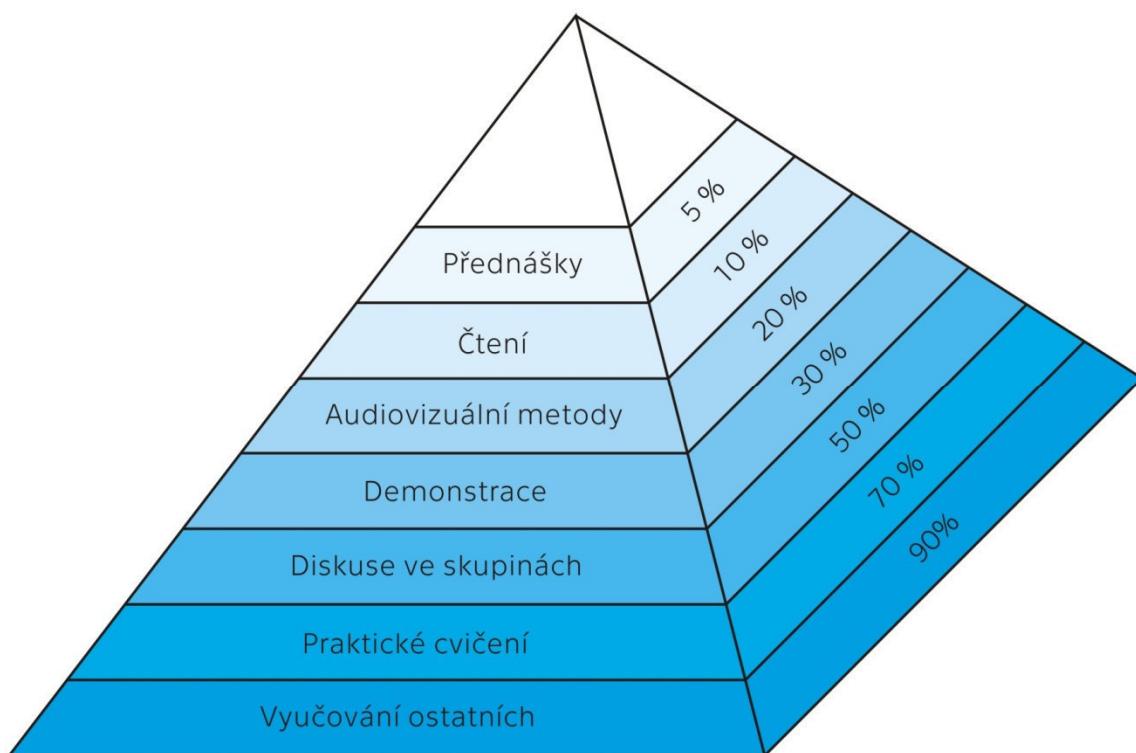
Obrázek 2 Vztahy mezi výukovou metodou, strategií učení/vyučování a stylem učení/vyučování [2]

Výukových metod existuje velké množství a mnozí pedagogičtí a vědečtí pracovníci se zabývali jejich účinností na učení studentů. Mezi nejznámější schémata účinnosti různých metod patří např. Kužel zkušenosti sestavený E. Dalem roku 1946 (později byl tento model aktualizován), nebo Pyramida učení, kterou publikoval S. Shapiro v roce 1992 (obrázek 3). Pyramida učení znázorňuje procento zapamatování poznatků s použitím vybraných výukových metod. [1]

Přestože se jednotlivé modely účinnosti výukových metod v detailech liší, shodují se v základním rozvrstvení – mezi nejméně efektivní metody patří poslech přednášek a čtení, mezi nejefektivnější metody patří praktická činnost, či výuka dalších lidí. Je možno tvrdit, že čím aktivněji je student zapojen do výuky, tím snazší a efektivnější je osvojování konkrétních vědomostí, případně dovedností. Modely také ukazují, že pro pochopení a zapamatování dané problematiky je vhodné použít audiovizuální metody, případně demonstrační metody. [1, 2]

Vzhledem k velkému množství výukových metod je možné se setkat s jejich roztříděním na základě nejrůznějších kritérií. V české didaktice je nejrozšířenější a nejuznávanější třídění podle J. Maňáka a I. J. Lernerera. [4]

Volba konkrétní metody závisí především na typu a stupni školy, vymezených cílech a úkolech výuky, obsahu a metody daného oboru vyučovaného předmětu, vnějších podmínkách výuky, zákonitosti výukového procesu a z nich plynoucí didaktické zásady, psychosociální charakteristice studentů a jejich učebních možnostech a osobnostních předpokladech. [5]



Obrázek 3 Model pyramidy učení dle S. Shapira [1]

Výukové metody podle I. J. Lernerera [1, 4, 6]

Klasifikace výukových metod podle I. J. Lernerera vznikla v roce 1986 a jako hlavní kritérium bere v potaz poznávací činnosti studenta při osvojování obsahu učební látky. Podle přístupu studenta lze rozlišit metodu reproduktivní (Informačně receptivní a reproduktivní metoda), hraniční (Metoda problémového výkladu) a produktivní (Heuristická a výzkumná metoda).

Výukové metody jsou rozděleny do pěti skupin:

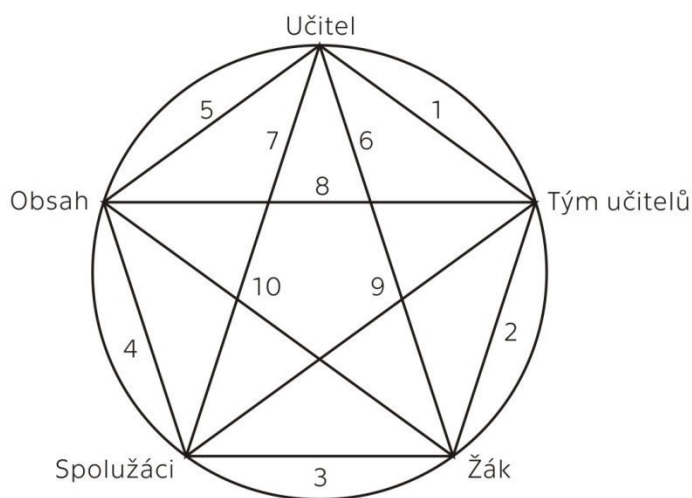
- Informačně-receptivní metoda
- Reproductivní metoda
- Metoda problémového výkladu
- Heuristická metoda
- Výzkumná metoda (částečně badatelská)

Výukové metody podle R. Winkela [2, 7]

Klasifikace výukových metod podle R. Winkela vznikla v roce 1987 a jejím cílem je komplexně rozdělit výukové metody na základě komunikativnosti výuky do pěti vztahových pólů (obrázek 4):

- Učitel
- Tým učitelů
- Žák
- Spolužáci
- Obsah

Na základě vztahů mezi póly lze výukové metody dělit do 17 skupin. Mezi dvoupólovou interakci Žák-Obsah patří: individuální práce, programová výuka, písemná práce a domácí práce. Do metod třípólové interakce Žák-Spolužáci-Obsah patří partnerská výuka, práce v malých skupinách, práce ve velkých skupinách a simulativní postupy. Metody čtyřpólové interakce Učitel-Žák-Spolužáci-Obsah zahrnuje výklad učitele, aktivity žáka, experimentování, rozhovor, dotazování žáků, diskusi, poradou a debatu. Do metod pětípólové interakce lze zahrnout týmovou výuku.



Obrázek 4 Členění metod podle komunikačních faktorů [2]

Výukové metody podle J. Maňáka [2, 4, 8, 9]

Klasifikace výukových metod podle J. Maňáka je v české didaktice preferována nejvíc, ze všech klasifikací. První komplexně pojaté uspořádání bylo publikováno v roce 1990 a je členěno podle aspektů na:

- Aspekt didaktický
- Aspekt psychologický
- Aspekt logický
- Varianty metod z hlediska fází výuky
- Aspekt organizační

Nicméně v poslední době došlo ke značnému vývoji ve výukových trendech a metodách a bylo tak potřeba klasifikaci inovovat. Novou klasifikaci vytvořil J. Maňák v roce 2003, kde třídí metody výuky podle složitosti edukačních vazeb mezi učitelem a žáky. Základní dělení je do tří skupin, které jsou dále děleny do podskupin:

- Klasické výukové metody
 - Metody slovní
 - Metody názorně-demonstrační
 - Metody dovednostně-praktické
- Aktivizující metody
 - Metody diskusní
 - Metody heuristické, řešení problémů
 - Metody situační
 - Metody inscenační
 - Didaktické hry
- Komplexní výukové metody
 - Frontální výuka
 - Skupinová výuka
 - Partnerská výuka
 - Individuální a individualizovaná výuka, samostatná práce žáků
 - Kritické myšlení
 - Brainstorming
 - Projektová výuka
 - Učení dramatem
 - Otevřené učení
 - Učení v životních situacích
 - Televizní výuka
 - Výuka podporovaná počítačem
 - Sugestopedie a superlearning
 - Hypnopedie

Velkou popularitu v současné době získává výuka podporovaná počítačem. Tento způsob výuky umožňuje zpřehlednit látku, vysvětlit ji a případně zopakovat, přičemž je přihlíženo k individuálnímu tempu jednotlivých studentů. Počítače umožňují vytváření simulací, testování, případně sebetestování a rozšiřují informační prostor studenta. Je však nutno dbát zásad ochrany zdraví při práci, které se týká především očí.

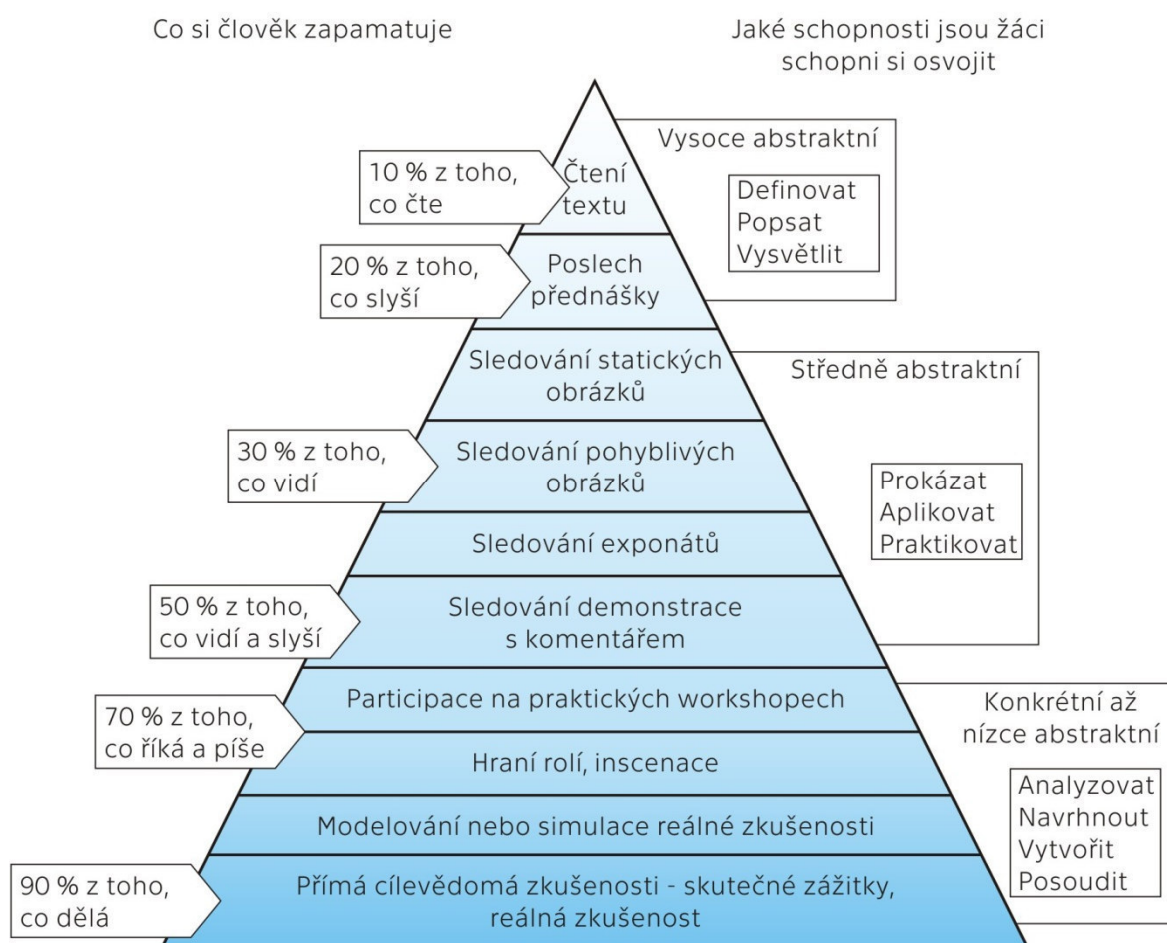
Daleho kužel zkušenosti

Kužel zkušenosti je diagram, který zobrazuje vztah efektivnosti vyučování v závislosti na užití výukové metodě (obrázek 5) a byl sestaven Edgarem Dalem, který patřil mezi uznávané odborníky na audio-vizuální materiály. Při svém výzkumu vycházel ze zkušeností a pozorování žáků. První verze kužele byla sestavena už v roce 1946 a zobrazovala průběh získávání zkušeností s tím, že na podstavě kuželu byly aktivity s nejvyšší účinností, směrem do vrcholu účinnost výukových metod klesá. Model byl postupně doplňován a rozšiřován. Z diagramu vyplývá, že čím více se výuková metoda přibližuje reálným věcem, tím je účinnější. Přestože se jedná o výsledky výzkumů, které byly realizovány v minulosti, jsou stále platné a doposud nebyly vyvráceny.

Jednotlivé části diagramu korespondují i se smyslovými vjemy, které daná výuková metoda podporuje. Praktické činnosti nutí studenty zapojit všechny smysly, směrem do vrcholu kužele nutnost zapojení smyslů ubývá. Nejméně přínosnou metodou je tak čtení textu, bez jakýchkoliv obrázků.

Dalův kužel zkušenosti je také někdy do češtiny překládán jako kužel učení, nebo kužel abstrakce. [10]

Překlad Dalova kužele zkušenosti do českého jazyka a jeho úpravu obstaral např. J. Dostál. [11]



Obrázek 5 Daleho kužel zkušenosti

Z různých hledisek mohou být posuzovány i styly učení a záleží jen na konkrétním studentovi, který způsob mu vyhovuje nejvíce. Styl učení se v průběhu života člověka může i měnit, a to na základě nových poznatků a zkušeností s různými typy učení.

Dělení stylů učení podle percepční preference [12]:

- Vizualní typ – vizualní typ studentů si nejnadhěji zapamatuje informace, které vidí ať už v psané podobě, nebo ve formě obrázku. Zapamatování informací také často napomáhá např. barevné zvýraznění učebního textu. Do vizualního typu lze také zařadit studenty s tzv. fotografickou pamětí. Tento způsob učení převládá především u malých dětí, v dospělosti pak často dochází ke změně způsobu učení.
- Auditivní typ – tento typ studentů má problém se zapamatováním informací z psaného textu, mnohem snáze se jim učí slyšené. Velkou pomocí pro tento typ studentů je, pokud mohou s vyučujícím na dané téma vést diskuzi.
- Kinestetický typ – pro studenty preferující kinestetický typ učení je důležité, aby se mohli danému problému aktivně věnovat, vše si vyzkoušet, nebo alespoň osahat. Při výkladu pedagoga kinesteticky zaměřeným studentům pomáhá i hraní si s různými předměty, což však není všemi vyučujícími vždy dovolováno. Důraz na tento způsob učení klade např. Waldorfská pedagogika.

Styl učení podle R. Dunnové [13, 14]:

Styly učení podle R. Dunnové jsou děleny na základě biologicky a vývojově podmíněné charakteristiky, která ovlivňuje studenta. Faktory, které ovlivňují učení, jsou podle tohoto přístupu děleny do pěti skupin:

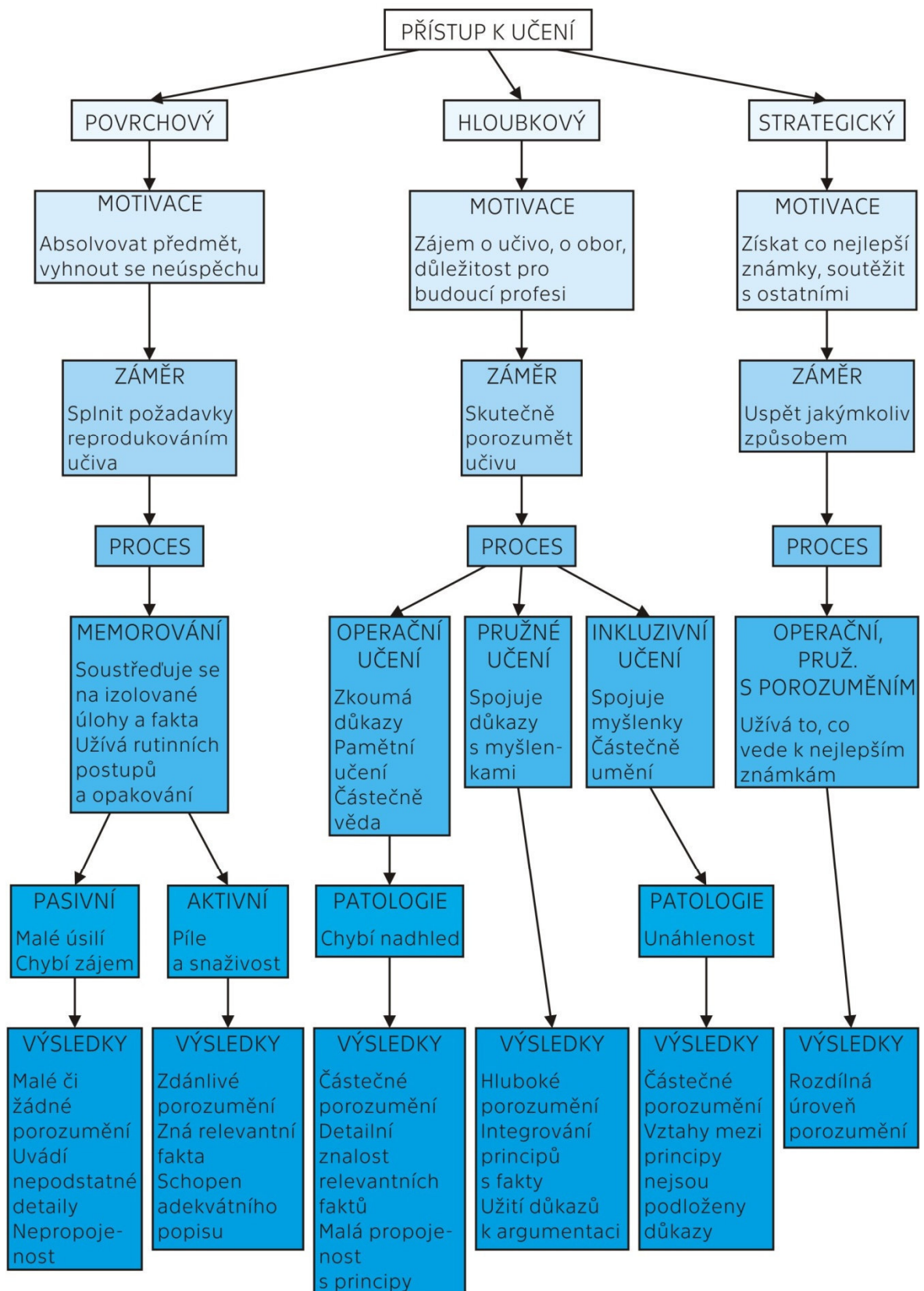
- Vlivy fyzikálního prostředí (zvuky, pracovní místo, teplota, osvětlení, nábytek v okolí pracoviště, ...)
- Intencionální aspekty (motivace, odpovědnost, vytrvalost, struktura, resp. flexibilita při učení)
- Sociální potřeby (učení o samotě nebo ve skupině – lze rozlišit velkou a malou skupinu, případně učení v páru)
- Psychofyzilogické potřeby (preference denní doby, nutnost přijímat, případně nepřijímat potravu a pití během procesu učení, potřeba pohybu, preference zapojení určitého smyslu – hmat, sluch, ...)
- Styl zpracování

Tyto aspekty jsou důležité především při vzdělávání studentů se specifickými potřebami učení.

Dělení stylů učení podle motivace a záměru [15]:

- Povrchový styl učení
- Hlubkový styl učení
- Strategický styl učení

Volba konkrétního stylu učení se odvíjí od obsahu učiva a od přístupu studenta k dané problematice. Zároveň je však žádoucí, aby zvolený styl nutil studenta k posunu (obrázek 6).



Obrázek 6 Model vztahů mezi motivací, záměry, průběhem a výsledky učení [13]

Metody názorně-demonstrační [1]

Výzkumy ukazují, že pro pochopení a zapamatování si informací je velice prospěšné doprovodit výklad konkrétní látky konkrétním pozorováním, předváděním, či experimentem. Jednotlivé modely se přibližně shodují, že úspěšnost zapamatování si informací při použití názorně-demonstračních metod se pohybuje okolo 30 % – 50 % (obrázek 1, obrázek 5).

Demonstrace různých jevů je obzvláště důležitá v technických předmětech, kdy textový popis často nedokáže dát studentovi zcela jasnou představu o dané problematice. Přínos předvádění pro studenty technických předmětů je především v oblastech zabývajících se stavbami, stroji a přístroji, jejichž funkci by bylo velice obtížné popsat pouze slovně. Pokud není možné pro demonstraci použít přímo samotný výrobek, je vhodné využití modelů, ať už fyzických nebo počítačových.

Pozorování dané problematiky může být prováděno buďto jednorázově (např. při výkladu na hodině či při exkurzi), nebo dlouhodobě, kdy student intenzivně sleduje daný jev po určitou dobu (především v rámci dlouhodobějších projektů).

Metody praktické činnosti žáků [1]

Metody praktické činnosti žáků / studentů, označované také jako metody dovednostně-praktické úzce souvisejí s názorně-demonstračními metodami. Používají se především při výuce odborných předmětů a praktickém výcviku. Studenti jsou při použití této metody vybaveni teoretickým základem k dané problematice a následně přichází do přímého kontaktu s objekty, přičemž s nimi mohou sami manipulovat.

Mezi tyto metody patří i laboratorní cvičení a provádění experimentů. Tím jsou studenti více zapojováni do výuky konkrétního učiva (je zapojeno více smyslových vjemů) a dochází tak k snazšímu osvojení dané problematiky, což je velká výhoda oproti názorně-demonstračním metodám.

Experimenty mohou probíhat buďto ve škole, v rámci vyučovací hodiny, nebo v domácím prostředí – pokud to daný experiment umožňuje.

Pokusy a experimenty mohou studenti provádět buďto samostatně, nebo i ve skupinách dle instrukcí vyučujícího, případně návodu v knize či pracovním listu.

Laboratorní úlohy lze dělit podle jejich charakteru na:

- ověřovací,
- důkazové,
- praktické,
- výzkumné.

Dlouhodobé výzkumy ukazují, že při aktivním zapojení studentů do konkrétní činnosti dochází k zapamatování 70 % – 90 % sdělovaných informací (obrázek 1, obrázek 5). Z tohoto důvodu je vhodné co největší zařazení praktických činností do výuky. Sekundární výhodou praktických činností je, že je při nich rozvíjena studentova zručnost a schopnost pracovat s přístroji a zároveň dochází k rozvíjení nejen samostatnosti, ale i spolupráci s kolektivem v případě skupinově prováděných experimentů.

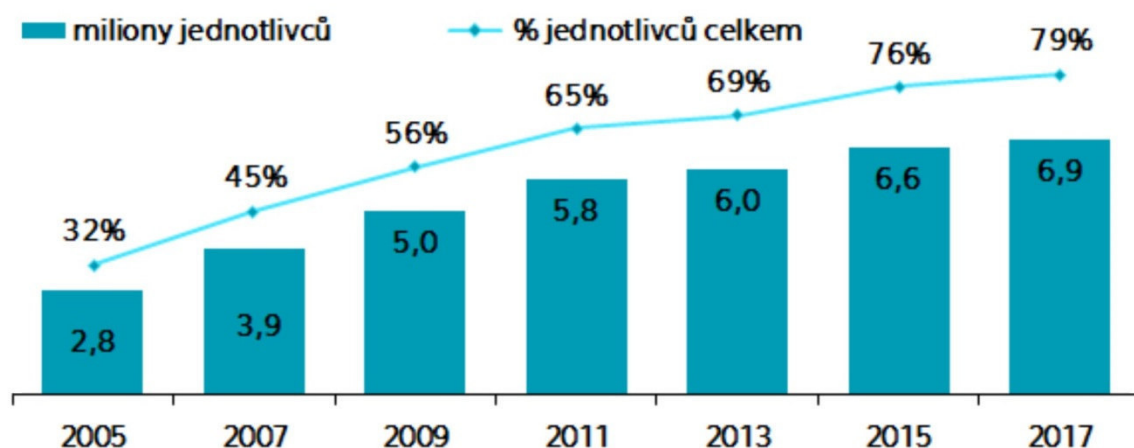
U většiny experimentů je nutno věnovat zvýšenou pozornost bezpečnosti práce a hygieně. Z tohoto hlediska jsou výhodné např. vzdálené laboratoře, o kterých pojednává následující kapitola.

2 VZDÁLENÉ LABORATOŘE

Existence vzdálených laboratoří by nebyla možná bez vytvořených podmínek informačně komunikačních technologií, tedy vzniku a masového rozšíření počítačové sítě internet.

2.1 Historie internetu

Politická i vojenská rivalita mezi Spojenými státy a bývalým Sovětským svazem, která vznikla po skončení druhé světové války, podnítila značnou podporu rozvoje nových zbraní i technologií. To vše ve snaze získat nad rivalem mocenskou převahu. Proto v roce 1969 vznikl v USA experimentální program Advanced Research Project Agency, nazývaný také ARPANET. Byla to vlastně malá počítačová síť, která zaručovala rychlý a hlavně bezpečný přenos informací mezi národní obranou, armádou a univerzitní laboratoří, zabývající se vojenským výzkumem. Krátce na to program začal sloužit i obchodním společnostem pod názvem DARPA Internet. Později zůstalo označení *Internet*, které zobecnělo. Slovo internet vzniklo z latinské předpony „inter“, česky znamenající „mezi“, a z anglického slova „network“, česky „sít“. Z původně malé experimentální počítačové sítě, propojující asi jen deset stanovišť ve Spojených státech, se na počátku devadesátých let stalo globální interaktivní médium, šířící se do celého světa obrovskou rychlostí. Hlavním jazykem internetu byla angličtina, která díky němu pronikla do celého světa. Již 4 roky po jeho zpřístupnění široké veřejnosti získal 50 milionů uživatelů. Pro srovnání, rádio získalo stejný počet uživatelů za 30 let a televize za 13 let. V roce 2005 měl internet 950 milionů uživatelů a na konci roku 2008 již 1,58 miliardy uživatelů, což bylo 23,6 % z celkové světové populace. K 31. březnu 2011 užívalo internet přes 2 miliardy lidí. K navyšování množství uživatelů internetu dochází i v rámci České republiky (obrázek 7).

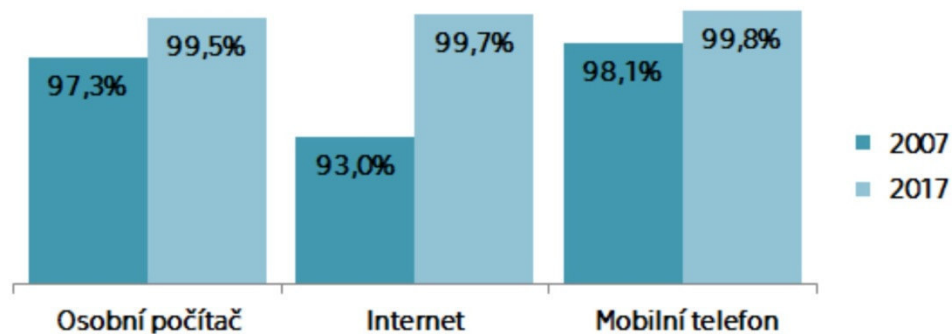


Obrázek 7 Množství jednotlivců používajících internet [17]

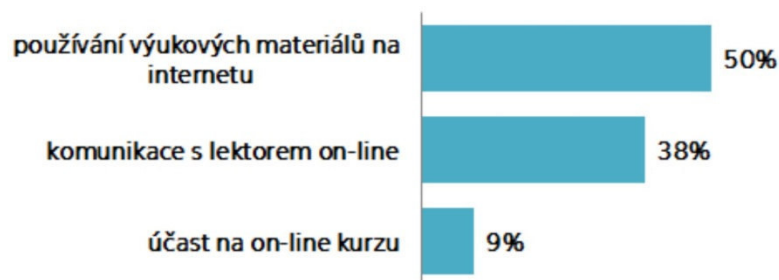
První pokusy o vytváření počítačové sítě se v bývalém Československu prováděly pomocí telefonních linek. V listopadu 1991 proběhly první neoficiální pokusy o připojení k internetu. První linka vedla z ČVUT v Praze do internetového uzlu v Linci. Oficiál-

ní připojení pak proběhlo 13. února 1992 opět v ČVUT. Dalšímu rozvoji bránil monopol firmy Eurotel, který se mimo jiné vztahoval i na veřejné služby přenosu dat. Komerční poskytovatelé připojení k internetu vstoupili na trh až na přelomu let 1995 a 1996. Od té doby užívání internetu rychle roste a slouží k přenosu nejrůznějších dat. Síť internet slouží nejen firmám, ale užívají ho i školy a zaveden je i v domácnostech. Připojena jsou i rekreační zařízení, knihovny, obchodní centra nebo čekárny lékařů. Služby internetu je možno využít i při jízdě vlakem. V současné době je internet rozšířen i díky možnostem, které nabízejí tzv. chytré telefony. [16]

Rozvoj tohoto informačního média byl prvním předpokladem pro vývoj a použití vzdálených laboratoří při výuce. Komunikační protokoly a prohlížeč Web 1.0 z počátku 90. let umožňovaly pouze pasivní prohlížení webových stránek. V této době si učitelé mohli stahovat z internetu pouze simulace či virtuální experimenty. S přelomem milénia přinesl pokrok vývoj Webu 2.0, který umožňoval aktivní zpětnou vazbu. To znamená, že vznikla možnost modifikace webových stránek v reálném čase. Vznik internetu a Webu 2.0 se staly základními kameny reálně řízených vzdálených experimentů. Svoji důležitou roli ve vývoji sehrál například projekt MŠMT Indoš (Internet do škol, 2003), prosazený v rámci *Koncepce státní informační politiky ve vzdělávání* (usnesení vlády č. 525/1999 z 31. 5. 1999, bod III.3.), jež se podílel na rozšíření internetu a informačně komunikačních technologií především na základních a středních školách. Dnes tedy, jak je patrné ze statistik Českého statistického úřadu, používání internetu studenty se blíží 100 procentům. Zároveň se stále zvyšuje podíl internetové komunikace mezi školou, učitelem a studentem (obrázek 8, obrázek 9).



Obrázek 8 Použití vybraných informačních technologií studenty [17]



Obrázek 9 Studenti využívající internet k vybraným vzdělávacím aktivitám [17]

2.2 Školský pokus – experiment ve výuce přírodních věd

Proces fylogeneze člověka nezpochybnitelně provází zkoumání okolního přirozeného prostředí, následované vystávajícími otázkami. Již staří filosofové pro svá prvotní zkoumání používali metodu pozorování. Pozorování je základním způsobem, jak na otázky nalézt odpovědi, popisující výskyt okolních (především fyzikálních) jevů, za kterými následují pokusy o zobecnění těchto jevů. Je-li zkoumán nový, neznámý jev, je tato empirická činnost efektivní. Pokud je však požadavkem známou zákonitost přesněji specifikovat nebo použít pozorování jako výukovou metodu, je nutné nejdříve provést důslednou logickou analýzu. Na jejím základě pak lze navodit opakovatelný jev s přesně stanovenými podmínkami. Taková metoda je nazývána experimentem (pokusem). Kvalitativní experiment prokazuje či neprokazuje existenci jevu. Kvantitativní experiment pak poskytuje měřitelné údaje fyzikálních zákonů či teorií. [18]

Demonstrační pokus ve výuce může být předváděný učitelem či společně učitelem a žákem celé třídy. Takový pokus především formuje žákům základní představy o zkoumaném jevu. Žákovský pokus, který provádějí žáci při výuce, přivádí žáka do bližšího kontaktu s řešenou problematikou. Propojuje kognitivní a psychomotorické dovednosti, rozvíjí tvůrčí aktivity a v neposlední řadě hraje klíčovou roli při osvojení učiva. Žák se zároveň má možnost rozhodnout při výběru řešení úloh. Má možnost ověřovat výsledky teoretického výpočtu například při sériovém či paralelním zapojení několika rezistorů. Daný pokus sám přivádí žáky ve třídě k diskuzi o způsobu řešení i výsledcích a v neposlední řadě i o využití v praxi. Tím se přirozeně rozvíjejí komunikační kompetence a sociální vztahy. [1]

Toto je v souladu s moderním pojetím vyučování, kdy se přestává klást důraz na memorování velkého množství poznatků, ale orientace je zaměřena na rozvoj v rovině kognitivní (rozvíjení tvořivosti, kritického myšlení a strategie řešení problémů) a v rovině sociální (rozvíjení komunikace, schopnosti vyjednávání a týmové spolupráce). [19]

Většinu demonstračních pokusů žáci vnímají jako zábavnou a motivující činnost, ale po skončení pokusu někteří žáci nedokáží vysvětlit, proč byl pokus prováděn a co bylo cílem jeho provedení. Je tedy důležité správně stanovit cíl pokusu, aby bylo jasné, zda výsledkem pokusu je ilustrace jevu, ujasnění si vztahů v konkrétní zákonitosti či procvičování určité dovednosti. [20]

Podle funkce didaktické je možno rozdělit pokusy na heuristické, verifikační, motivační, ilustrační, historické, fixační, diagnostické a uvádějící fyzikální problém. Didaktickou funkci pokusu ve výuce určuje učitel. Stejnému pokusu může být přiřazeno více funkcí.

Demonstrační pokus lze rozlišovat podle provedení na reálný, virtuální a modelový. Při reálných pokusech jsou přímo zapojeny smyslové vjemy na konkrétní děj, probíhající v reálném čase. Prostřednictvím virtuálního pokusu jsou data získávána bez fyzického kontaktu s měřicími přístroji. Pokud je použit náhradní model místo originálu, hovoří se o modelovém pokusu. Model koncepčně vychází ze stejného principu jako

originál, je však modifikován na požadavky výuky. Zpravidla je model jednodušší konstrukce, je snadno rozebíratelný a má vhodně přizpůsobenou velikost. [1]

Demonstrační pokus – experiment, je možné rozdělit podle použitých prostředků na reálný, myšlenkový a podporovaný počítačem. Tradiční reálný experiment zaměřený na konkrétní děj přímo působí na smyslové vjemy a děj probíhá v reálném čase.

Realizace myšlenkového experimentu vychází ze známých zákonů, z nich deduktivně odvozených postupů a výsledků, a to za předpokladu ideálních podmínek. Tímto způsobem se mohou zkoumat vědecky možné, ale v praxi mnohdy neuskutečnitelné jevy (např. Shrödingerova kočka, či Maxwellův démon). Za myšlenkový pokus / experiment se též považuje přípravná fáze reálného pokusu, ve kterém učitel promýšlí, jaké materiální, technické a didaktické prostředky použije ke zdárnému provedení reálného pokusu. [18]

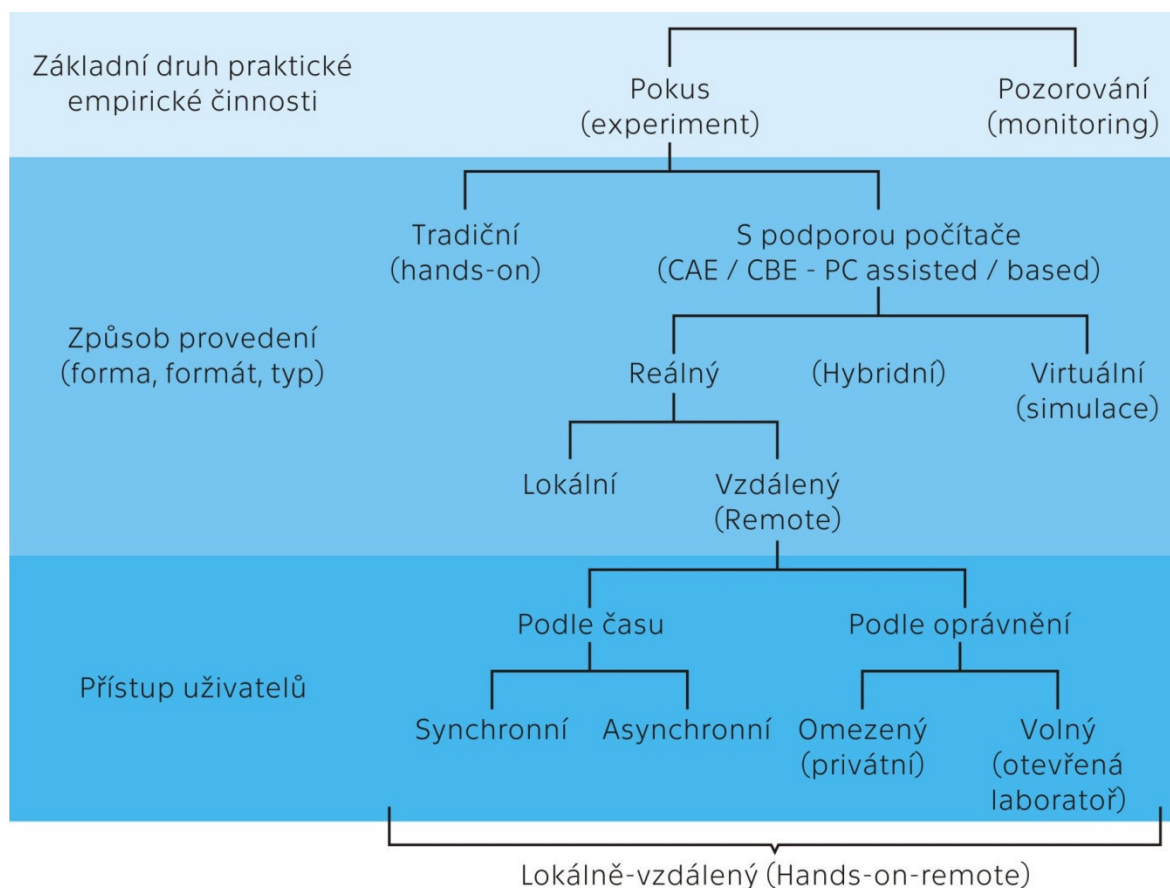
S přibývajícím věkem žáků se pokusy mění ve složitější komplexní laboratorní práce, při kterých se nejen zpracovávají data a protokoly, ale vyslovují se prognózy a hypotézy. Současně se získávají nové indicie pro provádění náročnějších experimentů. Ty je možné vykonávat nejen na speciálních školních pracovištích, ale třeba i v terénu. Experiment se dělí na vědecký, experiment ve výuce a praktický experiment. Specifičnost experimentu ve výuce spočívá v přesně stanovených podmínkách, které jsou opakovatelné, a tudíž je bezpečně známý výsledek. [2]

Díky technologické teorii vyvstal nový, nepředvídatelný problém. Zdá se, že učitel částečně přichází o kontrolu nad vyučovacím procesem. Specialisté z mediálních oborů, oborů informatiky, filmoví režiséři, zvukaři, a především programátoři a specialisté na vizuální efekty jsou nepostradatelní při výrobě didaktického materiálu, potřebného ke vzdělávacímu aktu. Učitel je tak mnohdy postaven do pozice statisty, kterému není umožněno ovlivnit cílování takto připravované didaktické pomůcky. To je důvodem, proč se učitelé vracejí k používání jednoduchých prostředků, které umožňují variabilně dle potřeb stanovovat výukové cíle. [21]

Příprava demonstračního pokusu či experimentu je pro učitele náročná disciplína zejména po časové stránce. Zařazení pokusu do výuky má svá pravidla z hlediska organizační formy výuky i bezpečnosti práce. Pokud je započítáno i časté omezení v podobě přístupnosti do laboratorních učeben, není divu, že učitel mnohdy volí zprostředkování pokusu formou vizuální prezentace ve třídě. Je třeba si uvědomit, že pokus – experiment má ve vyučovacím procesu svoji důležitou a nezastupitelnou roli a je proto jeho nedílnou součástí. Ideální variantou při provádění experimentu je zapojit žáky do činnosti nejvyšší možnou měrou.

2.3 Klasifikace experimentu s podporou počítače

Experimenty s podporou počítače lze dělit na základě různých kritérií (obrázek 10), jako je například způsob provedení či možnost přístupu uživatelů k experimentu. [22]



Obrázek 10 Rozdělení experimentů podle Broma [22]

S podporou počítače

Experiment podporovaný počítačem je možné rozdělit na reálný, virtuální a hybridní. Podle přístupu lze experiment rozdělovat na lokální a vzdálený. Podle času jsou experimenty děleny na experiment synchronní a asynchronní a dle oprávnění je možno odlišit experiment omezený (privátní) a volný (otevřený).

Tradiční *reálný experiment s podporou počítače*, zaměřený na konkrétní děj, přímo působí na smyslové vjemy a probíhá v reálném čase. Počítačem ovládané reálné experimenty zastupují převážně profesionální přístroje (Micronix), které umožňují měřená data přenést do počítače k dalšímu zpracování (např. dlouhodobé sledování počasí meteorologickou stanicí, zaznamenávání otřesů půdy, měření hluku, ...).

Virtuální experiment (simulaci) je možné označit jako nápodobu reálného experimentu, zprostředkovaném na obrazovce počítače. Graficky je napodobena struktura laboratoře včetně použitých komponentů a měřicích přístrojů. Významné postavení v tomto oboru získala University of Colorado Boulder díky svému projektu interaktivních simulací pro matematiku a přírodní vědy PhET. Nabízí nepřehledné množství simulací vhodné pro žáky a studenty od základní školy prvního stupně až po univerzitní

studenty. Poskytují studentům intuitivní hravé prostředí, které je určeno pro zkoumání a objevování přírodních zákonitostí. To vše je zaštiťováno pedagogickými výzkumy, které univerzita provádí. [23]

Další kategorií jsou takzvané *hybridní experimenty*. Prolínají se v ní obě předchozí kategorie – reálná a virtuální. Je to například simulace, matematicky zpracovávající reálná data, která se zobrazují dle zadaných parametrů.

Podle přístupu

Podle přístupu uživatele se jedná o experiment lokální tehdy, když je student přítomen v místě experimentu a manipuluje s laboratorní aparaturou. Pokud je laboratorní aparatura ovládaná vzdáleně (a dnešní ICT prostředky to umožňují), jedná se o přístup vzdálený (remote).

Podle času

Synchronní vzdálený experiment je charakterizován takřka bezprostřední změnou parametrů a okamžitými získanými naměřenými hodnotami. Například dne 23. března 2019 ve 20 hodin a 13 minut první časová odezva při ovládní vzdáleného experimentu iniciovaného Brazílskou univerzitou Universidad Federal de Santa Catarina en Brasil dostupné na webové stránce <https://rexlab.ufsc.br/> trvala 8 sekund. Odezva vzdálené laboratoře v tutéž dobu na Matematicko-fyzikální fakultě Univerzity Karlovy v Praze (vzdálenost cca 20 km) dostupné na webové stránce http://kdt-35.karlov.mff.cuni.cz/polarizace_univ.html se projevila za 1 sekundu.

Při asynchronním měření musí uživatel vyčkat fyzické obsluhy v laboratoři, bez které není možno provedení experimentu dokončit. Takové experimenty by ve školském prostředí měly být minimalizovány.

Podle oprávnění

Stěžejní myšlenka vytvořit otevřené laboratoře napříč školami i státy se zdá, že v současnosti stagnuje. Ať už jsou na vině obtíže při vytváření nových laboratoří nebo náklady a uvědomění si závazku v podobě trvalého servisu, počet veřejnosti otevřených laboratoří ve světě klesá. [24]

V roce 2004 vlastnilo například systém iSES, umožňující realizaci vzdáleného experimentu, zhruba 450 škol v České republice. Je tedy předpoklad, že je systém interně ve školách využíván. Ovšem školy své experimenty budou používat především pro výuku svých studentů. V tom případě tyto experimenty klasifikujeme jako privátní (veřejnosti nepřístupné). [25]

V roce 2018 zavedl nejnovější kategorii experimentu doc. František Lustig CSc. Použil označení *Hands-on-remote*, což v překladu znamená *lokálně vzdálený přístup*, respektive *lokálně blízký experiment*. Takový experiment je sestaven a vyzkoušen učitelem či žákem na aparatuře umožňující pozdější vzdálenou manipulaci. Experiment mohou okamžitě obsluhovat studenti přítomní v učebně při výuce. V tomto případě je experiment označován jako „Blízký vzdálený experiment“. Vzápětí ten samý experiment bez dalších úprav mohou využívat studenti mimo budovu školy. Tentokrát nese označení „Daleký vzdálený experiment“. [26]

2.4 Požadavky na vytvoření jednoho pracoviště

Velkou otázkou, kterou je nutno vyřešit před zřízením experimentu se vzdáleným přístupem, je, jaké jsou požadavky na vytvoření jednoho pracoviště otevřeného veřejnosti. Základním předpokladem pro provozování otevřené vzdálené laboratoře je přidělení veřejné IP adresy jednomu počítači v laboratoři. Ač se toto zdá jednoduchým požadavkem, v současnosti se internet dostal na rozcestí. Počet IP adres použitelných v tomto okamžiku je díky internetovému protokolu IPv4 udáván číslem 2^{32} , což je přibližně 4 miliardy adres. Zpočátku se to zdálo jako dostačující číslo, nicméně adresy protokolu IPv4 jsou již od roku 2011 vyčerpány. Z tohoto důvodu nově nastupuje protokol IPv6, jehož adresní prostor obsahuje 2^{128} adres. S protokolem IPv6 se naskýtá de facto 5×10^{28} adres pro každého člověka na planetě [27]. Nutností je však vlastnit zařízení podporující protokol IPv6. Zdá se tedy, že získání unikátní veřejné adresy nebude v budoucnosti alespoň po nějakou dobu problémem. S rozrůstajícím se tzv. internetem věcí však může opět nastat v protokolu IPv6 stejný problém, jaký je v současnosti řešen v souvislosti s protokolem IPv4. Stejně tak není v současnosti zcela jasné, kdy bude kompletně přestoupeno na fungování protokolu IPv6, byť je již v provozu.

Další problém však může být spatřován také ve vytíženosti vzdáleného pokusu. Nespornou výhodou vzdálené laboratoře je přístup dvacet čtyři hodin denně, sedm dní v týdnu, tři sta šedesát pět dní v roce. Pokud bude laboratorní úloha trvat např. dvacet minut a bude součástí výuky ve třídě s dvaceti pěti až třiceti žáky, tak je časové vytížení vzdálené laboratoře zhruba osm a půl hodin.

Používaná výpočetní technika pro zkonstruování vzdálené laboratoře může vykazovat v průběhu životnosti nejrůznější poruchy, ať již se bude jednat o zaseknutí či přehřátí počítače / serveru ovládajícího experiment nebo o jakýkoliv neočekávaný problém s počítačem, případně se samotným experimentem. Dalším problémem, který se může vyskytnout, je vypnutí ovládajícího počítače, případně jeho restart z důvodu aktualizací nebo pádu operačního systému. Také nelze vyloučit přerušení dodávky elektrické energie k experimentu. Je tedy zapotřebí počítat i s touto eventualitou a zajistit dálkovou správu vzbuzení počítače prostřednictvím technologie *WakeOnLan* včetně dálkového ovládání vypnutí a zapnutí síťového napětí (zdroje). V každém případě je pro bezproblémový provoz dobré ošetřit napájecí zdroj velkokapacitním záložním zdrojem (UPS). Zároveň je důležité zajistit a zaškolit i fyzickou obsluhu, která uvede v případě technických problémů aparaturu do funkčního výchozího stavu.

Ukazuje se, že při provozu otevřených vzdálených laboratoří by bylo příhodné vytvořit a aplikovat rezervační systém přístupů, který by potenciálnímu laborantovi rezervoval a sdělil čas, kdy bude aparatura připravena pouze pro něj. Předěšlo by se tak zklamání při přihlášení i případné ztrátě motivace. [28, 29]

PRAKTICKÁ ČÁST

3 Návrh a realizace vzdálené laboratoře

V praktické části této bakalářské práce je navržena a zrealizována jedna vzdálená laboratoř. Realizace takového zařízení vyžaduje hardwarové sestavení vlastního experimentu a softwarové naprogramování ovládacích a měřicích algoritmů.

Nejnsnazší zajištění komunikace mezi hardwarem experimentu a serverem je pomocí převodníku. V případě použití osobního počítače pro serverovou činnost je možné využít sériovou komunikaci přes USB port. Jako převodník je možné použít některou z prototypovacích platforem. Mezi nejrozšířenější prototypovací platformy patří mikroprocesorové desky Arduino a Raspberry Pi. Tyto platformy mají velkou komunitní základnu. I proto pro ně existuje celá řada různých modulů a čidel včetně podrobných návodů, jak vše zapojit a naprogramovat v jeden funkční celek.

Navržení a sestavení experimentu na základě Arduina naráží na problém způsobu komunikace mezi počítačem a Arduinem, který může být pro začínajícího uživatele přinejmenším matoucí. Díky komunitě lze najít více způsobů, jak komunikaci zajistit. V další fázi je ale nutné zajistit komunikaci počítače se světem pomocí internetu. V tu chvíli se obyčejná příprava malého experimentu rozrůstá a vyžaduje znalosti z více oborů. Jednou z možností je oželeť veškerý svůj osobní čas, naučit se vše od začátku a zpracovat dokonalé řešení. Nebo je možné použít řešení, které již někdo připravil, odzkoušel, příp. i odladil.

S rozvojem tzv. internetu věcí dochází k zaplňování trhu všemožnými možnostmi, jak komunikovat s jakýmkoliv hardwarem. Vždy je ale uživatel do jisté míry omezen a tlačěn využívat řešení nastíněné výrobcem. V takovém případě je nejelegantnějším řešením využít připravené řešení přímo pro vytváření vzdálených laboratoř.

Použitým řešením pro tvorbu vzdálené laboratoře proto bude systém iSES (Internet School Experimental Studio) od českých autorů z Matematicko-fyzikální fakulty Univerzity Karlovy v Praze. V následujících kapitolách bude zaznamenán detailní postup návrhu a realizace vzdáleného experimentu s využitím systému iSES v kombinaci s platformou Arduino (obrázek 11).



Obrázek 11 Oficiální loga systému iSES a prototypovací platformy Arduino [30, 31]

3.1 iSES – Internet School Experimental System

Systém iSES začal být vyvíjen pod vedením doc. RNDr. Františka Lustiga, CSc. v 90. letech dvacátého století. V původní podobě byl systém iSES určen pro laboratorní cvičení přímo v učebnách škol ve vyučovacích hodinách. Systém zahrnoval převodník určený k zabudování do počítače přes rozhraní PCI. Do systému bylo možné připojit mnoho různých senzorů (obrázek 12).

S postupem času z tohoto systému vznikl odvozený systém nazvaný **iSES Remote Lab SDK**. Jak lze z názvu vytušit, jednalo se nástroj pro realizaci vzdálených laboratoří. Systém iSES Remote Lab SDK využíval hardware původního systému.

Autoři nezůstali pozadu ani s vývojem situace ohledně změn operačních systémů a vývojem programovacích jazyků. V současné podobě je uživatelské rozhraní naprogramováno pomocí HTML (Hypertext Markup Language), Kaskádových stylů a JavaScriptu, což umožňuje multiplatformní přístup k experimentům.

Dalším důležitým krokem autorů systému iSES je reakce na stále oblíbenější prototypovací platformy. Proto došlo k přebudování iSES Remote Lab SDK pro platformu Arduino, čímž vznikl systém **iSES Remduino Lab**. Tento systém umožňuje nadšencům tvorbu dálkově ovládaných zařízení dle vlastních představ.



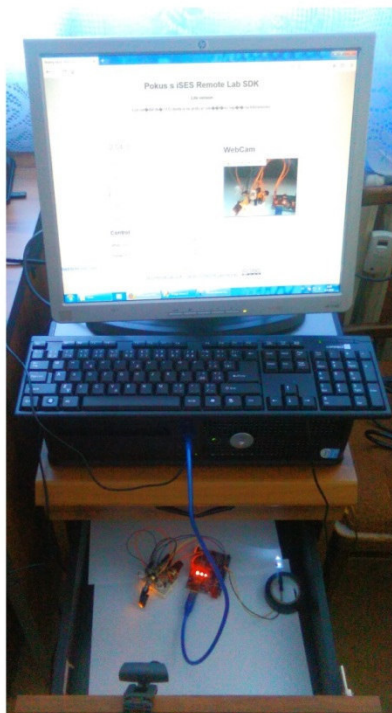
Obrázek 12 Pohled na systém iSES [30]

3.1.1 Základní princip fungování iSES Remduino Lab

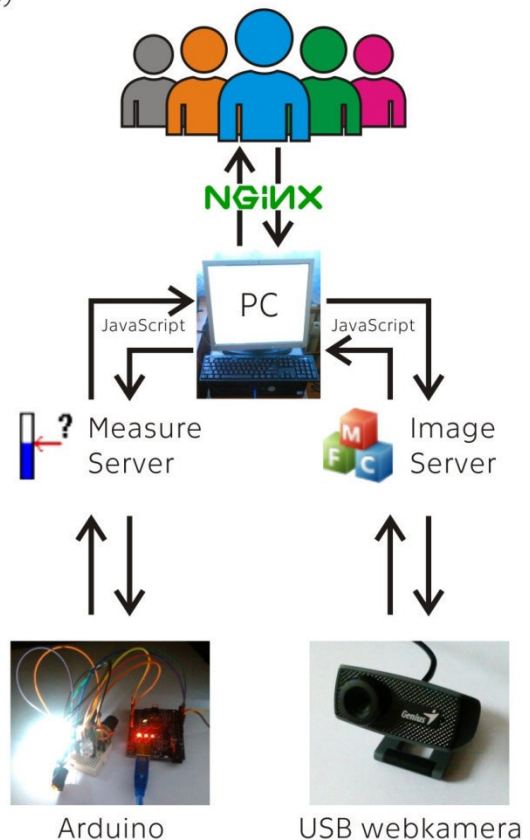
U všech vzdálených laboratoří je nutné zajistit komunikaci uživatele se serverem a komunikaci serveru s hardwarem experimentu. Tato komunikace musí být obousměrná, aby uživatel mohl experiment ovládat a zároveň zařízení předávalo naměřená data zpět serveru, resp. uživateli. Systém iSES Remduino Lab funguje na základě sériové komunikace serveru s mikroprocesorovou deskou Arduino pomocí USB portu. Server dále komunikuje s uživatelem pomocí TCP/IP protokolů.

Pro komunikaci s Arduinem byla týmem doc. RNDr. Františka Lustiga, CSc. vytvořena knihovna **Remduino**, sada skriptů v programovacím jazyce JavaScript a serverové aplikace [30]. Knihovna **Remduino** usnadňuje práci při tvorbě skriptu pro Arduino a obstarává komunikaci Arduina se serverovou aplikací **MeasureServer**, která ukládá přijatá data na disk a odesílá příkazy. JavaScript uložená data implementuje do HTML skriptu pro zobrazení uživateli a zároveň umožňuje předat **MeasureServeru** příkazy zadané uživatelem. Bez vizuálního kontaktu uživatele s experimentem by vzdálená laboratoř nedosahovala velké důvěryhodnosti. K tomu účelu slouží aplikace nazvaná **image_server**, která ukládá snímky z USB webkamery na disk. Uložené snímky jsou uživateli opět zobrazovány pomocí JavaScriptu. Pro vytvoření serveru z osobního počítače se systémem Windows je využíván HTTP a proxy server **nginx** (obrázek 13).

a)



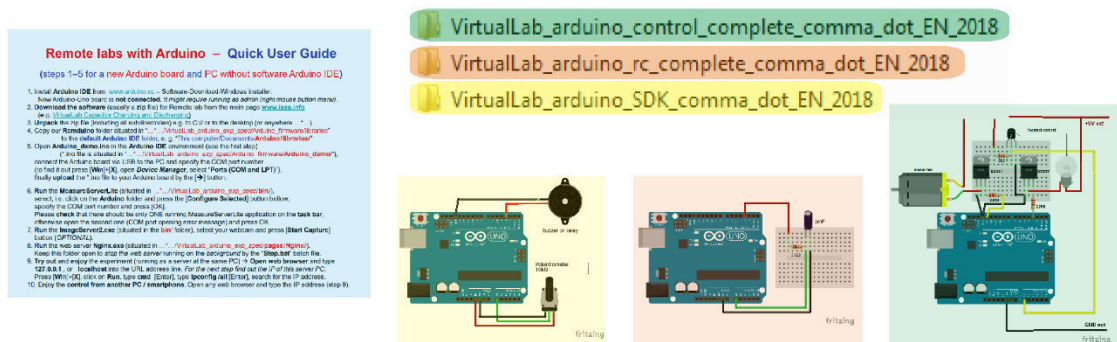
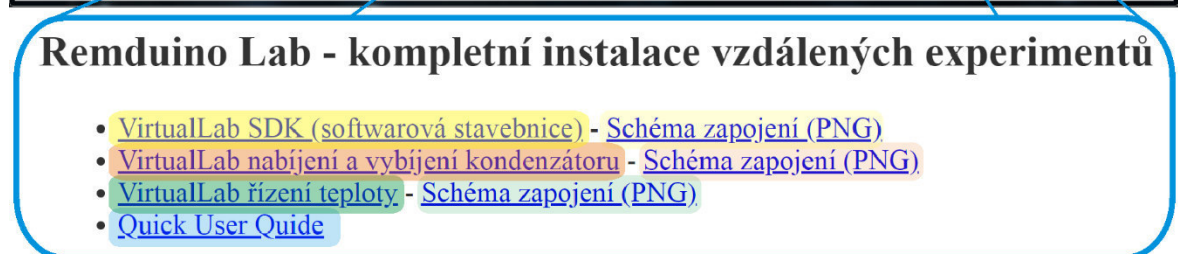
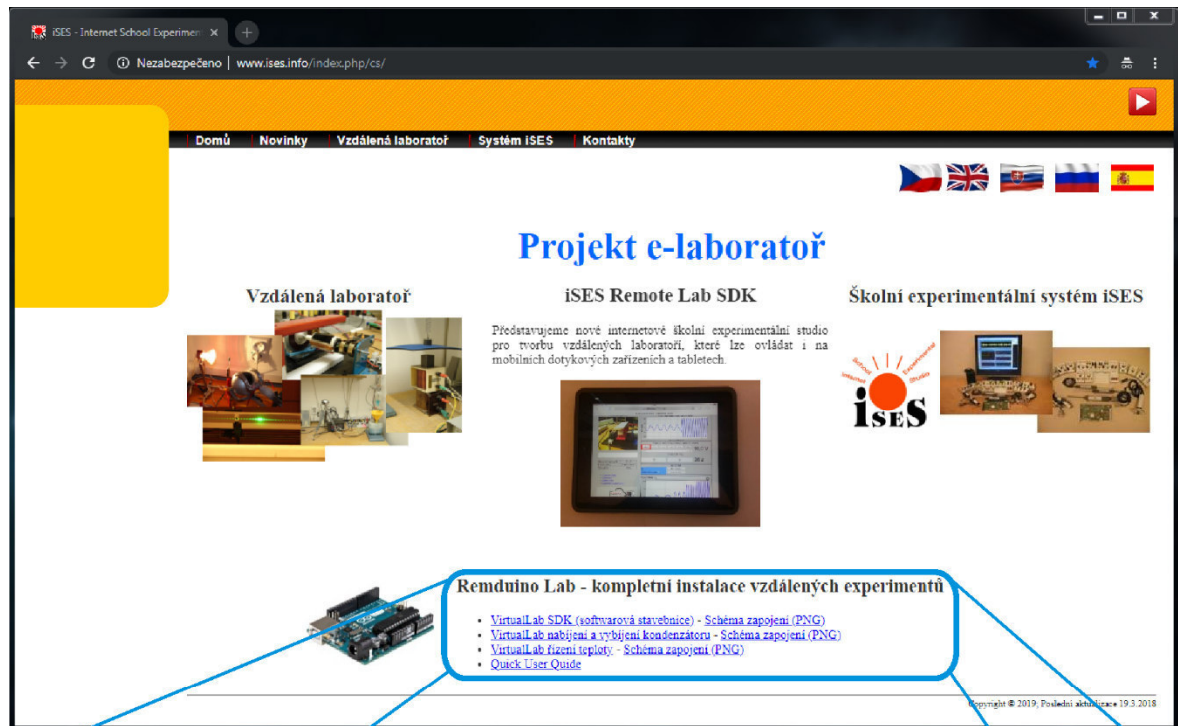
b)



Obrázek 13 Základní princip fungování vzdálené laboratoře
a) Fotografie plně funkční vzdálené laboratoře, b) Princip fungování iSES Remduino Lab

3.1.2 Spuštění vzorového příkladu

Pro zájemce o iSES Remduino Lab jsou na webových stránkách autorů [30] jednoduché příklady a anglický návod na spuštění (obrázek 14). Téměř všechny součásti jsou v plné verzi. Výjimku tvoří **MeasureServer**, který je ve verzi **Lite**, což znamená to, že je možné použít pouze jeden vstupní a jeden nebo dva výstupní prvky.



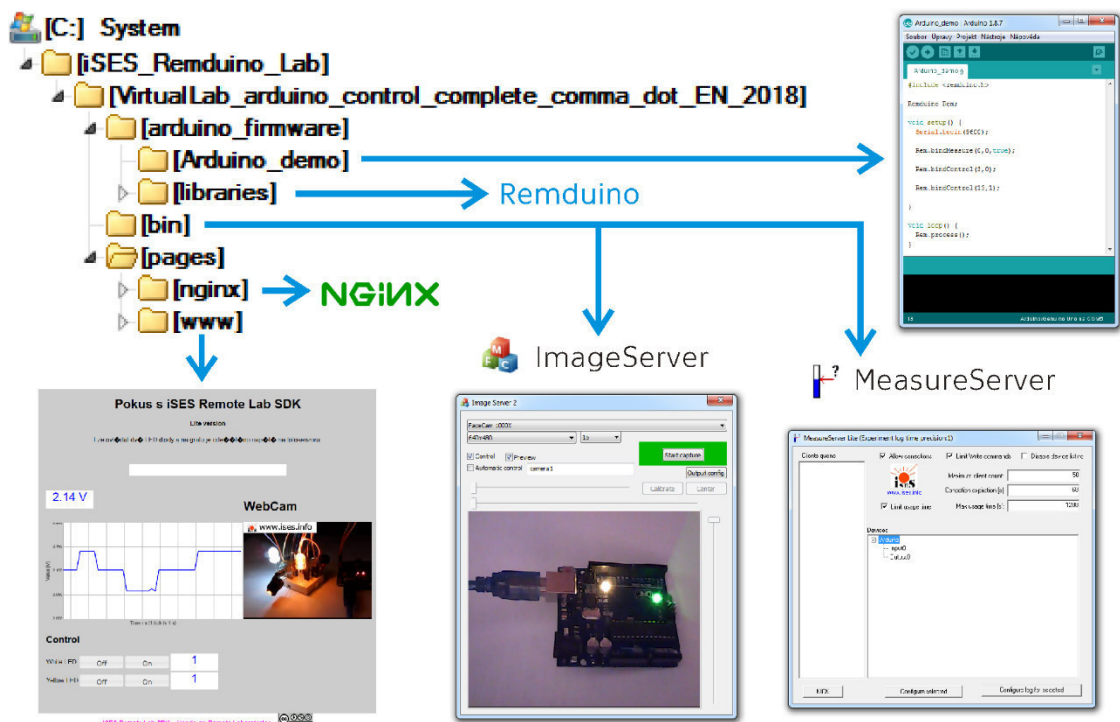
Obrázek 14 Stažení příkladů pro iSES Remduino Lab z webových stránek tvůrců [30]

Jednotlivé příklady obsahují knihovnu **Remduino**, skript pro Arduino, **image_server**, **MeasureServerLite**, **nginx** a webové skripty pro zobrazení webové stránky v prohlížeči. V druhém odkazu lze najít schéma zamýšleného experimentu. Experimenty využívají analogový pin A0 k měření a posílání výsledků serveru. K ovládání slouží digitální pin 3, v případě experimentu s řízením teploty i pin 13. V rámci zkušební verze je možné změnit čísla pinů, nebo měřit pouze logickou hodnotu na jednom digitálním pinu (viz kapitola 3.1.5).

V první řadě je potřeba sehnat součástky a zapojit obvod podle schématu. Pro pouhé vyzkoušení funkčnosti systému lze součástky nahradit jinými s podobnou funkcí, např. zaměnit relé za LED diodu. Jen je třeba učinit opatření pro správu funkce oné součástky. V případě záměny relé a LED diody je nutné LED diodu ošetřit odpovídajícím rezistorem, protože většina desek Arduino pracuje na 5V logice, která by většinu LED diod spolehlivě „odeslala do křemíkového nebe“.

Pokud je hardware připraven, může se přistoupit k připojení Arduina a webkamery k počítači, nahrání skriptu do Arduina a spuštění všech serverových aplikací. Veškerý potřebný software je k nalezení v ZIP souboru příkladu. Obsah ZIP souboru je potřeba zkopírovat do složky na disku. Pro správnou funkci všech součástí by názvy složek neměly obsahovat diakritiku. Složka **arduino_firmware** obsahuje ovládací skript a knihovnu **Remduino** (obrázek 15). Ve složce **bin** se nachází **MeasureServerLite** a **image_server**. Ve složce **pages** je vlastní webová stránka a server **nginx**.

Po nahrání skriptu do Arduina (viz kapitola 3.1.5) je vhodné zapnout server **nginx**. Aplikaci je nutné spustit pouze jednou, a to ve složce příkladu, který bude spuštěn. Pro vypnutí serveru se možné použít soubor **stop.bat** ve stejné složce.



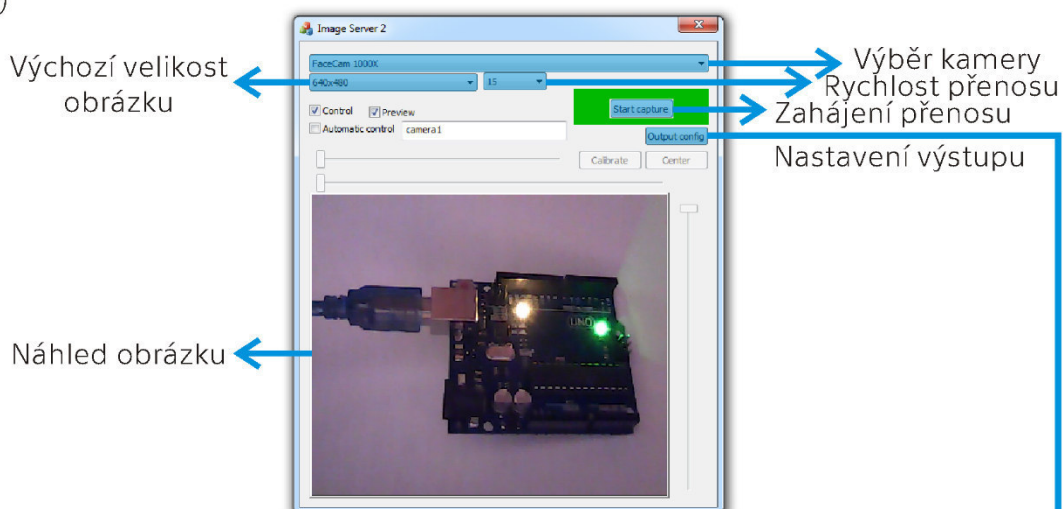
Obrázek 15 Umístění souborů příkladu iSES Remduino Lab

Image server

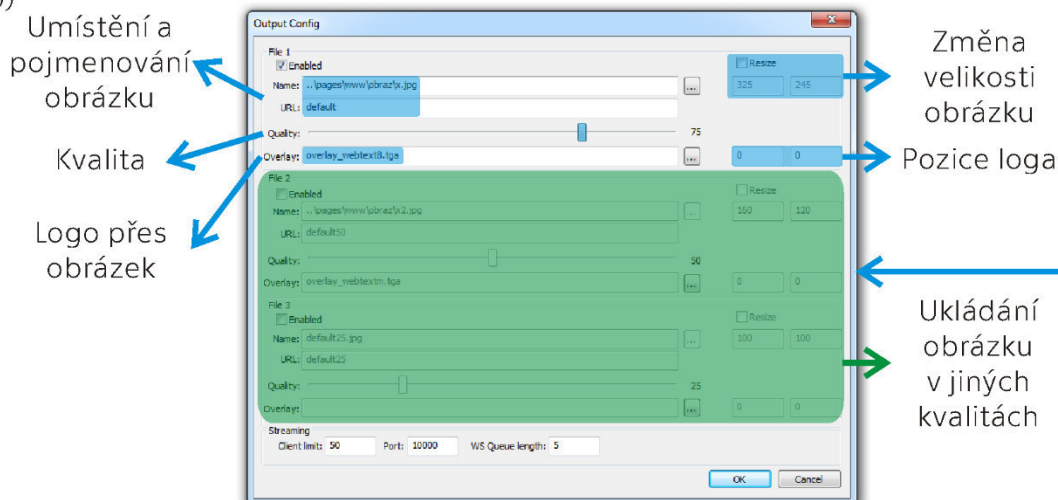
Aplikace **image_server** slouží k ukládání záznamu z USB kamery na disk. Před spuštěním je vhodné připojit požadovanou kameru, popř. nainstalovat ovladač kamery, je-li to vyžadováno. Při tvorbě bakalářské práce byla použita webová kamera Genius FaceCam 1000X, která uživatelskou instalaci ovladače nevyžadovala. Po spuštění aplikace je možné připojenou kameru nastavit jako zdroj. Vybraná kamera však nesmí být využívána jinou aplikací, neboť sériové propojení neumožňuje vícečetnou komunikaci. V úvodním dialogovém okně je dále možné měnit výchozí velikost obrázku nebo rychlost přenosu. Tlačítkem **Output config** se otevře podrobnější nastavení ukládaného obrázku. Je možné ukládat až 3 různě kvalitní obrázky. Každému je nutné nastavit cestu pro ukládání a název pro komunikaci s JavaScriptem. Dále je možné procentuálně měnit kvalitu nebo změnit velikost ukládaného snímku. Aplikace nabízí i vložení loga přes snímek. Pozici loga lze zadat souřadnicově, tzn. počtem pixelů. Zadané souřadnice označují levý horní roh vkládaného obrázku (obrázek 16).

Aby se zahájil přenos snímků na web, je nutné stisknout tlačítko **Start capture**.

a)



b)

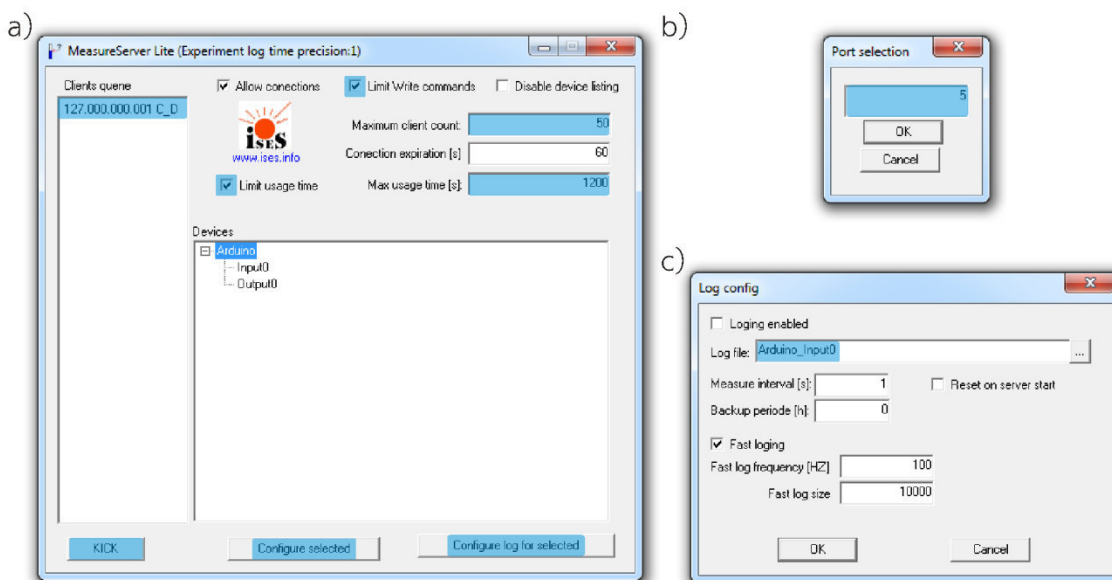


Obrázek 16 Nastavení image_serveru
a) Základní nastavení, b) Nastavení ukládaného obrázku

Measure server

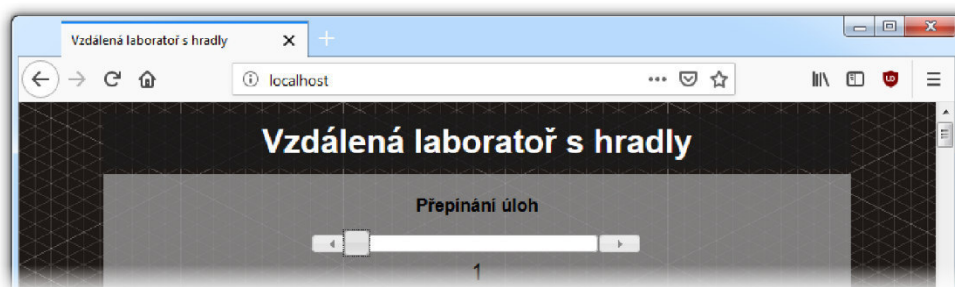
Pro nastavení **MeasureServeru** k vyzkoušení příkladu je nutné znát port, ke kterému je připojeno Arduino (viz kapitola 3.1.5). Po zapnutí aplikace je potřeba kliknout na nápis Arduino v okně **Devices** a poté stisknout tlačítko **Configure selected** (obrázek 17). Otevře se nové dialogové okno, kam lze napsat číslo portu. Není-li číslo portu správné, po chvíli se objeví chybové dialogové okno.

Pro vyzkoušení příkladu není třeba dalšího nastavování. Aplikace však skýtá další možnosti. V okně **Clients queue** je možné vidět zařízení (např. PC, smartphone atd.), která jsou k experimentu připojena. Tlačítkem **KICK** je možné vybrané zařízení od experimentu odpojit. Nabídka **Maximum client count** nabízí možnosti omezit množství uživatelů ve stejný čas. Zaškrtnutí políčka **Limit Write Comands** a **Limit usage time** dává tvůrci experimentu možnost předat ovládání pouze jedinému uživateli, přičemž **Max usage time** stanovuje, za kolik sekund se ovládání přepne na dalšího uživatele ve frontě. Pro pokročilé nastavení správy dat je možné použití tlačítka **Configure log for selected**. Tlačítko funguje jen s vybraným vstupem nebo výstupem.



Obrázek 17 Nastavení MeasureServeru
a) Úvodní dialogové okno, b) Nastavení portu, c) Nastavení správy dat

Po zapnutí všech částí je možné experiment zobrazit v prohlížeči po zadání adresy **localhost** nebo **http://127.0.0.1/** (obrázek 18).



Obrázek 18 Spuštění webového rozhraní pomocí adresy „localhost“

3.1.3 Základní funkce v iSES Remduino Lab

Knihovna **Remduino** poskytuje čtyři možnosti (funkce), jak komunikovat se serverem. Dvě funkce slouží k přijímání příkazů a dvě funkce slouží k odesílání naměřených dat. Pro přijímání příkazů souží funkce **bindControl** a **bindControlFunction**, zatímco pro odesílání dat slouží funkce **bindMeasure** a **sendValue** (obrázek 19).

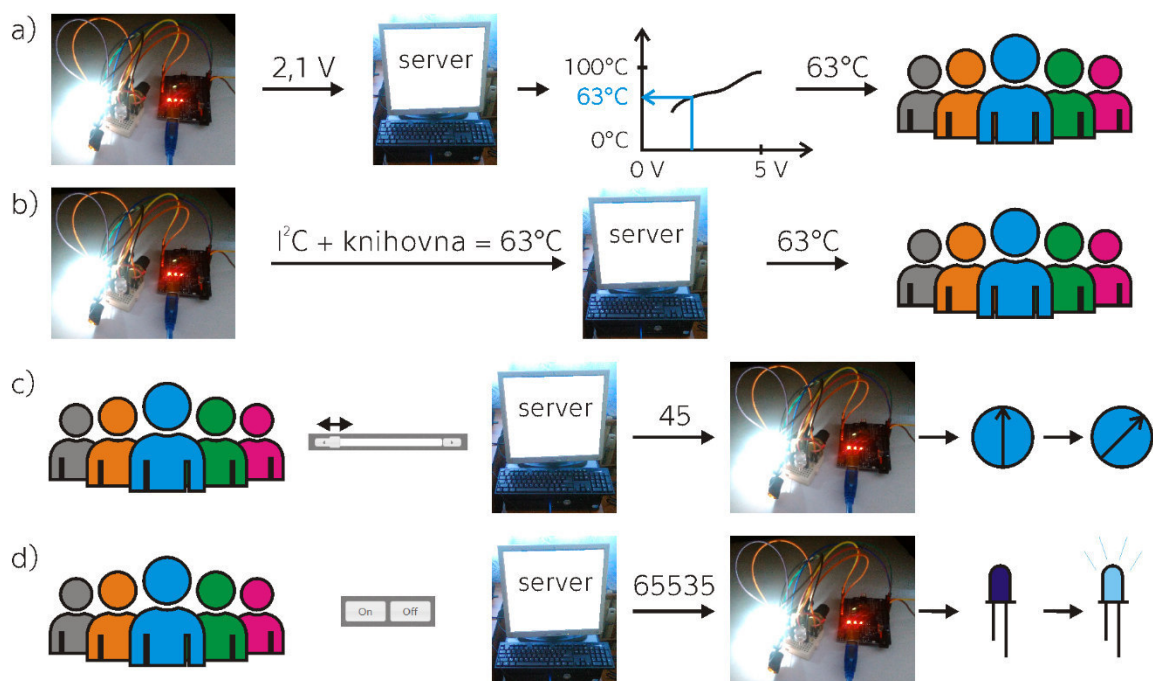
Funkci **bindControl** je možné použít k přepínání digitálního výstupního pinu, tzn., že lze měnit logickou 1 a 0. Funkce je tudíž vhodná k zapínání a vypínání vhodných součástek (relé, MOSFET, LED atd.)

Funkce **bindControlFunction** dokáže přijmout hodnotu v rozsahu 0 až $2^{16}-1$ a vložit ji do uživatelem definované funkce. Tuto funkci knihovna **Remduino**, resp. hlavní funkce knihovny **process**, automaticky v každé smyčce provádí, aniž by musel uživatel programovat cokoliv navíc.

Funkce **bindMeasure** slouží k odeslání hodnoty napětí na analogovém vstupním pinu nebo k odeslání logické hodnoty digitálního pinu. Zpracování této hodnoty provádí JavaScript na straně serveru. Zajímavé je, že 10-bitový převodník v Arduino dává hodnoty 0 a 1023, ale knihovna je převede opět na 16-bitovou informaci. To usnadňuje programování skriptu na straně serveru.

Existují čidla, která mají od výrobce dodanou knihovnu pro získání měřené veličiny rovnou v daných jednotkách, nebo čidla komunikující přes I²C sběrnici, která umožňují na dvou pinech odečítat větší množství čidel. V takovém případě je nutné vyhodnotit data v Arduino a odeslat jen výslednou hodnotu pomocí funkce **sendValue**.

Lze si všimnout, že funkce **bindControlFunction** a **sendValue** by zvládly obstarat veškeré operace, ale je třeba mít na paměti, že Arduino má jen omezenou kapacitu paměti a funkce **bindControl** a **bindMeasure** jsou důležitými nástroji knihovny.



Obrázek 19 Princip základních funkcí v iSES Remduino Lab
a) *bindMeasure*, b) *sendValue*, c) *bindControlFunction*, d) *bindControl*

3.1.4 Potřebné znalosti

Pro vytvoření vlastního experimentu pomocí iSES Remduino Lab je nutné znát několik programovacích jazyků, ovládat základy elektrotechniky, a především mít nápad na vhodný experiment. Naštěstí iSES Remduino Lab je připraven tak, aby byly požadavky na počáteční znalosti minimalizovány.

Knihovna **Remduino** je připravena tak, že v naprosté většině případů stačí vložit na správná místa ukázkové kódy funkcí, upravit čísla pinů a čísla vstupních nebo výstupních zařízení ke spárování s JavaScriptem a Arduino IDE skript je hotov. V případě využití modulů s knihovnamy je samozřejmě nutné připojit i ukázkový kód výrobce.

Pro propojení Arduina s moduly, popř. dalšími součástkami je vhodné ovládat základy elektrotechniky. Pokud však tvůrce využije tzv. **Arduino Friendly** moduly s ukázkovým zapojením, je možné sestavit experiment i s minimálními znalostmi.

Na straně serveru jsou využity tři programovací jazyky k zobrazení a ovládní laboratoře. Hlavním zobrazovacím jazykem je HTML, který zajišťuje zobrazování laboratoře ve webovém prohlížeči. Skripty kaskádových stylů (CSS) slouží k vylepšení vizuálních efektů. Nejedná se jen o design samotné webové stránky, ale také o vzhled ovládacích prvků (tlačítka, posuvníky atd.) a zobrazovacích komponent (grafy atd.). Skripty JavaScriptu slouží ke komunikaci s Arduinem. Pro tvůrce experimentu jsou připravené ukázkové kódy komponent a jedinou starostí je tedy správné označení odchozího nebo příchozího zařízení a přiřazení identifikačního čísla.

Tyto tři typy skriptů je možné zapsat do jediného souboru s příponou *.html nebo je možné zapsat každý do svého vlastního souboru s příponami *.html, *.css a *.js, přičemž jsou vzájemně propojeny odkazy. Pozor ale, že zápis do jediného souboru má jiný syntax než zápisy v dílčích souborech. Tvůrci iSESu vhodně zvolili zápis v dílčích souborech, neboť se jedná o velké množství kódu a tvůrce experimentu potřebuje upravit jen pár řádek kódu, které jsou v samostatných souborech.

Při programování je podstatné komentování kódu, aby se v něm i sám tvůrce po čase vyznal. Komentář je část kódu, která se neprovádí:

Komentáře v Arduino IDE skriptu, v JavaScriptu a v CSS skriptu:

```
// Toto je řádkový komentář.
```

Toto komentář není // a odtud je komentář až do konce řádku.

```
/* Toto komentář je*/ a toto už není.
```

```
Toto komentář není. /*
```

```
Toto komentář je.
```

```
Toto je také komentář.
```

```
*/ A toto už zase komentář není.
```

Komentáře v HTML skriptu:

```
<!-- Toto je komentář. -->
```

```
<!--
```

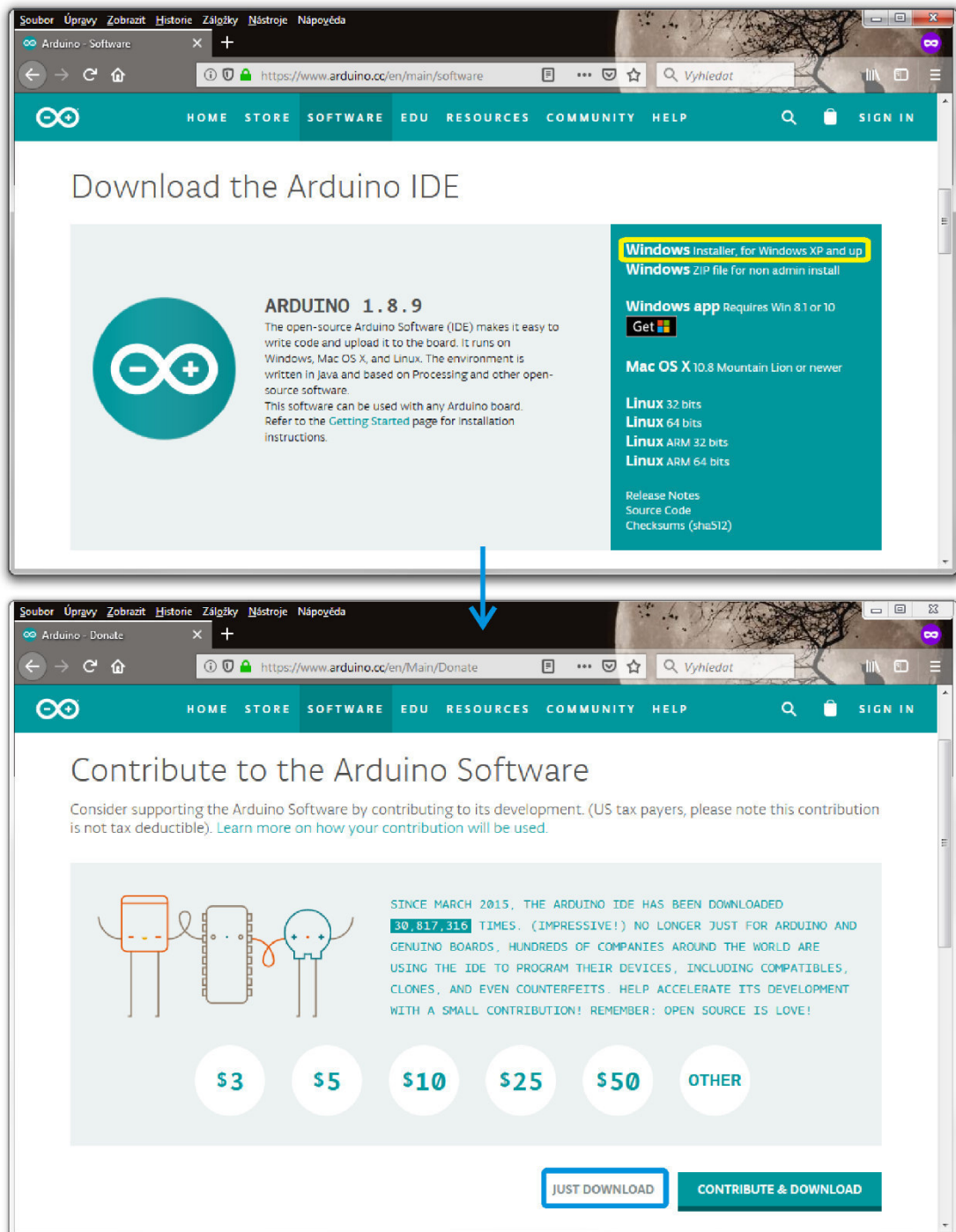
```
Toto je také komentář.
```

```
-->
```

Toto komentář není <!--toto komentář je --> a toto opět není.

3.1.5 Arduino IDE

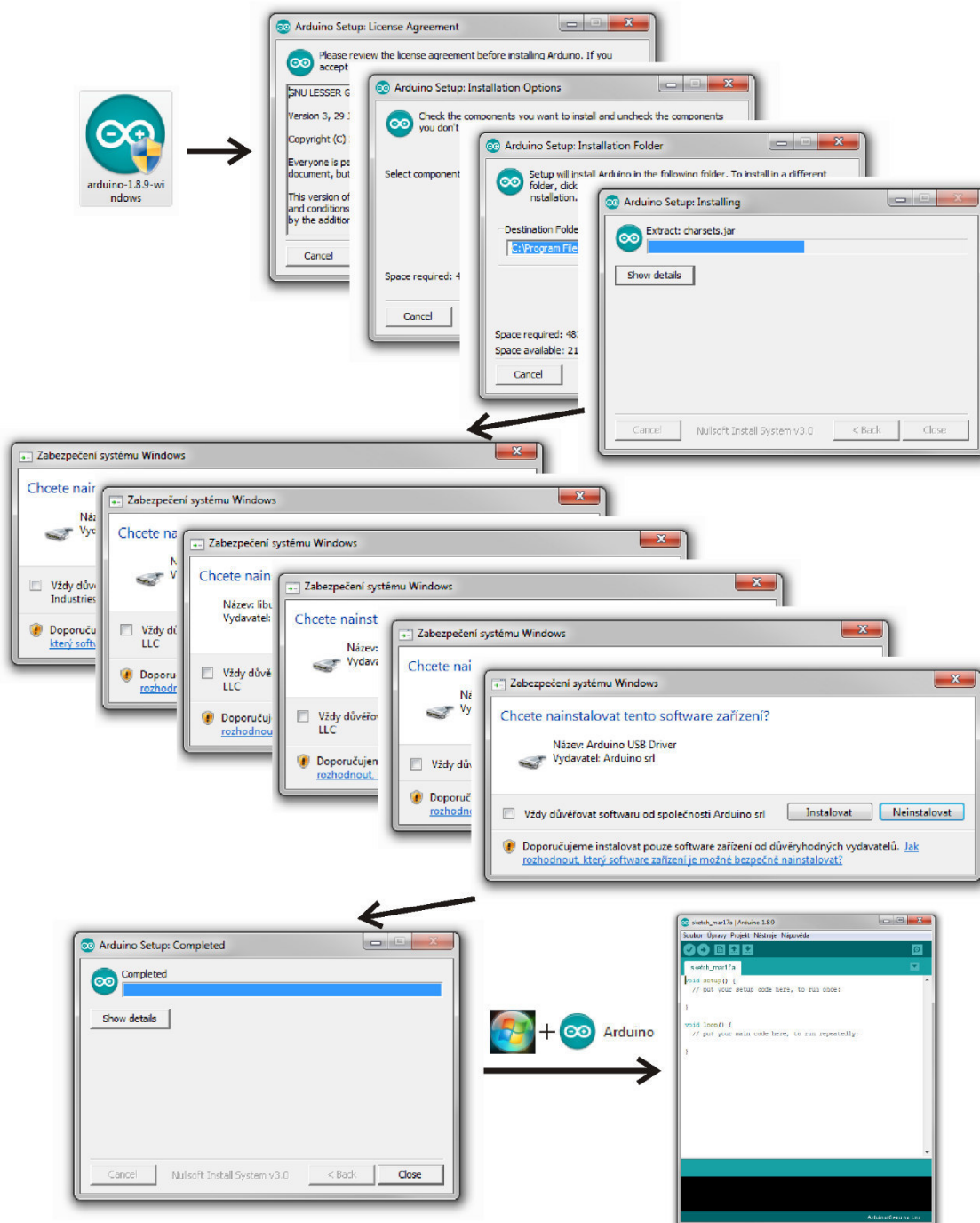
Pro tvorbu a nahrání skriptu do Arduina slouží program Arduino IDE. Tvůrci klonů originálních desek Arduino zpravidla využívají pro nahrání skriptu totožný program. Software je možné stáhnout z oficiálních stránek výrobce [31]. Výrobci nabízejí program pro různé operační systémy. Po vybrání typu programu se zobrazí obrazovka s výzvou k příspěvku na další rozvoj programu (obrázek 20). Zájemce, který neovládá angličtinu, by vzhled stránky mohl odradit. Pro bezplatné stažení stačí stisknout odkaz **JUST DOWNLOAD**. Přispět na vývoj a zlepšování si ovšem vývojáři zaslouží.



Obrázek 20 Stažení Arduino IDE z webových stránek vývojářů

Instalace Arduino IDE

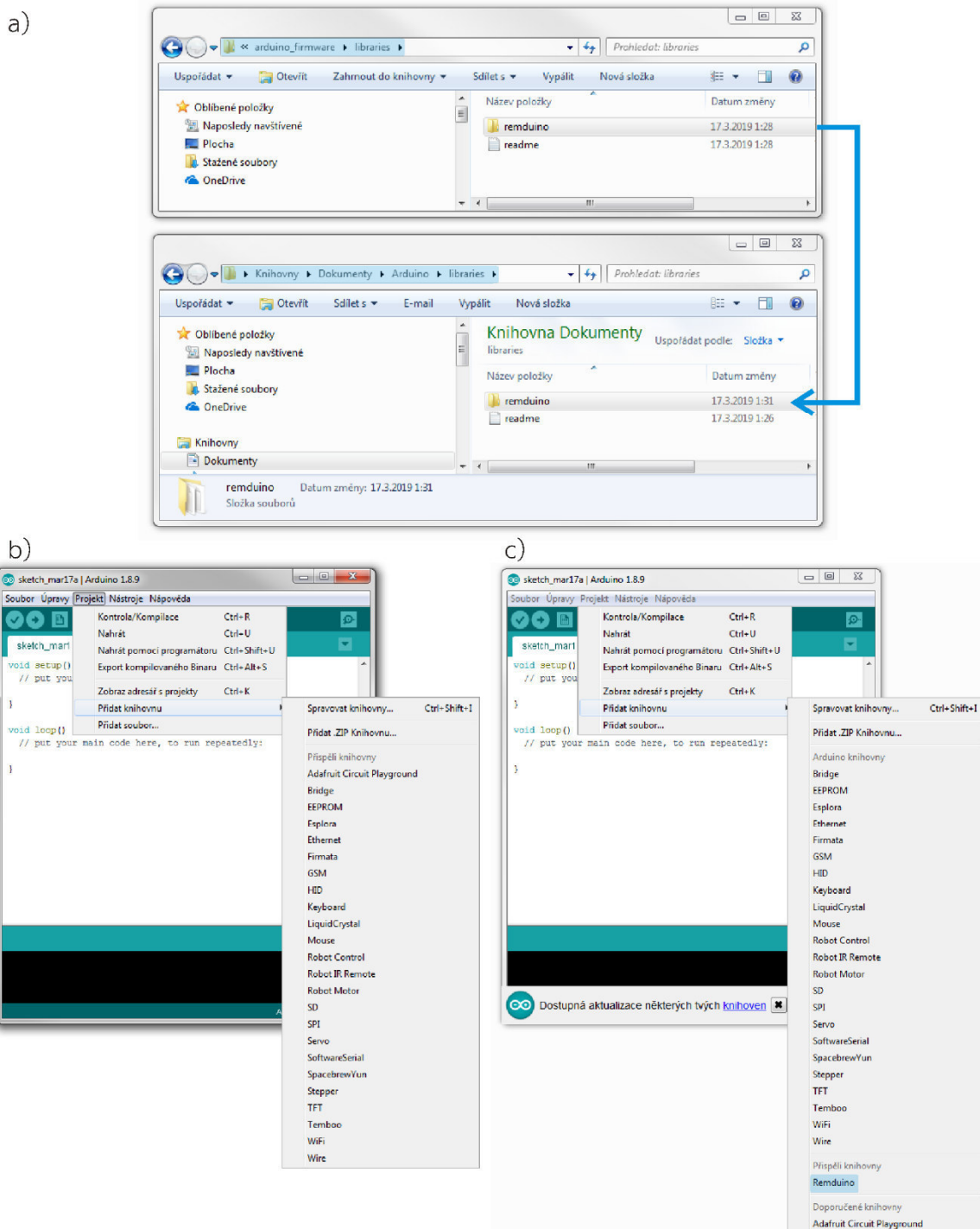
V rámci zpracování bakalářské práce byl využit počítač s operačním systémem MS Windows 7. Pro instalaci programu v tomto operačním systému stačí spustit instalační soubor, v úvodních třech dialogových oknech odsouhlasit licenční podmínky, vybrat všechny požadované součásti pro instalaci a nastavit umístění pro instalaci. Dalším krokem je instalace oficiálních ovladačů (obrázek 21). Těch je několik a je třeba je vždy odsouhlasit. Dialogová okna pro potvrzení instalace mají výchozí pozici, kde se otevřou (obvykle střed prvního monitoru). V případě počítače s více monitory je tedy třeba sledovat celou plochu. Po instalaci lze program okamžitě spustit.



Obrázek 21 Instalace programu Arduino IDE

Instalace knihovny Remduino

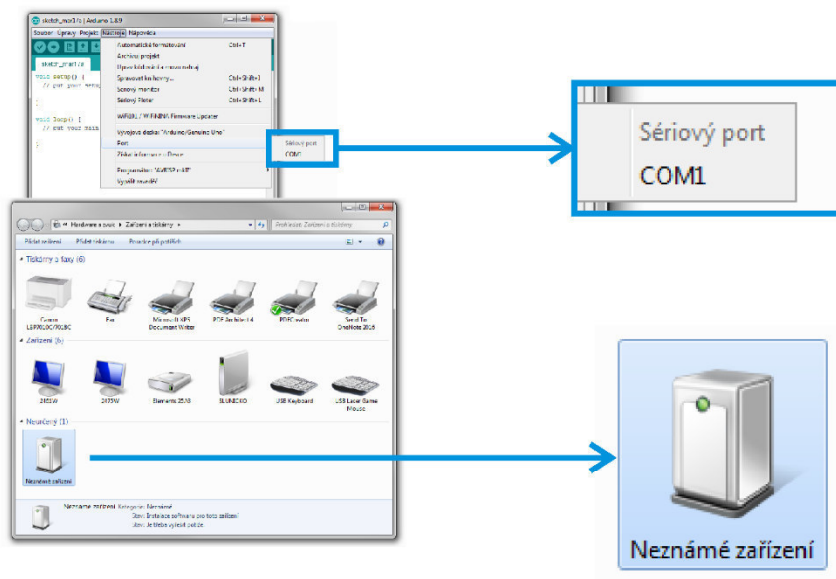
Knihovna Remduino není součástí instalace programu. Je nutné knihovnu stáhnout spolu s příkladem z webových stránek tvůrců iSES Remduino Lab [30]. V adresáři příkladu je složka **Arduino_firmware\libraries**, kterou je třeba zkopírovat do složky knihoven v dokumentech uživatele (obrázek 22). Složka knihoven Arduina se vytvoří až po prvním spuštění programu Arduino IDE. Po zkopírování knihovny nebude knihovna v otevřeném programu viditelná. Pro aktualizaci seznamu knihoven je nutný restart programu.



Instalace ovladačů neoriginálních desek

Kromě originálních desek existuje celá řada tzv. klonů. Klony mají hodně výhod. Hlavní výhodou je cena, která je v tuzemských obchodech často o 75 % nižší než cena originální desky. Kvalita provedení je přitom shodná s originální deskou. Klony ovšem nebývají podporovány oficiálními ovladači a počítač je po připojení přes USB nerozpozná. Při koupi klonu je třeba zjistit typ převodníku a najít a nainstalovat vhodný ovladač. Prodejci často stažení ovladače nabízí v popisku produktu (obrázek 23).

a)



b)

ARDUINO-SHOP.CZ

IOT Vývojové platformy > ARDUINO > Klon Arduino UNO R3 ATmega328P CH340 mini USB

Popis produktu

MCU: ATmega328
 Převodník USB: CH340
 Pracovní napětí: 5V
 Vstupní napětí: 7-12V
 Vstupní nap max.: 6-20V
 I/O Piny: 14 (6 použitelných jako PWM výstup)
 Abnalog. vstupy: 6
 DC Proud na pin: 40 mA
 Flash: 32 KB (ATmega328) 0.5 KB použito pro bootloade
 SRAM 2 KB (ATmega328)
 EEPROM 1 KB (ATmega328)
 Krystal: 16 MHz

Ovladače pro převodník USB CH340

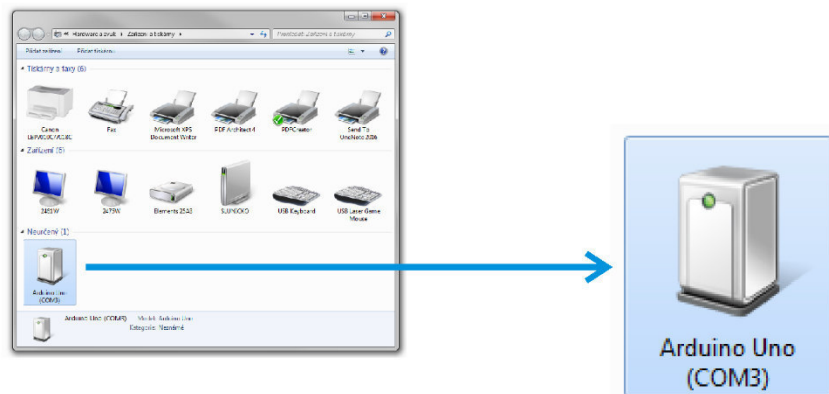
- Pro Windows (klikni pro stažení)
- Pro Linux (klikni pro stažení)
- Pro MAC (klikni pro stažení)

Obrázek 23 Instalace ovladačů pro klon Arduina
 a) Port COM1 neodpovídá „neznámému“ Arduinu, b) Potup instalace ovladačů

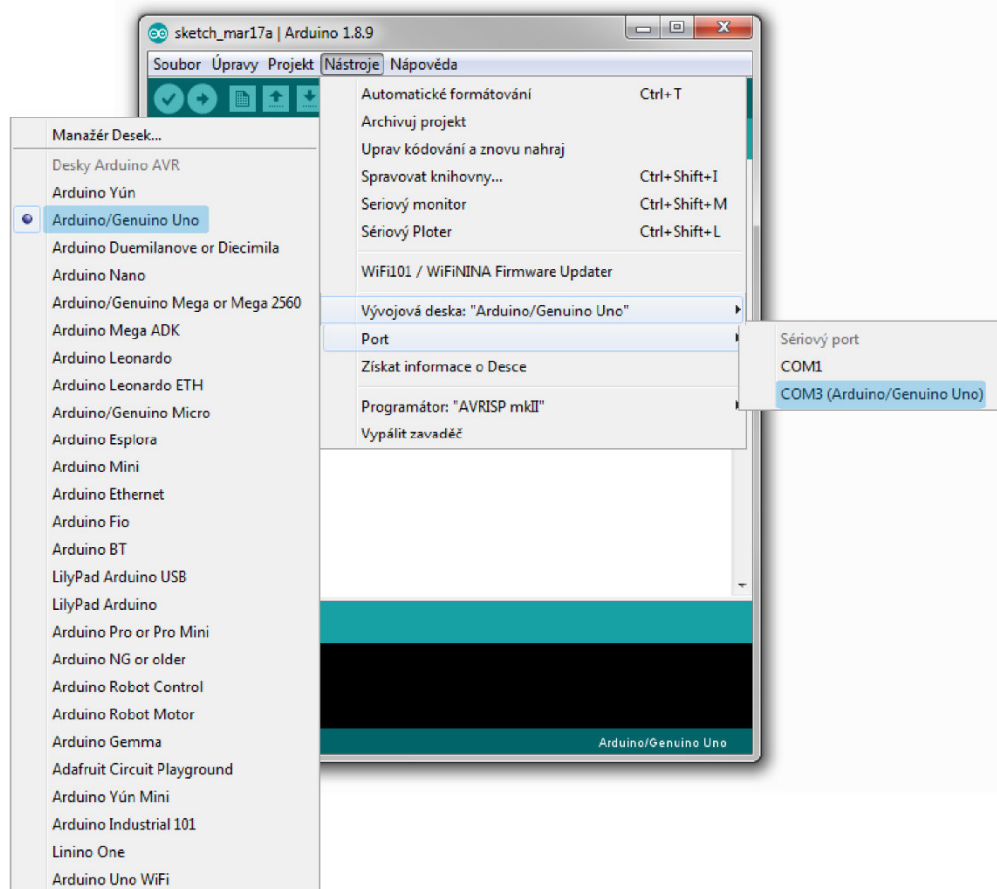
Po instalaci vhodného ovladače není obvykle potřeba Arduino odpojovat a znovu připojovat, aby se ovládacím panelu zařízení a tiskáren načel typ zařízení a přidělený port (obrázek 24). Pokud se zařízení nenačte, je vhodné vyzkoušet opětovné připojení Arduina, případně i restart celého počítače. Pokud nezabere ani to, je třeba najít jiné ovladače. To je oproti originálním deskám drobná nevýhoda.

Jakmile se Arduino hlásí počítači a je mu přidělen port, je možné tento port nastavit v programu Arduino IDE, které ani není nutné restartovat. Pro nahrání skriptu je rovněž nutné nastavit kromě správného portu i správný typ desky.

a)



b)



Obrázek 24 Nainstalovaný správný ovladač klonu Arduina
a) Arduino hlášené na port COM3, b) Port COM3 nabídce portů a výběr typu desky

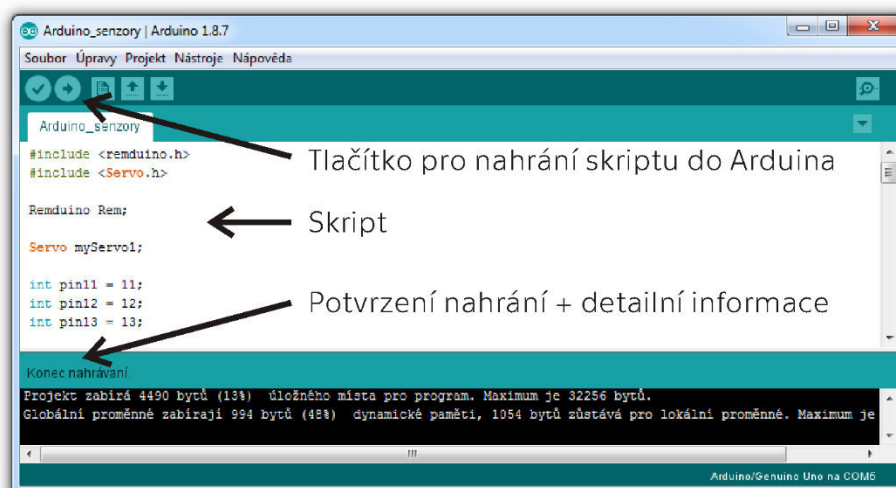
Nahrání skriptu do Arduina

Je-li Arduino IDE schopno rozeznat připojenou mikroprocesorovou desku, zbývá už jen otevřít soubor s příkladem a nahrát skript do Arduina (obrázek 25). Pro vyzkoušení funkčnosti je možné vyzkoušet některý z příkladů přímo v programu. Vhodný je příklad s blikáním vestavěné diody na pinu 13 Arduina Una. Příklad je možné najít v záložce **Soubor/Příklady/01.Basics/Blink**. Pokud vše funguje, jak má, skript se do desky nahraje a dioda začne blikat. Ve spodní části programu je možné získat detailní informace o skriptu, jako je např. rozsah využití kapacity paměti.

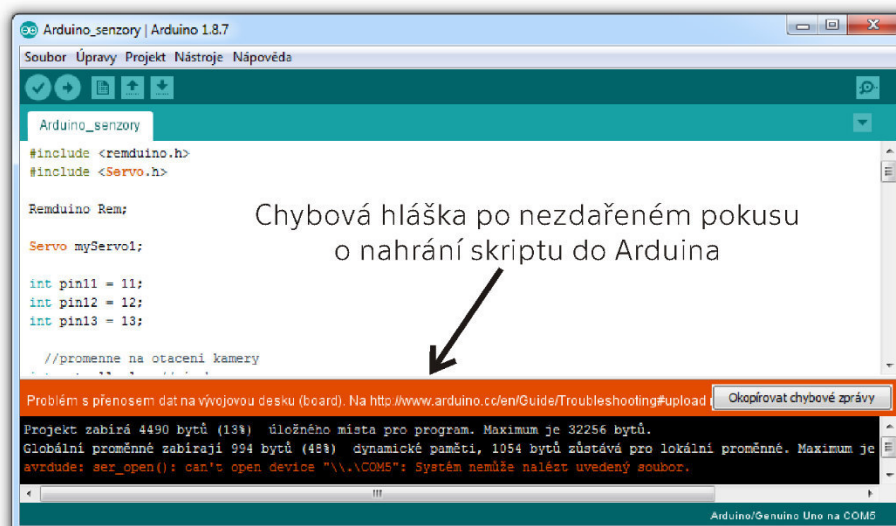
Pokud nastane chyba, je potřeba zkontrolovat nastavení portu a typu desky. V případě chyby ve skriptu program napoví očekávané místo chyby. Chybou je i chybějící knihovna.

Pokud vše funguje, je možné otevřít soubor **Arduino_demo** iSES Remduino Lab příkladu a vyzkoušet připravený experiment.

a)



b)



Obrázek 25 Nahrávání skriptu do Arduina

a) Úspěšné nahrání skriptu, b) Chybová hláška při neúspěšném pokusu o nahrání skriptu

Psaní skriptu s Remduinem

Nejprve je potřeba uvést, jak skript pro Arduino funguje. Základní struktura obsahuje pouze funkce **setup** a **loop**. První zmíněná je zaváděcí funkce, která se provede po spuštění Arduina. Funkce **loop** se pak po zbytek běhu skriptu ve smyčce opakuje, jak napovídá název.

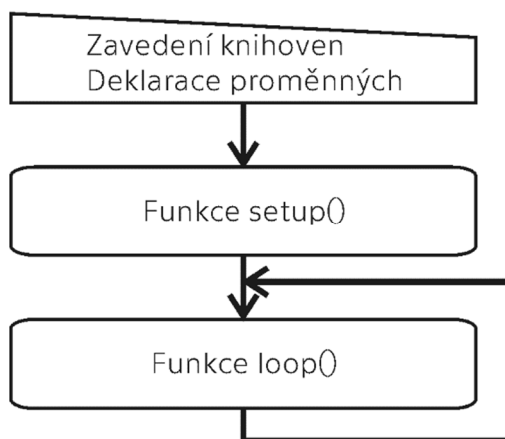
Je-li třeba zavést nějakou knihovnu, je nutné to provést ještě před funkcí **setup**. V této části je možné deklarovat i proměnné (obrázek 26). Pro zadání uživatelských funkcí, které chce programátor volat do jiných části kódu, je vyhrazena oblast pod funkcí **loop**. Zápis skriptu vypadá následovně:

```
// Místo pro zavedení knihoven a deklaraci proměnných

void setup() {
  // Zaváděcí funkce, která proběhne jen jednou, a to při spuštění Arduina.
}

void loop() {
  // Smyčka je funkce, která probíhá pořád dokola, pokud ovšem není pozastavena
  // např. přerušovací funkcí.
}

// Místo pro napsání uživatelských funkcí, které je možno v běhu skriptu volat.
```



Obrázek 26 Vývojový diagram skriptu pro Arduino

Pro použití funkcí z knihovny **Remduino** je potřeba na začátku skriptu knihovnu zavést. Funkce z knihovny se volají napsáním názvu knihovny, tečky a názvu funkce se závorkami obsahujícími parametry. Pro zkrácení kódu je možné změnit název knihovny potřebný pro volání funkcí. Skript dále musí obsahovat spuštění sériové komunikace, která se spouští v zaváděcí funkci **setup**. Poslední nezbytnou součástí kódu je vložení funkce **process** do hlavní smyčky skriptu.

Funkce **bindMeasure**, **bindControl** a **bindControlFunction** se vkládají do zaváděcí funkce a jejich činnost je řízena funkcí **process**. Pouze pro funkci **bindControlFunction** je nutné ve spodní části nadefinovat vlastní spouštěnou funkci.

Naproti tomu funkce **sendValue** vyžaduje vložení do hlavní smyčky společně s kódem aktualizujícím odesílanou hodnotu. Zápis jednotlivých funkcí do samostatných funkčních skriptu je ukázán v následujících kódech.

Skript pro měření analogového pinu:

```
#include <remduino.h>      // Zavedení knihovny Remduino

Remduino Rem;             // Zkrácení názvu pro snazší psaní kódu

void setup() {            // Zavaděcí funkce
  Serial.begin(9600);     // Spuštění sériové komunikace

  // Měření analogového vstupního pinu 0 přesměrovaného
  // na vstupní zařízení 1 (Arduino_Input1)
  Rem.bindMeasure(0,1,true);
}

void loop() {             // Hlavní funkce Remduina
  Rem.process();
}
```

Skript pro měření digitálního pinu:

```
#include <remduino.h>      // Zavedení knihovny Remduino

Remduino Rem;             // Zkrácení názvu pro snazší psaní kódu

void setup() {            // Zavaděcí funkce
  Serial.begin(9600);     // Spuštění sériové komunikace

  // Měření digitálního vstupního pinu 2 přesměrovaného
  // na vstupní zařízení 0 (Arduino_Input0)
  Rem.bindMeasure(2,0,false);
}

void loop() {             // Hlavní funkce Remduina
  Rem.process();
}
```

Skript pro odeslání hodnoty:

```
#include <remduino.h>      // Zavedení knihovny Remduino

Remduino Rem;             // Zkrácení názvu pro snazší psaní kódu

unsigned int odesilanaHodnota; //deklarace proměnné

void setup() {            // Zavaděcí funkce
  Serial.begin(9600);     // Spuštění sériové komunikace
}

void loop() {             // Hlavní funkce Remduina

  // Zde je možnost aktualizovat „odesilanouHodnotu“

  //Odeslání hodnoty vstupnímu zařízení 3 (Arduino_Input3)
  Rem.sendValue(3, odesilanaHodnota);
}
```

Skript pro ovládání digitálního pinu:

```
#include <remduino.h>      // Zavedení knihovny Remduino

Remduino Rem;             // Zkrácení názvu pro snazší psaní kódu

void setup() {            // Zaváděcí funkce
  Serial.begin(9600);     // Spuštění sériové komunikace

  // Ovládání digitálního pinu 3 pomocí informace
  // od výstupního zařízení 0 (Arduino_Output0)
  Rem.bindControl(3,0);
}

void loop() {             // Hlavní funkce Remduina
  Rem.process();
}
```

Skript pro příjem informace a spuštění funkce:

```
#include <remduino.h>      // Zavedení knihovny Remduino

Remduino Rem;             // Zkrácení názvu pro snazší psaní kódu

void setup() {            // Zaváděcí funkce
  Serial.begin(9600);     // Spuštění sériové komunikace

  // Příjem informace od výstupního zařízení 1 (Arduino_Output1)
  // a spuštění funkce „spouštenaFunkce“
  Rem.bindControlFunction(1,spouštenaFunkce);
}

void loop() {             // Hlavní funkce Remduina
  Rem.process();
}

// Zápis uživatelské funkce pro hodnotu od zařízení 1
void spouštenaFunkce (unsigned int prijataHodnota){
  // Tělo funkce
}
```

Jak lze vidět z předchozích příkladů kódů, práce s knihovnou **Remduino** je snadná a přehledná. Stačí nastavit číslo měřeného nebo ovládaného pinu, číslo vstupního nebo výstupního zařízení ke spárování s JavaScriptem a příp. vlastní funkce pro plné využití všech možností knihovny.

Při návrhu experimentu je nutné počítat s tím, že komunikace Arduina se serverem probíhá přes USB port. Sériová komunikace u Arduina není umožněna jen přes USB port, ale i přes piny označené **TX** a **RX**. Pro Arduino Uno se jedná o digitální piny **0** a **1**. Tyto piny jsou spojené s datovými piny na USB portu, proto není možné je využít při návrhu, resp. realizaci vzdáleného experimentu.

3.1.6 Webové skripty

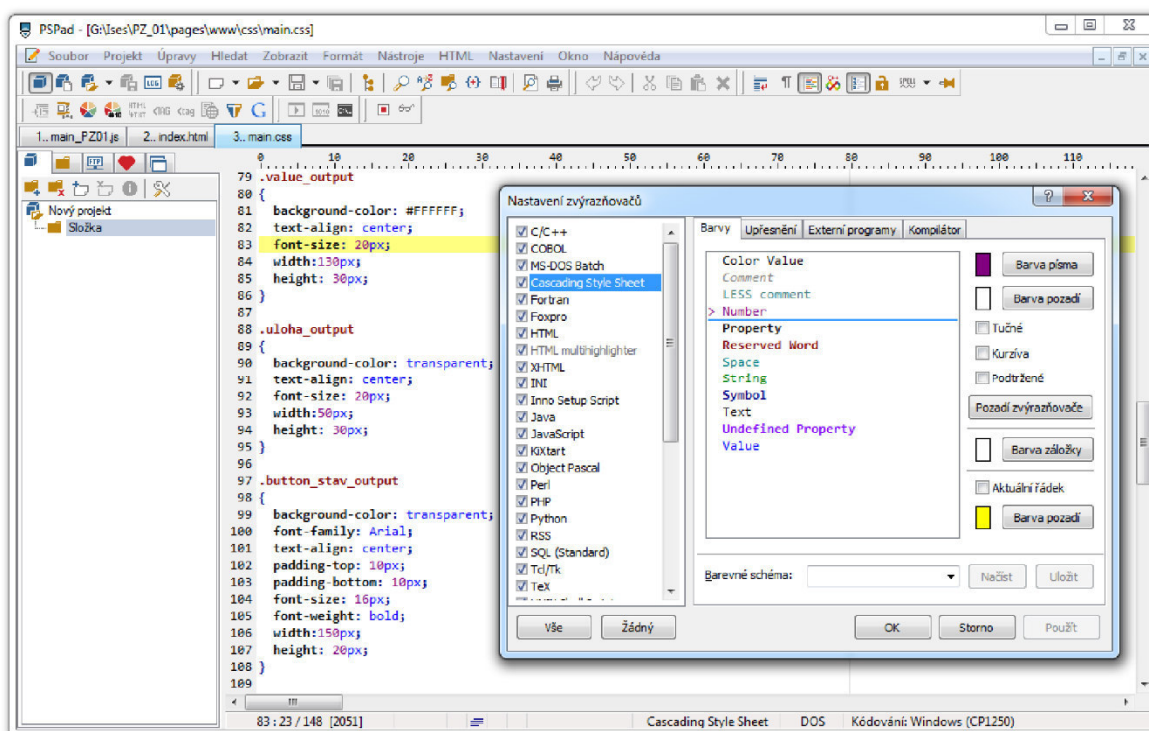
Autoři iSES Remduino Lab vytvořili webové skripty tak, aby zájemce o vytvoření vzdálené laboratoře nemusel nic instalovat do počítače. Přestože by se k naprogramování webových stránek vzdálené laboratoře zdála jako lepší varianta využití programovacího jazyka Python, zvolená kombinace programovacích jazyků vyžaduje pouze jakýkoliv textový editor. Je však vhodné využít editor zvýrazňující syntaxi. Lze doporučit PSPad od českého programátora Jana Fialy [32]. Tento editor je distribuován jako freeware, je v češtině a umožňuje uživatelské nastavení zvýrazňovačů (obrázek 27).

Základním způsobem tvorby webové stránky je napsání HTML struktury do souboru s příponou *.html. HTML struktura je následně prohlížečem přeložena do vizuální podoby. Možnosti HTML jsou ale značně omezené. Proto v průběhu let vznikly různé způsoby, jak tyto možnosti rozšířit.

Jedním způsobem pro tvorbu dynamických webových stránek je využití jazyka PHP. Ten je ovšem založen na skládání HTML struktury na straně serveru na základě požadavků uživatele. To by znamenalo neustálé aktualizování webové stránky, které se ovšem neshoduje s principem fungování serverových aplikací iSESu.

Vhodným způsobem je využití JavaScriptu, který slouží k vytváření aktivních prvků na webových stránkách. Nevýhodou je, že uživatelé často ve svých prohlížečích provádění JavaScriptu zakazují. Proto je vhodné před spuštěním vzdáleného experimentu uživatele upozornit na nutnost povolení tohoto jazyka.

Pro vizuální efekty autoři iSES Remduino Lab využili kaskádové styly, které však není potřeba využívat v plném rozsahu, ale pouze pro aktivní prvky.



Obrázek 27 Nastavení zvýrazňovačů v programu PSPad

JavaScript pro komunikaci s Arduinem

JavaScriptový soubor pro nastavení komunikace mezi serverem a Arduinem je dobré ukládat do složky „js“, ale obecně může být umístěn kdekoli v adresářích webu, pokud se na něj HTML skript odkazuje. Při nastavování komunikace je třeba vycházet z JavaScriptů, které jsou ve složce „virtuallab\js“. V nich jsou naprogramované funkční skripty a aktivní komponenty. V rámci bakalářské práce bylo prozkoumáno jen několik základních komponent, ale možnosti tohoto nástroje jsou daleko větší. Naneštěstí manuál k funkcím iSES Remduino Lab nebyl v době vypracování bakalářské práce k dispozici, komentáře v kódu jsou poměrně stručné a psané v anglickém jazyce. Kód navíc zabírá několik tisíců řádků, přičemž se funkce odkazují napříč přibližně dvou desítek samostatných souborů.

Základní struktura JavaScriptového kódu pro nastavení připojení uživatelů a komunikace s **MeasureServerem** a **image_serverem** je následující:

```
$( document ).ready(function() { // Začátek hlavní funkce

// Připojení k Measure serveru
VL.Connect("ws://" + window.location.host + ":80/ws/measure");

$("#camera").VLCamera({ // Nastavení kamery

// Odkaz na využívaný záznam image_serveru (viz obrázek 16b)
  address: window.location.host + ":80/ws/image/default",
  dropCount: 0,
  width: 640,           // Šířka obrázku
  height: 480          // Výška obrázku
});

$("#connection").VLConnection(); // Nastavení připojení uživatelů

// Zde je prostor pro další komponenty

}); // Konec hlavní funkce
```

Pro ovládání je potřeba nastavit další komponenty. Skripty jejich funkcí je možné najít v souboru „virtuallab\js\virtuallab\virtuallab_components.js“. Pro jednoduché experimenty si uživatel vystačí s ovládacími prvky, jako jsou tlačítka nebo posuvníky, a se zobrazováním naměřených hodnot v číselné podobě nebo pomocí grafu.

Komponentám je potřeba přiřazovat název zařízení k synchronizaci se skriptem nahraným do Arduina. V základní podobě jsou všechna výstupní zařízení pojmenována jako **Arduino_OutputX** a vstupní jako **Arduino_InputY**, kde **X** a **Y** za názvem je nutné nahradit pořadovým číslem zařízení. Na zařízení je pohlíženo ze strany serveru, proto výstupní zařízení slouží k odesílání příkazů Arduinu a vstupní k přijímání hodnot.

Pro snazší orientaci v kódu je vhodné definovat proměnné, které mají příhodnější název a odkazují na určité zařízení nebo např. hodnotu maximálního rozsahu:

```
// Pojmenování výstupního zařízení 0 jako „prepinaci_slider“
var prepinaci_slider = "Arduino_Output0";

var button_control_off = 0;           // Hodnota pro logickou 0 na pinu
var button_control_on = 65534;       // Hodnota pro logickou 1 na pinu
```

Vlastní pojmenování zařízení má výhodu i při potřebě změnit číslo zařízení. Potom stačí přejmenovat pouze uživatelsky definovanou proměnnou a všechny komponenty využívající tuto proměnnou tím jsou předefinovány bez potřeby je dohledávat.

Pro zapínání a vypínání digitálních pinů je v kombinaci s funkcí **bindControl** v Arduinu nožné využít tlačítko:

```
// Funkce v Arduinu
// Ovládání digitálního pinu 3 pomocí informace
// od výstupního zařízení 0 (Arduino_Output0)
Rem.bindControl(3,0);

// JavaScriptová komponenta .VLButton
var zapinaniCerpada = "Arduino_Output0"; // Pojmenování zařízení
var button_off = 0; // Hodnota pro logickou 0 na pinu
var button_on = 65534; // Hodnota pro logickou 1 na pinu

// Nastavení tlačítka spouštěcího pin 3, který ovládá chod čerpadla
$("#Cerpadlo_on").VLButton({ // Tlačítko má identifikátor „Cerpadlo_on“
  devices: [
    { deviceName : zapinaniCerpada, value: button_on }
    // Následující zápis je shodný:
    // { deviceName : "Arduino_Output0", value: 65534 }
  ],
  description : "On" // Nápis na tlačítku
});

// Nastavení tlačítka vypínacího pin 3, který ovládá chod čerpadla
$("#Cerpadlo_off").VLButton({ // Tlačítko má identifikátor „Cerpadlo_off“
  devices: [
    { deviceName : zapinaniCerpada, value: button_off }
    // Následující zápis je shodný:
    // { deviceName : "Arduino_Output0", value: 0 }
  ],
  description : "Off" // Nápis na tlačítku
});
```

Tlačítkem lze posílat jakoukoliv hodnotu v rozsahu 0 až $2^{16}-1$. V kombinaci s funkcí **bindControlFunction** je tedy možné např. nastavit úhel natočení servo motoru:

```
// Funkce v Arduinu
// Příjem informace od výstupního zařízení 1 (Arduino_Output1)
// a spouštění funkce „spoustenaFunkce“
Rem.bindControlFunction(1,spoustenaFunkce);

// Funkce knihovny Servo.h, která by byla v těle spousteneFunkce
myservol.write(input); // input je hodnota „value“ vyslaná tlačítkem

// JavaScriptová komponenta .VLButton
var servoMotor = "Arduino_Output1"; // Pojmenování zařízení

$("#servoMotor_30").VLButton({ // Tlačítko má identifikátor „servoMotor_30“
  devices: [
    { deviceName : servoMotor, value: 30 }
  ],
  description : "30°" // Nápis na tlačítku
});
```

Pro plynulé nastavení hodnoty je vhodné použít posuvník v kombinaci s funkcí **bind-
ControlFunction**:

```
// Pojmenování výstupního zařízení 1 jako „prepinaci_slider“
var prepinaci_slider = "Arduino_Output1";
var prepinaci_slider_min = 0; // Minimální hodnota
var prepinaci_slider_max = 65534; // Maximální hodnota

$("#prepinaci_slider").VLSlider({ // Posuvník má identifikátor „prepinaci_slider“
  devices : [
    { deviceName : prepinaci_slider },
    { deviceName : prepinaci_slider }
  ],
  refresh : 100, // Obnovení v ms
  min: prepinaci_slider_min, // Minimální hodnota
  max: prepinaci_slider_max, // Maximální hodnota
  select_step:1, // Velikost kroku při posunu jezdce
  button_step:10 // Velikost kroku při stisku tlačítka s šipkou
});
```

Pro zobrazení hodnoty jako číslo je možné využít komponentu **VLValueDisplay**. Tato komponenta zobrazí hodnotu jak vstupního, tak výstupního zařízení. U výstupního zařízení je nutné dávat pozor, protože hodnota se ve webovém prohlížeči zobrazí rychleji, než je možné pozorovat odezvu vzdáleného experimentu. Pokud je tedy potřeba měřit závislost jedné veličiny na druhé, je vhodnější obě hodnoty číst z Arduina.

```
// Zobrazení hodnoty vstupního zařízení
$("#hodnota0").VLValueDisplay({ // Komponenta má identifikátor „hodnota0“
  devices : [
    // Čtení probíhá ze vstupního zařízení 0
    // Hodnoty sMin a sMax je maximální rozsah přijatých hodnot
    // Hodnoty dMin a dMax je rozsah na který se přichází hodnota mapuje
    { deviceName : "Arduino_Input0", sMin: 0, sMax: 65534,
      dMin: 0, dMax: 5}
  ],
  refresh : 200, // Obnovení v ms
  format: "0.00 V" // Formát zobrazování
  // Přijde-li tedy hodnota 32767, skript zobrazí: 2,50 V
});

// Zobrazení hodnoty výstupního zařízení
$("#slider_hodnota").VLValueDisplay({ //Komponenta má id=„slider_hodnota“
  devices : [
    // Čtení probíhá z výstupního zařízení 1
    // Hodnoty sMin a sMax je maximální rozsah přijatých hodnot
    // Hodnoty dMin a dMax je rozsah na který se přichází hodnota mapuje
    { deviceName : prepinaci_slider, refresh: 100,
      sMin: prepinaci_slider_min, sMax: prepinaci_slider_max,
      dMin: 1, dMax: 5 }
  ],
  refresh : 200, // Obnovení v ms
  format: "0" // Formát zobrazování
  // Přijde-li tedy hodnota 32767, skript zobrazí: 3
});
```


Hodnoty je někdy vhodnější zobrazovat do grafu. Připravené skripty umožňují zobrazování veličiny v závislosti na čase, nebo jako XY graf, závislý na dvou měřených veličinách. Právě u XY grafu je potřeba dát pozor na využívání odesílané hodnoty k vykreslování grafu, protože např. k posunu pojezdu dojde až nějaký čas po odeslání požadavku. Tudíž by se ke zvolené hodnotě postupně zobrazovaly všechny hodnoty zaznamenané v průběhu posunu.

XY graf vyžaduje vyplnění většího množství parametrů, včetně **chartSeries** nebo **postProc**. Proto je v následujícím kódu ukázán a popsán pouze graf pro zobrazování veličiny v závislosti na čase:

```
$( "#casovy_graf" ).VLCart({ //Komponenta má id=„casovy_graf“
    devices : [

        // Čtení probíhá ze vstupních zařízení 0 a 1
        // Hodnoty sMin a sMax je maximální rozsah přijatých hodnot
        // Hodnoty dMin a dMax je rozsah na který se přichází hodnota mapuje
        // Time limit udává po kolika vteřinách od vzniku se mají data mazat
        { deviceName : "Arduino_Input0" , loadMethod : "Fast",
          sMin: 0, sMax: 65534, dMin: 0, dMax: 5, timeLimit: 5},
        { deviceName : "Arduino_Input1" , loadMethod : "Fast",
          sMin: 0, sMax: 65534, dMin: 0, dMax: 5, timeLimit: 10},
    ],

    loadOnCreate : true,
    chartParams : { // Vlastnosti grafu

        // Barvy vykreslovaných křivek v pořadí zadaných zařízení
        colors: ["#0000ff", "#00ff00"],
        yaxis : { // Parametry osy y
            max: 5.1, // Maximální hodnota na ose y
            min: 0, // Minimální hodnota na ose y
            color: "#000000", // Barva zobrazených hodnot osy y
            title: 'Napeti [V]' // Popis osy y (pouze bez diakritiky)
        },
        xaxis : { // Parametry osy x
            showLabels:false, // (ne)Zobrazení hodnot u svislých čar
            color: "#000000", // Barva zobrazených hodnot osy x
            noTicks: 10, // Hustota svislých čar
            title: 'Cas / s // Popis osy x },

        grid : { // Parametry mřížky
            tickColor: "#BOB0B0", // Barva mřížky
            backgroundColor: "#ffffff" // Barva pozadí
        },
        shadowSize : 0, // Velikost stínu křivky

        // HtmlText definuje, jestli je popis grafu jako kopírovatelný text
        HtmlText: false // Text je převeden do obrázku
    },

    autoScroll : true, // Povolení automatického posunu
    /// Rychlost posunu grafu -používá se jen jedna vlastnost
    autoScrollWindowX : 10 // Zpomalení rychlosti ve směru x
    //autoScrollSpeedX : 10 // Urychlení rychlosti ve směru x
});
```

Kaskádové styly (CSS) pro aktivní prvky

Použití kaskádových stylů, zkráceně též CSS (Cascading Style Sheets), je vhodné především pro formátování vizuálního vzhledu webových stránek (obrázek 28). Největší výhoda kaskádových stylů se projevuje především u rozsáhlých webových stránek, kde lze pomocí jednoho souboru měnit vzhled u všech částí. Případně je vhodné kaskádové styly používat při propojení webových stránek s JavaScriptem pro formátování aktivních prvků. CSS lze zapisovat i přímo do HTML kódu, ale tím se ztrácí jejich výhoda formátování celého webu naráz. Vzhledem k tomu, že CSS není původní součástí HTML, tak se požadované vlastnosti zapisují jinak.

Pomocí CSS lze u webových stránek vytvořit moderní vzhled, který pouhým použitím HTML není často dosažitelný. Nevýhodou může být podpora internetových prohlížečů, které některé funkce nemusí zobrazovat správně.

Při použití externího souboru CSS lze odkazy na konkrétní formáty v HTML provádět buďto pomocí identifikátoru (id), nebo tzv. třídy (class). Je nutno dát pozor na rozdíl při deklarování vlastností objektu v CSS, protože se liší označení identifikátoru a třídy. Při současném použití JavaScriptu je dobré používat v CSS pro aktivní objekty především třídy, aby nedošlo ke kolizi syntaxe.

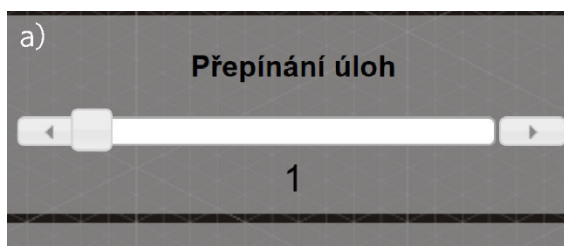
Formátování, na které bude v HTML kódu odkazováno pomocí identifikátoru:

```
#odkazy_lab a:link { // Vizuální styl odkazu, který ještě nebyl použit
    color: black; // Barva textu - černá
    text-decoration: none; // Bez podtržení textu
}
```

Formátování, na které bude v HTML kódu odkazováno pomocí třídy:

```
.uloha_output { // Číslo úlohy zobrazované pod posuvníkem
    background-color: transparent; // Pozadí prvku je bez barvy
    text-align: center; // Zarovnání textu na střed
    font-size: 20px; // Velikost textu 20 px
    width: 50px; // Maximální šířka prvku 50 px
    height: 30px; // Maximální výška prvku 30 px
}

.uloha_output2 { // Jiný styl čísla úlohy zobrazované pod posuvníkem
    background-color: white; // Pozadí prvku je bílé
    text-align: center; // Zarovnání textu na střed
    font-size: 30px; // Velikost textu 30 px
    width: 50px; // Maximální šířka prvku 50 px
    height: 40px; // Maximální výška prvku 40 px
    font-family: Courier; // Font písma je Courier
    font-weight: bold; // Text je tučný
}
```



Obrázek 28 Ukázka užití kaskádových stylů na zobrazení čísla úlohy
a) Formátování pomocí „uloha_output“, b) Formátování pomocí „uloha_output2“

Doporučená struktura HTML kódu při využití iSES Remduino Lab

K vytvoření HTML struktury stačí znát základy tvorby HTML skriptů. Základy jsou dobře popsány i na mnohých českých webech, jako je např. www.jakpsatweb.cz. K vytvoření webové stránky je zapotřebí párového tagu `<html></html>`, ve kterém je hlavička `<head></head>` a tělo stránky `<body></body>`. Do hlavičky se vkládají informace, které jsou potřebné k fungování stránky, ale nejsou většinou zobrazovány. Výjimku tvoří titulek `<title></title>`, který se zobrazuje na záložce otevřené stránky v prohlížeči. V hlavičce stránky pro iSES Remduino Lab musí být odkazy na externí kaskádové styly a externí JavaScripty. Pokud je vyžadováno zobrazování diakritiky, je také nutné nastavit znakovou sadu na `CHARSET=WINDOWS-1250`.

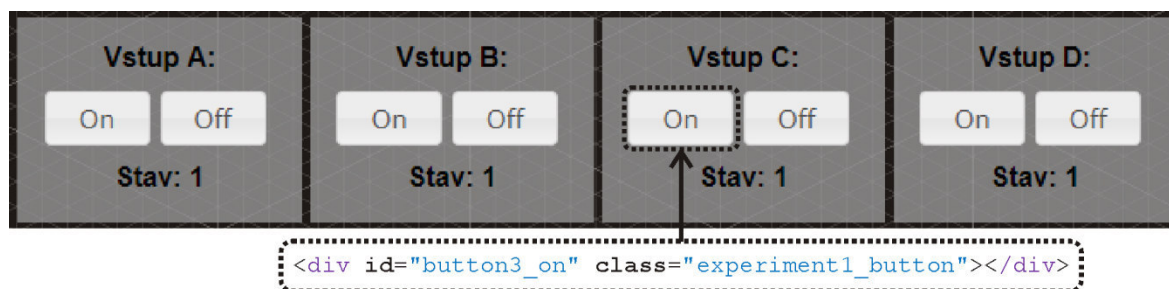
Do těla HTML struktury se vkládá kód, ze kterého webový prohlížeč pozná, jak má webová stránka vypadat. Vzhledem k tomu, že je potřeba načíst větší množství externích souborů, dá se doporučit vytvořit vizuální podobu především ze základních HTML prvků, jako je např. tabulka, a vzhled doladit interními kaskádovými styly. Aktivní prvky se pak do HTML zavádí pomocí blokového elementu `<div></div>` (obrázek 29). Je však nutné pro aktivní prvek zajistit skripty i ve všech ostatních jazycích:

```
// Zavedení funkce bindControl v Arduino IDE skriptu.
// Ovládání digitálního pinu 3 pomocí informace
// od výstupního zařízení 0 (Arduino_Output0).
Rem.bindControl(3,0);

//Nastavení odeslání hodnoty po stisknutí tlačítka pomocí externího JavaScriptu.
//Do Arduina dorazí informace přiřadit výstupnímu zařízení 0 (pin 3) logickou 1.
$("#button3_on").VLButton({
  devices: [
    { deviceName : "Arduino_Output0", value: 65534 }
  ],
  description : "On" // Nápis na tlačítku
});

//Nastavení vizuálního vzhledu tlačítka pomocí externích kaskádových stylů (CSS)
.experiment1_button{
  background-color: #FFFFFF; // Pozadí prvku je bílé
  text-align: center; // Zarovnání textu na střed
  font-size: 15px; // Velikost textu 15 px
  width: 60px; // Maximální šířka prvku 60 px
  height: 30px; // Maximální výška prvku 30 px
}

<!-- Zobrazení tlačítka pomocí HTML skriptu -->
<div id="button3_on" class="experiment1_button"></div>
```



Obrázek 29 Vizuální podoba spouštěcího tlačítka

3.2 Experiment s hradly

V rámci praktické části bakalářské práce byla zrealizována vzdálená laboratoř pro experiment s hradly. Hradla, nazývaná též logické členy, jsou základními prvky logických obvodů. Přestože se již nejedná o nejnovější „hi-tech“ technologii dnešní doby, uplatnění dodnes nacházejí. Ve výuce je pochopení funkce logických členů důležité pro rozvoj kreativního myšlení studenta. Logické členy totiž nemusí být využity jen jako fyzické součástky, ale jejich využití se objevuje i v programování softwaru.

Laboratorní cvičení je rozvrženo do pěti úloh, které je možné vypracovat pomocí vzdálené laboratoře. Na závěr je vytvořena šestá úloha, která nevyžaduje přístup k experimentu, ale vyžaduje zamyšlení nad problematikou. Přestože je experiment založen pouze na doplňování pravdivostních tabulek logických obvodů, je navržen tak, aby se obtížnost postupně zvyšovala. Ovládání experimentu navíc není nijak složité, neboť každá úloha disponuje maximálně čtyřmi vstupy, kterým uživatel nastavuje logickou 1 nebo 0. Přepínání mezi úlohami probíhá pomocí posuvníku.

V prvních třech úlohách jsou zkoumány základní druhy logických členů, které se vyrábí jako integrované obvody. Jedná se o invertor, logický součet, negovaný logický součet, logický součin, negovaný logický součin a exkluzivní logický součet. V rámci třetí úlohy je vyžadováno doplnění pravdivostních tabulek opakovače a negovaného exkluzivního logického součtu, které se běžně nevyrábí, ale je možné je sestavit z ostatních hradel. Již ve třetí úloze si tedy student může vyzkoušet, zda pochopil funkci logických členů.

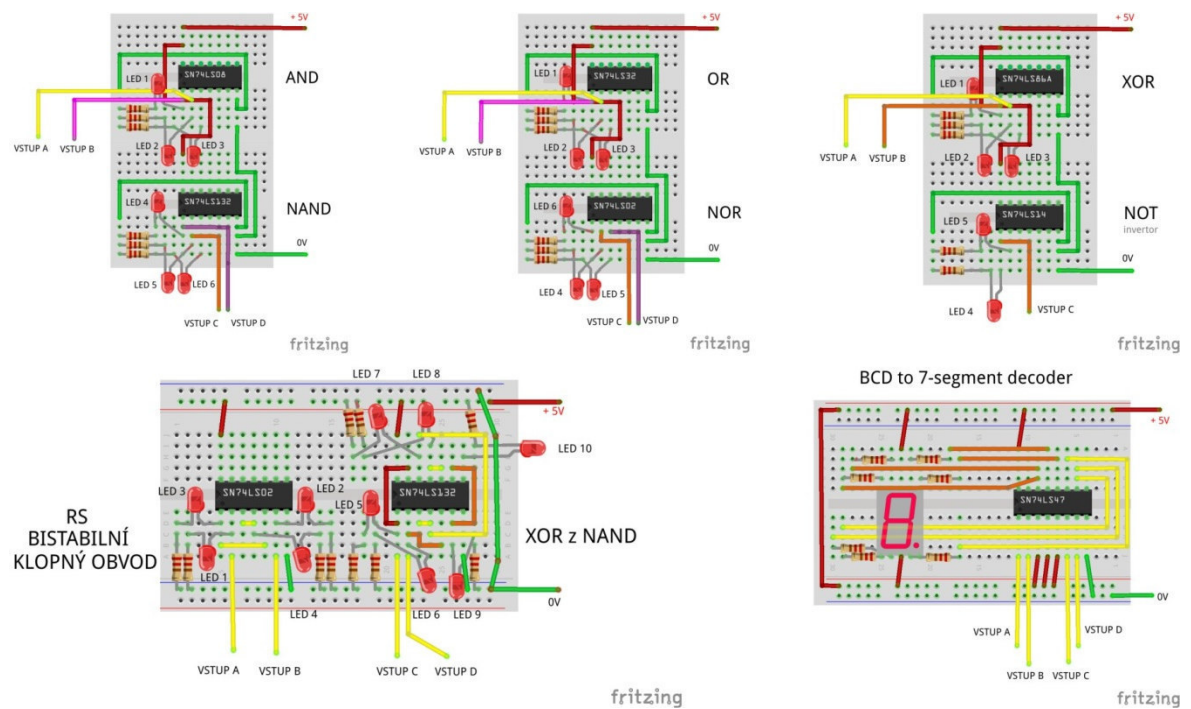
Čtvrtá úloha je rozdělena na dvě části. V první části je řešena pravdivostní tabulka RS bistabilního klopného obvodu tvořeného logickými členy NOR. V této úloze je zajímavé chování obvodu při nastavení logické 0 na obou vstupech, neboť si obvod zachová předchozí nastavení. Ve druhé části jsou propojeny čtyři hradla typu NAND, čímž vytvářejí funkci exkluzivního logického součtu. Student má za úkol kromě vypracování pravdivostní tabulky všech částí obvodu určit vytvořenou funkci XOR.

V páté úloze je využit BCD dekodér pro rozsvícení čísel na sedmisedimentovém LED panelu. V protokolu k laboratornímu cvičení (viz příloha) je kromě schématu vnitřního zapojení, které odhaluje celou soustavu logických členů, připravena pravdivostní tabulka pro všech 16 čtyřbitových vět. Pro usnadnění je pod tabulkou připraveno 16 prázdných schémat LED panelu, do kterých si student může zakreslit rozsvícené kombinace LED diod a pravdivostní tabulku může doplnit bez nutnosti dalšího pozorování experimentu. V páté úloze je student upozorněn, že LED diody na panelu mají společnou anodu, tudíž oproti předchozím úlohám rozsvícená dioda neznamena logickou 1, ale logickou 0.

V poslední úloze jsou tři příklady. V prvním příkladu má student vymyslet, jak vytvořit všechny typy logických členů s použitím jen čtyř součástek, přičemž každá součástka musí obsahovat jiný typ hradla. V druhém příkladu má student sestavit všechny typy logických členů z minimálního množství typů hradel. A nakonec se má student zamyslet, co se stane s bistabilním klopným obvodem, přepnou-li se naráz oba vstupy z logické 1 na logickou 0.

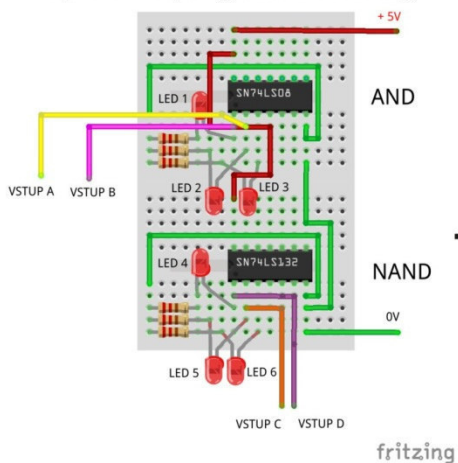
3.2.1 Použitá zařízení a součástky

Při návrhu vzdálené laboratoře pomocí Arduina a systému iSES Remduino Lab je dobré využít program Fritzing [33]. V tomto programu je možné poskládat součástky do nepájivých polí a propojit vodiči (obrázek 30). Program nabízí velké množství součástek a modulů určených k propojení s Arduinem. Součástky, které program neobsahuje, lze často dohledat na webových fórech, neboť je program značně oblíbený a komunita kolem něho je velmi aktivní. Pro pokročilejší uživatele program nabízí i další funkce, jako je příprava plošného spoje vymodelovaného zapojení. V případě přípravy vzdáleného experimentu byl program využit pro určení počtu součástek a pro přípravu vedení vodičů po nepájivém poli, protože v celém experimentu nebyl použit jediný pájený spoj (obrázek 31).

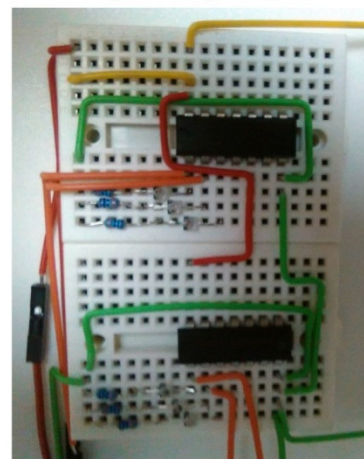


Obrázek 30 Schémata všech úloh vytvořená v softwaru Fritzing

Příprava v programu Fritzing



Reálně vyrobený experiment



Obrázek 31 Příprava úloh experimentu od návrhu v softwaru pro reálné sestavení

Server

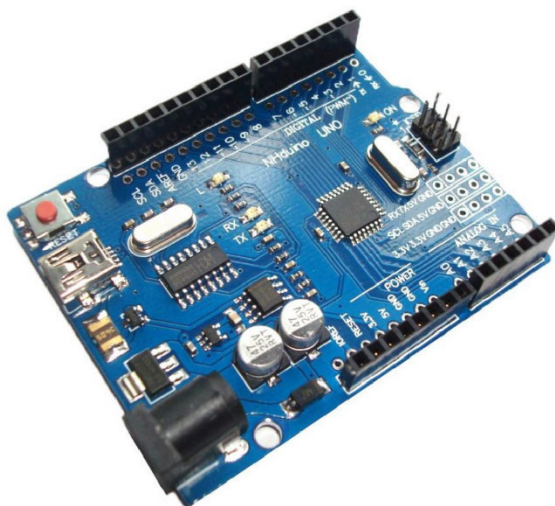
Systém iSES Remduino Lab je navržen tak, že komunikaci s Arduinem obsluhuje počítač, který současně funguje jako server. Pokud se nepředpokládá, že bude vzdálená laboratoř navštěvovaná velkým počtem uživatelů v jednu chvíli, stačí pořídit nejlevnější repasovaný počítač (obrázek 32). Operační systém počítače by měl být MS Windows, protože serverové aplikace jsou právě pro něj zkompileovány. Z hlediska možnosti zamezit počítači v nechtěných aktivitách, jako je restart počítače po neplánované aktualizaci, lze doporučit Windows 7, kde lze aktualizace minimalizovat.



Obrázek 32 Repasovaný počítač využitý jako server

Arduino Uno

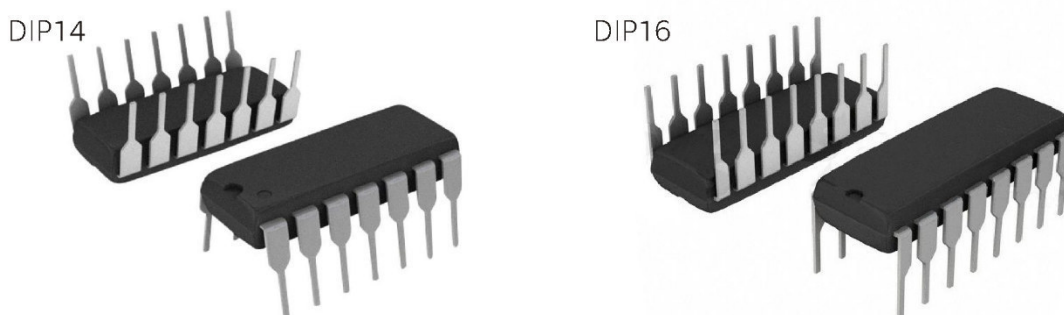
Experiment vyžadoval čtyři digitální piny na ovládání vstupů hradel a jeden digitální pin umožňující PWM modulaci pro ovládání servo motoru. Těmto požadavkům vyhovuje valná většina Arduin. Nakonec byl pro experiment využit plně kompatibilní klon Arduina Una, z důvodu integrovaného USB portu (obrázek 33).



Obrázek 33 Klon Arduina Una [34]

Hradla a BCD dekodér

Celý experiment se zaměřuje na logické členy. Ty se vyrábějí v integrovaných obvodech po několika kusech. Hradla AND, NAND, OR, NOR, XOR a NOT bylo možné zakoupit v pouzdře DIP14 a BCD dekodér byl v pouzdře DIP16. Označení pouzdra znamená integrovaný obvod se 14, resp. 16 nožičkami (obrázek 34). Pořízená hradla byla založena na TTL logice.



Obrázek 34 Hradla a BCD dekodér v provedení DIP14 a DIP16 [35]

LED diody a sedmissegmentový panel

Vizuální znázornění logické 1 nebo 0 je provedeno rozsvěcením a zhasínáním LED diod. Pro úlohy se základními hradly byly použity červené LED diody. Funkci BCD dekodéru znázorňuje sedmissegmentový LED panel se společnou anodou (obrázek 35).



Obrázek 35 LED dioda a sedmissegmentový LED panel [35]

Rezistory

Arduino Uno i použitá hradla fungují na 5V logice. Zatímco použité LED diody mají maximální přípustnou úroveň napětí 2,0 V - 2,5 V. Nejsnazším řešením ochrany LED diod před zničením je sériové vřazení rezistoru s odporem 330 Ω (obrázek 36).



Obrázek 36 Rezistor [35]

Kamera

Kamera pro přenos záznamu na webovou stránku pomocí nástrojů iSES Remduino Labu může být jakákoliv USB webová kamera, která má ovladač pro operační systém serveru. Při sestrojování vzdálené laboratoře byla zvolena nejlevnější varianta, kterou byla webová kamera Genius FaceCam 1000X (obrázek 37).



Obrázek 37 Webová kamera Genius FaceCam 1000X

Servo motor

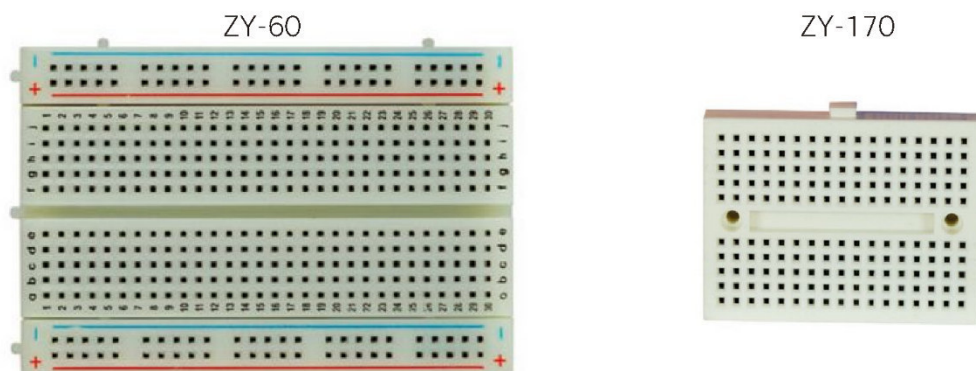
Vzdálená laboratoř má pět úloh. Kamera nemá natolik kvalitní záznam, aby se všechny úlohy vešly do jednoho záběru a byly pořád použitelné. Proto došlo k rozhodnutí, že přepnutím posuvníku se kamera natočí na zvolenou úlohu. K tomu byl zapotřebí motor (obrázek 38). Servo motor má oproti krokovému motoru výhodu, že na ovládání stačí jeden digitální pin podporující pulzně šířkovou modulaci, zatímco krokový motor vyžaduje driver ovládaný obvykle čtyřmi piny a pro stanovení výchozího úhlu je potřeba další minimálně jeden.



Obrázek 38 Servo motor [34]

Nepájivá pole a propojovací prvky

Celá vzdálená laboratoř je propojená pomocí nepájivých polí a konektorů. Nepájivá pole nejsou vhodná pro velké zatížení, ale ta v případě navrženého experimentu nastávají. Bylo též vyzkoušeno, že nepájivá pole neohrozila funkci logických členů. Kromě nepájivých polí typu ZY-60 a ZY-170 byly využity propojovací vodiče DuPont a propojovací drátky Pro'sKit (obrázek 39).



Propojovací vodiče DuPont



Propojovací drátky Pro'sKit



Obrázek 39 Propojovací materiál [35]

3.2.2 Skripty

Rozdělením experimentu na pět úloh došlo k výraznému snížení počtu ovládacích pinů. Každá úloha si vystačí s maximálně čtyřmi výstupními zařízeními a přepínání si v kombinaci s funkcí **bindControlFunction** vyžádalo další jedno výstupní zařízení. Celkově tedy bylo využito pouze pět výstupních zařízení.

Vstupní zařízení nebylo použito ani jedno, jelikož vyplnění protokolu je možné na základě vizuální odezvy LED diod. Když už je vyráběna vzdálená laboratoř, je vhodné využít její potenciál a nezobrazovat takto jednoduše zobrazitelnou skutečnost jen číslem, o kterém může student pochybovat, zda není jen výtvozem webového skriptu.

Na následujících řádcích je nastíněn skript v JavaScriptu. Vzhledem ke stále se opakujícímu téměř stejnému kódu pro všechna tlačítka je zobrazený kód zkrácen na nejpodstatnější části. Zato skript pro Arduino je vypsán v plném rozsahu.

JavaScript

```
$("#prepinaci_slider").VLSlider({
  devices : [          // přiřazení výstupního zařízení 0
    { deviceName : "Arduino_Output0" },
    { deviceName : "Arduino_Output0" }
  ],
  refresh : 100,      // Obnovení v ms
  min: 1,             // Minimální hodnota
  max: 5,             // Maximální hodnota
  select_step:1,     // Velikost kroku při posunu jezdce
  button_step:1      // Velikost kroku při stisku tlačítka s šipkou
});

$("#prepinaci_slider_value").VLValueDisplay({
  devices : [          // přiřazení výstupního zařízení 0
    { deviceName : "Arduino_Output0", refresh: 100,
      sMin: 1, sMax: 5, dMin: 1, dMax: 5 } // Rozsah zůstane stejný
  ],
  refresh : 200,      // Obnovení v ms
  format: "0"         // Formát zobrazování
});

$("#button1_on").VLButton({
  devices: [          // přiřazení výstupního zařízení 1 a hodnoty 65534
    { deviceName : "Arduino_Output1", value: 65534 }
  ],
  description : "On" // Popis tlačítka
});

$("#button1_value").VLValueDisplay({
  devices : [          // přiřazení výstupního zařízení 0
    { deviceName : "Arduino_Output1", refresh: 100,
      sMin: 0, sMax: 65534, dMin: 0, dMax: 1 } // Z rozsahu 0-65534 na 0-1
  ],
  refresh : 200,      // Obnovení v ms
  format: "Stav: 0 " // Formát zobrazování
});
```

Arduino IDE skript

```
#include <remduino.h> // Zavedení knihovny Remduino
#include <Servo.h> // Zavedení knihovny pro ovládání servo motoru

Remduino Rem; // Zkrácení názvu knihovny

Servo myServo1; // Pojmenování servo motoru

int actualAngle=90; // Proměnná: Skutečné natočení kamery
int newAngle; // Proměnná: Požadované natočení kamery
int servoSpeed=2; // Proměnná: Rychlost natáčení kamery

void setup() { // Zaváděcí funkce
  Serial.begin(9600); // Zahájení sériové komunikace

  myServo1.attach(9); // Přiřazení pinu 9 k servo motoru

  // Výstupní zařízení 0 spíná funkci "prepinaniUloh"
  Rem.bindControlFunction(0,prepinaniUloh);
  Rem.bindControl(10,1); // Výstupní zařízení 1 na pinu 10
  Rem.bindControl(2,2); // Výstupní zařízení 2 na pinu 2
  Rem.bindControl(3,3); // Výstupní zařízení 3 na pinu 3
  Rem.bindControl(4,4); // Výstupní zařízení 4 na pinu 4
}

void loop() { // Hlavní smyčka
  Rem.process(); // Hlavní funkce knihovny Remduino

  // Funkce zpomalující otáčení servo motoru
  if (newAngle!=actualAngle) { // Když je požadován jiný úhel
    if (newAngle<actualAngle) { // Když je nový uhel menší než stávající
      actualAngle=actualAngle-servoSpeed; // zmenšení skutečného úhlu
      myServo1.write(actualAngle); // Otočení na upravený úhel
    }
    else{ // Když je nový uhel větší než stávající
      actualAngle=actualAngle+servoSpeed; // zvětšení skutečného úhlu
      myServo1.write(actualAngle); // Otočení na upravený úhel
    }
  }
}

void prepinaniUloh(unsigned int input){ // Přijatá hodnota je proměnná input
  if (input==1) { // Když se přijatá hodnota rovná 1
    newAngle=154; // nastav požadovaný úhel na 154.
  }
  else if (input==2) { // Když se přijatá hodnota rovná 2
    newAngle=124; // nastav požadovaný úhel na 124.
  }
  else if (input==3) { // Když se přijatá hodnota rovná 3
    newAngle=86; // nastav požadovaný úhel na 86.
  }
  else if (input==4) { // Když se přijatá hodnota rovná 4
    newAngle=54; // nastav požadovaný úhel na 54.
  }
  else { // Když se přijatá hodnota nerovná předchozím
    newAngle=22; // nastav požadovaný úhel na 22.
  }
}
```

Struktura HTML

Struktura HTML stránky pro vytvoření uživatelského rozhraní pro ovládání vzdáleného experimentu má standardní rozložení. V hlavičce (head) je realizováno propojení s externími soubory a v těle stránky (body) jsou zadány zobrazované prvky.

Hlavička HTML stránky v iSES Remduino Lab:

```
<head>

<!-- Nastavení zobrazování českého jazyka ve webové stránce -->
  <meta http-equiv="Content-Type" content="text/html; CHARSET=WINDOWS-1250">

<!-- Propojení s externími soubory kaskádových stylů - CSS -->
  <link rel="stylesheet" href="virtuallab/css/normalize.css">
  <link rel="stylesheet" href="virtuallab/css/main.css">
  <link rel="stylesheet" href="virtuallab/css/jquery-ui.css">
  <link rel="stylesheet" href="virtuallab/css/virtuallab.css">
  <link rel="stylesheet" href="css/main.css">

<!-- Propojení s externími soubory JavaScriptu -->
  <script src="virtuallab/js/vendor/modernizr-2.6.2.min.js"></script>
  <script src="virtuallab/js/vendor/flotr2.min.js"></script>
  <script src="virtuallab/js/vendor/jquery-1.10.2.min.js"></script>
  <script src="virtuallab/js/vendor/jquery-ui.min.js"></script>
  <script src="virtuallab/js/vendor/jquery.selectboxes.js"></script>
  <script src="virtuallab/js/vendor/jshashtable-3.0.js"></script>
  <script src="virtuallab/js/vendor/jquery.numberformatter-
1.2.4.min.js"></script>
  <script src="virtuallab/js/vendor/FileSaver.js"></script>
  <script src="virtuallab/js/vendor/Blob.js"></script>
  <script src="virtuallab/js/vendor/jpg.js"></script>
  <script src="virtuallab/js/plugins.js"></script>
  <script src="virtuallab/js/virtuallab/virtuallab.js"></script>
  <script src="virtuallab/js/virtuallab/virtuallab_helpers.js"></script>
  <script src="virtuallab/js/virtuallab/virtuallab_components.js"></script>
  <script src="virtuallab/js/virtuallab/virtuallab_processing.js"></script>
  <script src="virtuallab/js/virtuallab/virtuallab_CZ_lang.js"></script>
  <script
src="virtuallab/js/virtuallab/virtuallab_flotrplugins.js"></script>
  <script src="js/main_PZ01.js"></script>

<!-- Název webové stránky viditelný v internetovém prohlížeči -->
  <title>Vzdálená laboratoř s hradly
  </title>
</head>
```

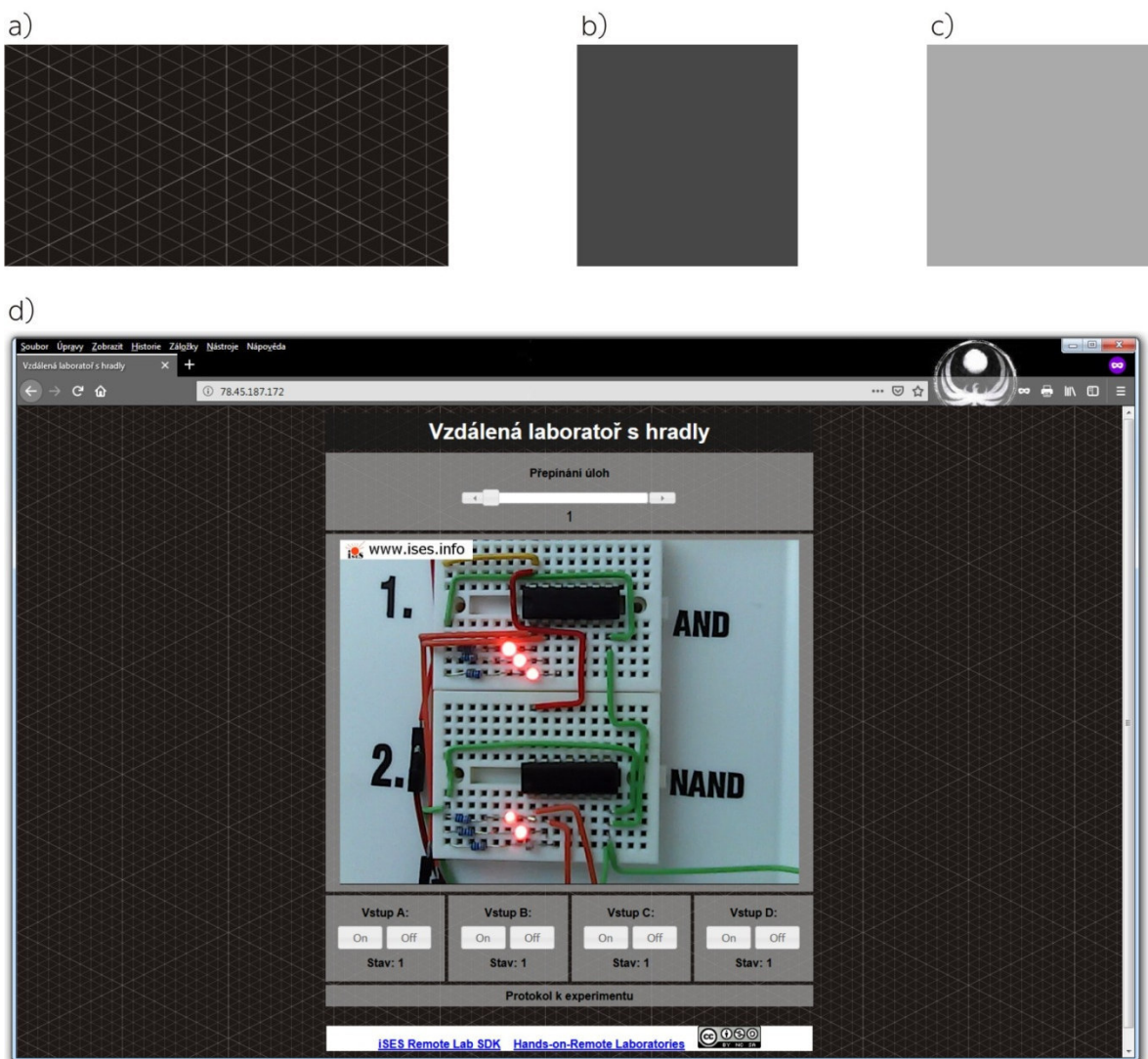
Design původní webové stránky v iSES Remduino Lab je velice strohý. Při tvorbě laboratoře byla na zřetel brána i snaha přiblížit vzdálené laboratoře co nejširší skupině lidí. Proto byl pro vzdálenou laboratoř vytvořen nový, modernější design.

Vzhledem k počtu odkazovaných externích souborů JavaScriptu a CSS bylo rozložení prvků na stránce realizováno pomocí tabulky. V současnosti se jedná o zastaralejší způsob, modernější variantou je pro rozmístění prvků využít kaskádové styly. V tomto případě však bylo použití tabulky spolehlivější a z hlediska uživatele skriptu iSES Remduino Lab jednodušší.

Design neaktivních prvků (prostý text, pozadí webové stránky, podbarvení tabulky rozložení prvků, ...) byl vytvořen pomocí kaskádových stylů zapsaných přímo do struktury HTML souboru, čímž byla určena nadřazenost jejich formátu nad formátem definovaným v externích souborech. Pro aktivní prvky (posuvník, tlačítka, aktivní čísla, odkazy...) byl design nastaven pomocí externích souborů CSS.

Pozadí webové stránky bylo vytvořeno pomocí jednoho neustále se opakujícího obrázku. Stejně tak i šedé podbarvení v jednotlivých polích tabulky, kterému však byla kromě odstínu šedé nastavená i průhlednost, aby lehce prosvítalo pozadí stránky (obrázek 40).

Vzhledem k tomu, že původní iSES Remduino Lab není vytvořen autorem bakalářské práce, byly na stránku s experimentem umístěny i odkazy na webové stránky autorů.



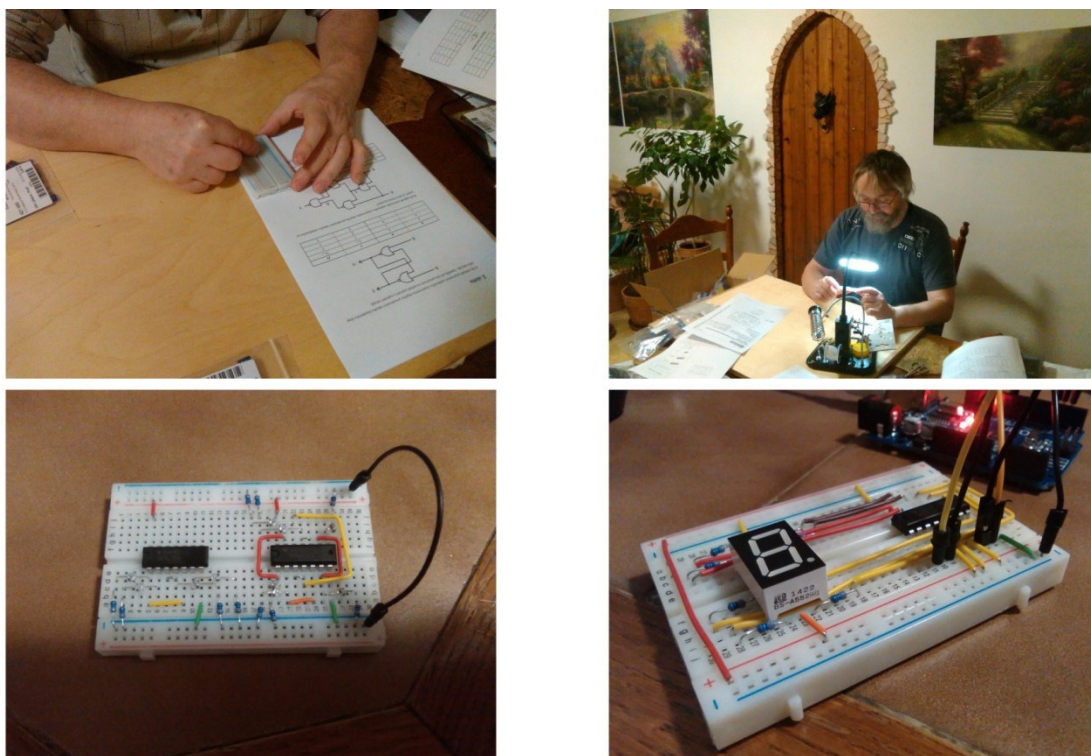
Obrázek 40 Design webové stránky vzdálené laboratoře

a) Obrázek použitý na vytvoření pozadí webové stránky, b) Obrázek použitý na podbarvení záhlaví tabulky, c) Obrázek použitý na podbarvení ostatních buněk tabulky, d) Výsledný design webové stránky

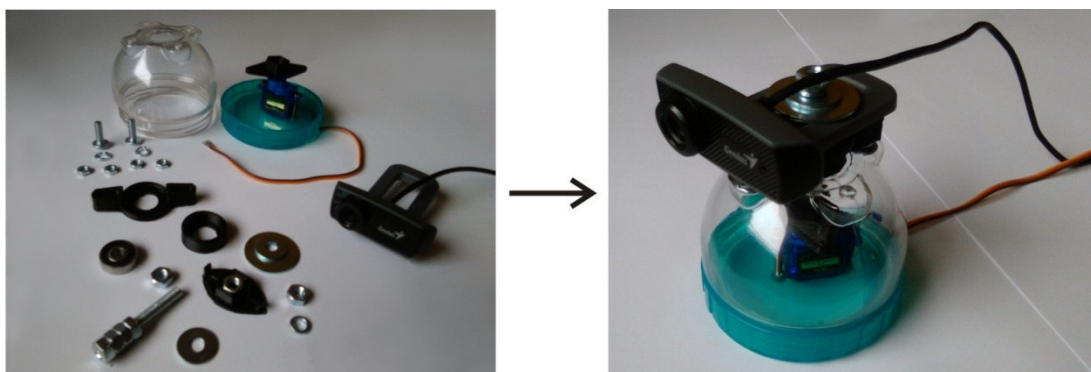
3.2.3 Výroba

Na základě rozmyšlených úloh bylo přistoupeno k výrobě vzdálené laboratoře. Nejprve byly vyrobeny jednotlivé úlohy, jejichž funkčnost byla postupně zkoušena pomocí propojení s počítačem (obrázek 41). Mezi problémy spojené s funkčností patřilo nejčastěji nepřesné propojení kabelů v nepájivých polích. Mezi další překážky však patřilo i obrácené zapojení LED diody do nepájivého pole, či nefunkční propojovací kabel, přestože se jednalo o originální výrobek.

Druhým krokem bylo sestavení mechanismu pro otáčení webové kamery. Vzhledem k tomu, že samotný mechanismus není během provozu laboratoře viditelný, nebylo nutno volit vizuálně hezké zpracování, na prvním místě bylo zajistit funkčnost. Proto byla jako hlavní nosič zvolena plastová miska, ke které byl přidělán servo motor. Pro připevnění kamery k motoru pak byly použity šrouby M6 s maticemi a některé díly byly vyrobeny na míru pomocí 3D tiskárny (obrázek 42).

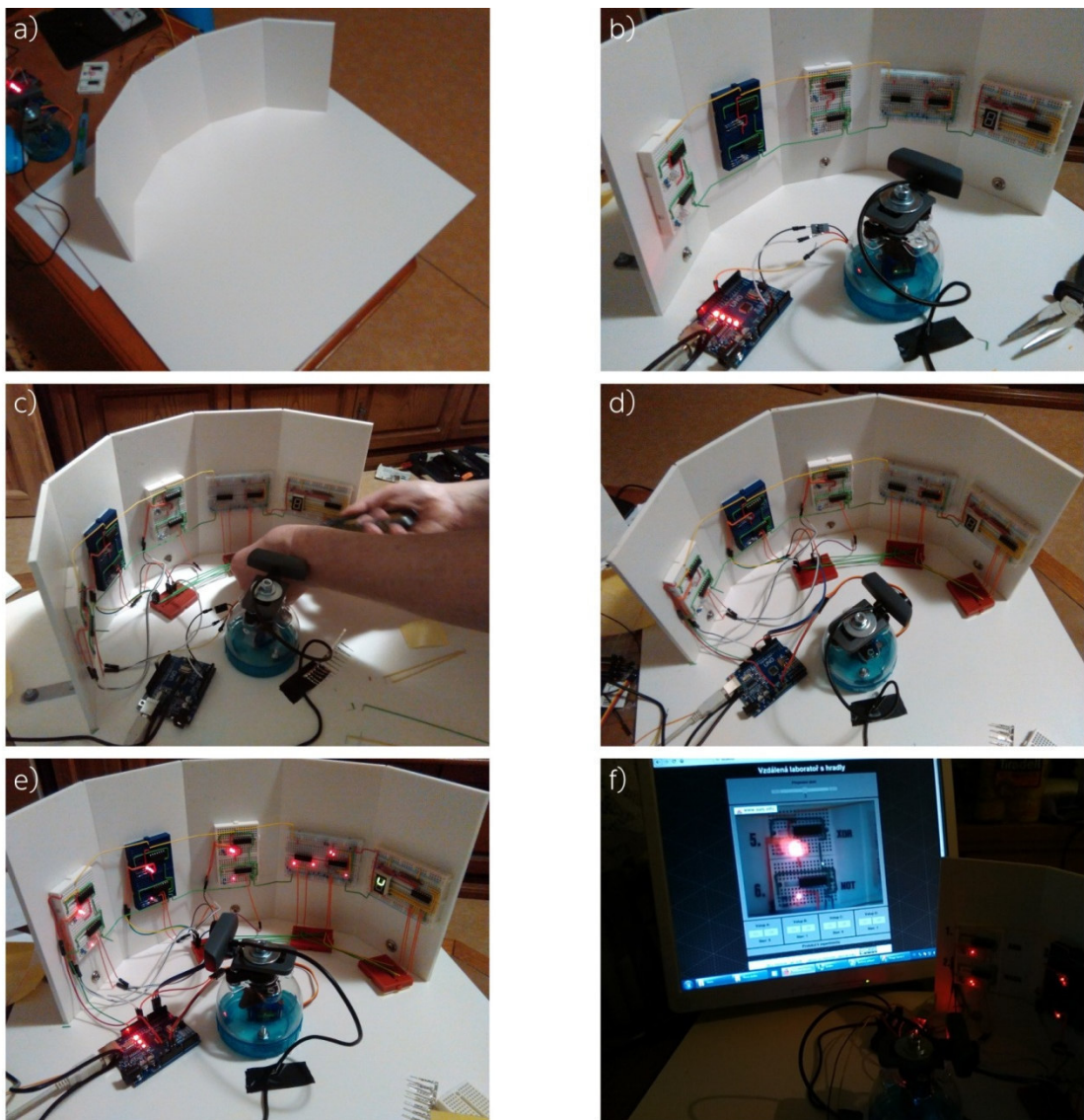


Obrázek 41 Výroba jednotlivých úloh



Obrázek 42 Výroba otočného systému pro webovou kameru

Po dokončení jednotlivých úloh byla vytvořena scéna. Vzhledem k tomu, že se jedná o práci s elektro-materiálem, byly použity desky z nevodivého materiálu. Kvůli nežádoucím odleskům původního povrchu byla deska polepena matnou bílou fólií, která zároveň slouží jako propojení jednotlivých segmentů scény. Poté byla k podstavové desce připevněna kamera na základě vzdálenosti od prostředního segmentu. Ostatní segmenty byly rozestaveny a připevněny k desce, k nim byly připevněny jednotlivé úlohy na nepájivých polích. Jednotlivé úlohy byly mezi sebou a Arduinem propojeny pomocí nepájivých polí připevněných k podstavové desce. Ani zde nebylo třeba dbát na vizuální stránku (stejně jako v případě kamery), jelikož při provozování experimentu tato propojení nejsou viditelná. Vizuálně upravené byly pouze konkrétní úlohy, u kterých byla i snaha o stejnou barevnou konvenci propojovacích kabelů (obrázek 43). Poté bylo ještě potřeba softwarově upravit úhly kamery pro jednotlivé úlohy a rychlost otáčení.



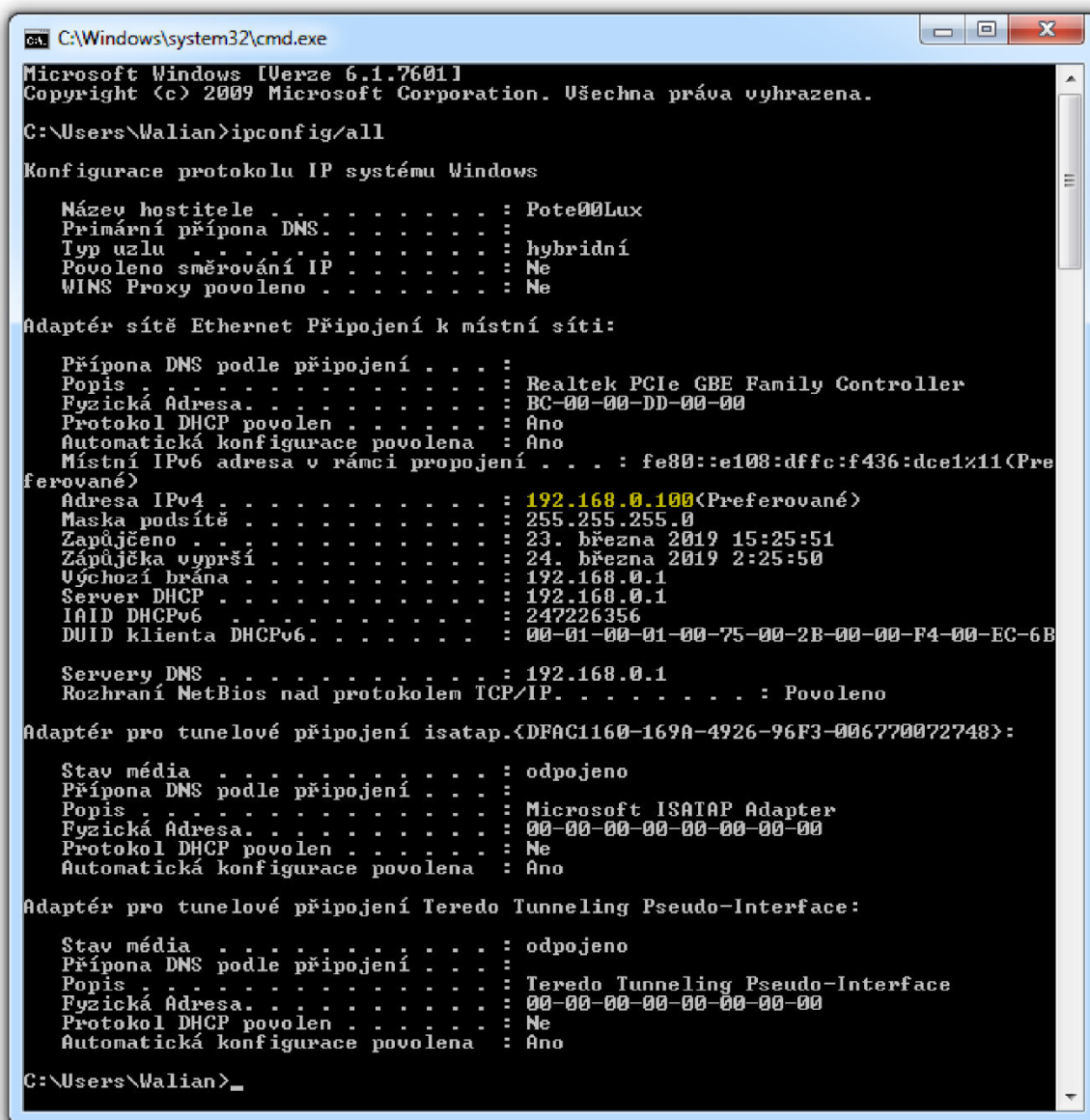
Obrázek 43 Výstavba vzdálené laboratoře

a) Základní rozestavení nosné konstrukce scény, b) Připevnění webové kamery a jednotlivých úloh do scény, c) Propojování úloh, d) Propojení jednotlivých komponent mezi sebou, e) Zkouška funkčnosti laboratoře, f) Softwarové nastavování optimálních úhlů a rychlostí otáčení

3.3 Vzdálený přístup k experimentu

Ve chvíli, kdy je vzdálená laboratoř fyzicky vyrobena a jsou naprogramovány a správně umístěny všechny skripty pro Arduino i server, může dojít ke spuštění všech serverových aplikací. V tu chvíli lze na počítači, kde je experiment umístěn, spustit vzdálenou laboratoř pouhým zadáním adresy *localhost* do prohlížeče.

Vzdálená laboratoř ale není vytvářena, aby se k ní přistupovalo z počítače, který laboratoř obsluhuje. Pro přístup z jiného počítače je potřeba zjistit IP adresu serveru. V rámci lokální sítě je přístup snadný, protože lze využít lokální IP adresu. Tu je možné získat například přes webovou stránku <http://www.mojeip.cz/lokalniip/>. Pokud není počítač připojen k internetu, neboť je počítáno s užíváním pouze v rámci vnitřní sítě, je nejspíš nejlehčí metodou pro systém Windows puštění příkazového řádku **cmd.exe** a zadání příkazu **ipconfig/all**. Ve vypsaných údajích je pak třeba požadovanou IP adresu vyhledat (obrázek 44).



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Verze 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Všechna práva vyhrazena.

C:\Users\Walian>ipconfig/all

Konfigurace protokolu IP systému Windows

Název hostitele . . . . . : Pote00Lux
Primární přípona DNS . . . . . :
Typ uzlu . . . . . : hybridní
Povoleno směrování IP . . . . . : Ne
WINS Proxy povoleno . . . . . : Ne

Adaptér sítě Ethernet Připojení k místní síti:

Přípona DNS podle připojení . . . :
Popis . . . . . : Realtek PCIe GBE Family Controller
Fyzická Adresa . . . . . : BC-00-00-DD-00-00
Protokol DHCP povolen . . . . . : Ano
Automatická konfigurace povolena . : Ano
Místní IPv6 adresa v rámci propojení . . . : fe80::e108:dffc:f436:dce1%11 (Preferované)
Adresa IPv4 . . . . . : 192.168.0.100 (Preferované)
Maska podsítě . . . . . : 255.255.255.0
Zapůjčeno . . . . . : 23. března 2019 15:25:51
Zápůjčka vyprší . . . . . : 24. března 2019 2:25:50
Účchází brána . . . . . : 192.168.0.1
Server DHCP . . . . . : 192.168.0.1
IAD DHCPv6 . . . . . : 247226356
DUID klienta DHCPv6 . . . . . : 00-01-00-01-00-75-00-2B-00-00-F4-00-EC-6B

Servery DNS . . . . . : 192.168.0.1
Rozhraní NetBios nad protokolem TCP/IP . . . . . : Povoleno

Adaptér pro tunelové připojení isatap.<DFAC1160-169A-4926-96F3-006770072748>:

Stav média . . . . . : odpojeno
Přípona DNS podle připojení . . . :
Popis . . . . . : Microsoft ISATAP Adapter
Fyzická Adresa . . . . . : 00-00-00-00-00-00-00-00
Protokol DHCP povolen . . . . . : Ne
Automatická konfigurace povolena . : Ano

Adaptér pro tunelové připojení Teredo Tunneling Pseudo-Interface:

Stav média . . . . . : odpojeno
Přípona DNS podle připojení . . . :
Popis . . . . . : Teredo Tunneling Pseudo-Interface
Fyzická Adresa . . . . . : 00-00-00-00-00-00-00-00
Protokol DHCP povolen . . . . . : Ne
Automatická konfigurace povolena . : Ano

C:\Users\Walian>_
```

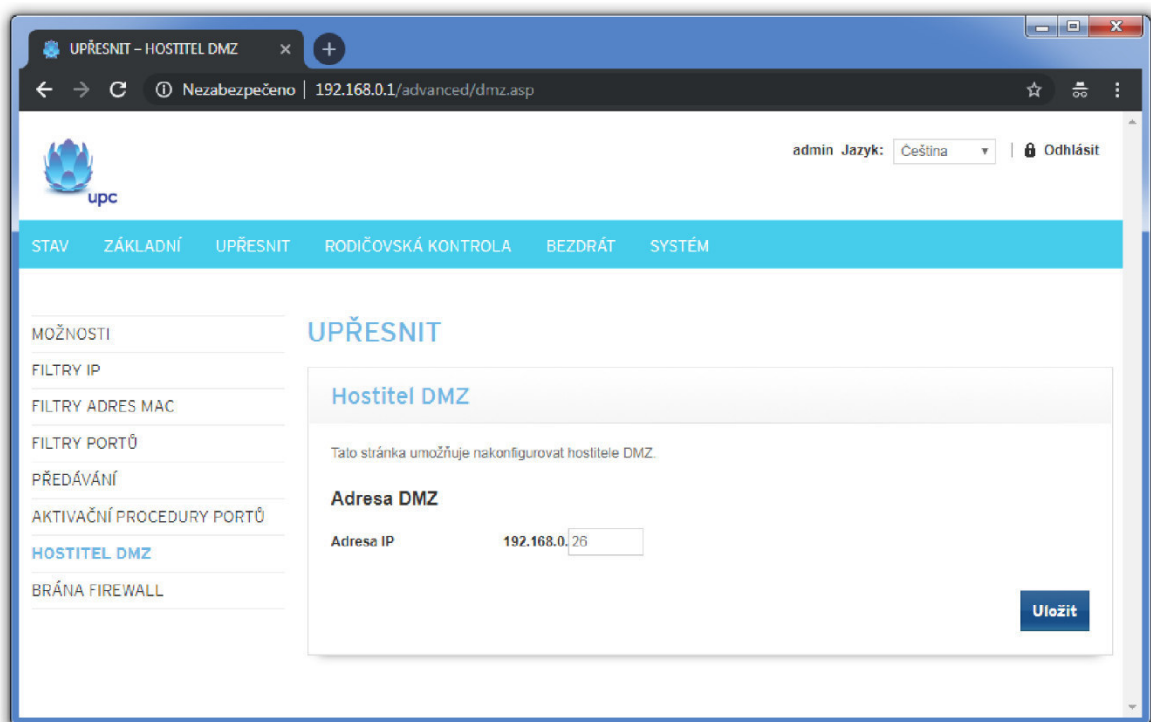
Obrázek 44 Lokální IP adresa nalezená přes příkazový řádek

Pro přístup ke vzdálené laboratoři z počítače mimo lokální síť je nutné přistupovat přes veřejnou adresu routeru, který komunikuje s lokální sítí. Nastavení přístupu by v takovém případě v rámci školního serveru řešil tamní IT specialista.

V rámci bakalářské práce byl využit router od poskytovatele internetového připojení, firmy UPC. Aby mohlo být k routeru přistoupeno z internetu, je nezbytné, aby měl veřejnou IP adresu. V tomto požadavku většina poskytovatelů internetu nemá problém. Druhým požadavkem je, aby byla IP adresa routeru známá. Globální IP adresa routeru jde snadno zjistit na webové stránce <https://www.mojeip.cz/>. Adresa, která je po načtení stránky zobrazena, je opravdu globální IP adresou routeru, ale neznamená to, že bude globální adresou i následující den. K tomu, aby byla jistota, že se IP adresa nebude měnit, sloužila tzv. statická IP adresa, kterou ale většina poskytovatelů internetu přestala nabízet jako standardní službu.

Klouzavá IP adresa, kterou má router od UPC, se nemění příliš často. Firmware routeru umožňuje přesměrování uživatelů dotazujících se na jeho IP adrese na jedno ze zařízení v lokální síti. Ve vlastnostech je nutné najít nastavení hostitele DMZ. Nastavením lokální IP adresy dojde k vystrčení jednoho zařízení do tzv. demilitarizované zóny (obrázek 45). Pokud je tímto zařízením právě počítač se spuštěným vzdáleným experimentem, stává se dostupným ze všech koutů internetu.

Toto řešení umožňuje vzdálený přístup, ale je potřeba hlídat změnu IP adresy routeru a tu pak sdělovat zájemcům o vzdálený experiment. Řešením může být zaregistrováním domény a přiřazováním aktuální IP adresy, nebo odkazováním na laboratoř pomocí hypertextového odkazu na jiných webových stránkách.



Obrázek 45 Nastavení lokální IP adresy počítače v demilitarizované zóně

3.4 Porovnání vzdáleného a reálného experimentu

Vzdálený a reálný experiment je možné porovnávat z různých hledisek. Jednoznačné porovnání je možné provést z hlediska finanční náročnosti. Již méně jednoznačné je porovnání z hlediska časové náročnosti, protože pro přípravu vzdáleného experimentu je zapotřebí znalostí nejen v oblasti prováděného experimentu, ale především v oblasti elektrotechnické a informačně technologické. A nejtěžší je porovnání experimentů z hlediska dopadu na studenta, neboť obě metody mají své velké výhody i velké zápory (tabulka 1).

Tabulka 1 Komplexní porovnání reálného a vzdáleného experimentu

	Reálný experiment	Vzdálený experiment
	1 Ks	1Ks
Promyšlení experimentu	Ano	Ano
Znalost logiky a TTL hradel	Ano	Ano
Znalost základů elektroniky	Ano	Ano
Znalost HTML, JavaScript, CSS	Ne	Ano
Znalost ISES Remduino Lab SDK	Ne	Ano
Znalost ovládání Arduino UNO	Ne	Ano
Znalost IP konektivity	Ne	Ano
Finance na materiál	307 Kč	4 211 Kč
Finance na celoroční provoz	0 Kč	8 640 Kč
Finance připojení k internetu	0 Kč	500 Kč/měsíc
Finanční ohodnocení učitele za mimoškolní práci	nezohledněno	nezohledněno
Zdroj	plochá baterie	zásuvka 230 V
Připojení k internetu	Ne	Ano
Prostorové nároky při provozu	učebna	cca 1 metr ²
Přístupnost	při výuce	24/7/365
Osoba kontrolující funkčnost laboratoře	Ne	Ano
Čas na promyšlení a přípravu	120 minut	3 měsíce
Čas na úpravu skriptů ISES, tvorba web stránky a připojení na veřejnou IP	0	1200 minut
Čas na výrobu	240 minut	720 minut
Čas na přípravu protokolu	120 minut	240 minut
Rozvíjení kooperace studentů	Ano	Ne
Rozvíjení kognitivity	Ano	Ano
Rozvíjení psychomotoriky	Ano	Ne
Rozvíjení komunikace	Ano	Ano
Rozvíjení ICT kompetencí	Ne	Ano
Student se podílí na tvorbě experimentu	Ano, jen velmi jednoduché exp.	Ano, jako mimoškolní aktivita

3.4.1 Porovnání finanční náročnosti

Při cenové kalkulaci reálného experimentu je třeba nejprve rozmyslet, jakým stylem bude experiment prováděn. Ideální varianta je, že si každý student provádění experimentu vyzkouší. Další možností je práce ve skupinách. Práce ve skupinách má z edukativního hlediska navíc tu výhodu, že student je učen komunikaci a kooperaci s ostatními, což je v současnosti velice důležité, neboť v mnoha zaměstnáních je vyžadována práce v týmech. Studenti jsou tím navíc nuceni ohledně experimentu diskutovat, čímž jsou více aktivizováni. Při vytvoření skupin po čtyřech studentech je možné ušetřit i 75 % nákladů. Což znamená, že při snížení celkových nákladů lze předpokládat, že výuka bude mít na studenty větší dopad. Poslední možností je předvedení experimentu učitelem. Tato varianta je vhodná především u experimentů, kde je nutné dbát zvýšené opatrnosti při provádění.

Navržený experiment s hradly je možné provádět bez nutnosti pájet spoje. Je proto možné zapojit všechny studenty, aniž by přibyla starost s bezpečností práce. Následující cenová kalkulace (tabulka 2) nejprve ukazuje náklady na jednoho studenta a následně náklady na materiál pro třicetičlennou třídu.

Při cenové kalkulaci byly uvažovány ceny z tuzemských obchodů, kdy byl uvažován nákup většího množství součástek, čímž je možné získat výrazně výhodnější ceny.

Tabulka 2 Cenová kalkulace reálného experimentu na hodině

Položka	Jednotková cena [Kč/ks]	Počet kusů [Ks]	Cena [Kč]
Hradlo AND (SN74LS08)	11,00	1	11,00
Hradlo NAND (SN74LS132)	18,00	1	18,00
Hradlo OR (SN74LS32)	14,00	1	14,00
Hradlo NOR (SN74LS02)	10,00	1	10,00
Hradlo XOR (SN74LS86A)	7,00	1	7,00
Hradlo NOT (SN74LS14)	9,00	1	9,00
LED diody	1,50	6	9,00
Rezistor 330R	0,20	13	2,60
BCD dekodér (SN74LS47)	37,00	1	37,00
Sedmisegmentový LED panel	11,00	1	11,00
Nepájivé pole ZY-60	56,00	1	56,00
Sada propojovacích vodičů	99,00	1	99,00
Baterie	24,00	1	24,00
Náklady na 1 experiment			307,60 Kč
Náklady pro 30 studentů = 30 experimentů			9228,00 Kč
Náklady pro 8 skupin po 4 studentech			2460,80 Kč

Cenová kalkulace vzdáleného experimentu zahrnuje dvě složky. Tou první jsou pořizovací náklady (tabulka 3) a druhou jsou provozní náklady. V pořizovacích nákladech je zahrnut i jednorázový licenční poplatek 500,- Kč na jeden experiment, který je zaveden především pro financování vylepšování systému iSES Remote Lab SDK [36].

Hlavními provozními náklady se rozumí náklady na elektrickou energii, náklady na údržbu, náklady na skladovací prostory nebo náklady na připojení k internetu. Navržený experiment nevyžaduje žádnou speciální údržbu, která by si vyžádala víc než čas učitele. Připojení k internetu je v současné natolik rychlé a kvalitní, že není zapotřebí připlácet za nadstandardní služby. Experiment obvykle nezabírá tolik místa, aby nemohl být umístěn v méně využívaných prostorách školy. Hlavní provozní náklady tedy tvoří spotřebovaná elektrická energie, která ovšem nebyla vyčíslena na základě měření skutečné spotřeby, ale na základě teoretického výpočtu. Webová kamera i Arduino jsou napájeny z USB portu počítače, tudíž je celý experiment je napájen 250W zdrojem počítače. Při maximálním odběru a průměrné ceně 4,- Kč/kWh lze náklady odhadnout na 24,- Kč/den, resp. 720 Kč/měsíc, resp. 8640,- Kč/rok.

Z dostupných kalkulací lze vyvodit, že finančně je vzdálená laboratoř daleko víc náročná, což způsobují především provozní náklady.

Tabulka 3 Cenová kalkulace pořizovacích nákladů vzdáleného experimentu

Položka	Jednotková cena [Kč/ks]	Počet kusů [Ks]	Cena [Kč]
Hradlo AND (SN74LS08)	11,00	1	11,00
Hradlo NAND (SN74LS132)	18,00	2	36,00
Hradlo OR (SN74LS32)	14,00	1	14,00
Hradlo NOR (SN74LS02)	10,00	2	20,00
Hradlo XOR (SN74LS86A)	7,00	1	7,00
Hradlo NOT (SN74LS14)	9,00	1	9,00
LED diody	1,50	27	40,50
Rezistor 330R	0,20	34	6,80
BCD dekodér (SN74LS47)	37,00	1	37,00
Sedmisegmentový LED panel	11,00	1	11,00
Nepájivé pole ZY-60	56,00	2	112,00
Nepájivé pole ZY-170	29,00	9	261,00
Sada propojovacích vodičů	99,00	1	99,00
Sada propojovacích drátků	249,00	1	249,00
Arduino Uno (klon)	198,00	1	198,00
Repasovaný počítač	2000,00	1	2000,00
Servo motor	120,00	1	120,00
Webová kamera	280,00	1	280,00
Materiál pro upevnění experimentu	200,00	1	200,00
Licenční poplatek pro iSES Remduino Lab	500,00	1	500,00
Celkové pořizovací náklady			4211,30 Kč

3.4.2 Porovnání časové náročnosti

Ať se jedná o jakýkoliv experiment, je nutná řádná příprava. Základem všeho je volba vhodného experimentu a stanovení, co má laboratorní cvičení studentovi předat. Vzdálené laboratoře mohou poskytnout zajímavá data, ale laboratoř musí být předem vyrobena včetně podkladů. Vhodná je příprava teoretické průpravy problematikou, seznámení s experimentem, seznámení s ovládáním experimentu a protokolů pro vypracování laboratorního cvičení. Oproti tomu je u reálného experimentu možné se studenty experiment sestavovat. Cvičení s hradly s využitím nepájivých polí je názornou ukázkou takového experimentu. I v takovém případě je však nutné připravit podklady, podle kterých se bude během vyučování postupovat. Výhodou ale je, že při nejasnosti v podkladu může učitel problematiku část upřesnit.

Během přípravy reálného experimentu musí učitel dobře znát problematiku, které se experiment týká. I přesto je příprava veškerých podkladů zdlouhavá. Vzhledem k omezenému času během výuky je nutné, aby bylo vše předem připravené a aby nedošlo k přerušení experimentu z důvodu skončení vyučovací hodiny. Podle náročnosti experimentu se doba přípravy mění, ale obvykle vyžaduje několik hodin.

K přípravě vzdáleného experimentu nestačí jen znát problematiku, které se experiment týká. Je potřeba celý experiment připravit tak, aby bylo možné veškeré činnosti ovládat pomocí softwaru. V tu chvíli se z banální operace, jako je např. sepnutí spínače, stává problematika zahrnující znalosti z elektrotechniky a informačních technologií. Jen díky existenci prototypovacích mikroprocesorových desek typu Arduino nebo Raspberry Pi je v současné době reálné pomýšlet na to, že realizaci vzdálené laboratoře dokáže vytvořit i běžný člověk. Když se k tomu přidá existence podpůrných systémů, jako je např. iSES Remote Lab SDK, šance na zdárné vytvoření rapidně stoupá.

Je potřeba kriticky podotknout, že i přes všechny podpůrné pomůcky učitel při vytváření svého prvního vzdáleného experimentu stráví možná několik desítek hodin studováním způsobů, jak vše propojit, aby vznikl fungující celek. Při využití Arduina a iSES Remduino Lab, což je v současnosti jedna z nejsnazších možností, je potřeba se vyznat ve čtyřech programovacích jazycích. Je tedy nutné počítat s tím, že první vlastní vzdálená laboratoř vznikne pravděpodobně až po několika měsících snažení.

Jako příklad rozdílu časové náročnosti reálného a vzdáleného experimentu je možné uvést kvalitativní experiment pro prokázání bezdrátového přenosu energie. Reálný experiment na hodině by obnášel lakovaný drát průměru 0,5 mm, výkonový tranzistor NPN, rezistor, LED diody a 4,5V nebo 9V baterii. Učitel by sestrojil z drátu, tranzistoru, rezistoru a baterie vysílací cívku, tzv. oscilátor. Studenti by potom měli za úkol namotat cívky různých tvarů s různými počty závitů a připojit k nim LED diody. Následně by přibližováním k oscilátoru prokazovali jev.

Pro vzdálený experiment stejného typu je navíc potřeba sestrojil lineární pojezd, který je vysouvaný motorem, který je ovládán mikroprocesorem převodníku, který dostává příkazy od serveru, který komunikuje s uživatelem na druhé straně světa.

Časová náročnost na první vzdálený experiment je velká, ale po proniknutí do tajů této problematiky se doba výrazně zkracuje.

Závěr

Bakalářská práce je zaměřena na možnost využití vzdálených laboratoří, jelikož mnohými výzkumy je dokázáno, že názorně demonstrační metody jsou pro výuku technických předmětů velice prospěšné vzhledem k úspěšnosti zapamatování si množství předkládaných informací a pochopení celkové problematiky přednášené látky.

Práce je rozdělena na teoretickou a praktickou část. V teoretické části jsou rozebrány metody výuky se zaměřením na metody názorně demonstrační a metody praktické činnosti žáků. Dále pak byl zmapován současný stav problematiky vzdálených laboratoří z hlediska nejen České republiky, ale i z hlediska celosvětového. V praktické části pak byl navržen a zrealizován vlastní experiment se vzdáleným přístupem a byly popsány z toho plynoucí závěry, čímž bylo naplněno zadání bakalářské práce.

Pro tvorbu vzdáleného experimentu byla použita prototypovací platforma Arduino, která se v současnosti těší velké oblibě především z hlediska poměru cena / výkon. Na základě konzultací s doc. RNDr. Františkem Lustigem, CSc. byl pro softwarovou část použit jím a jeho týmem vyvíjený ovládací systém iSES Remduino Lab. Z hlediska výroby vzdálené laboratoře se jednalo o velké usnadnění, nicméně i tak bylo nutno velkou část naprogramovat, aby byl realizovaný experiment plně funkční. Součástí práce je tudíž kromě fyzické výstavby laboratoře popsán i metodický způsob, jak lze dosáhnout funkčního propojení softwarových částí, včetně podrobně komentovaných ukázek použitých počítačových skriptů.

Z hlediska hardwaru a softwaru bylo prvním důležitým krokem pečlivě rozmyslet konkrétní experiment. V rámci bakalářské práce byl vybrán experiment založený na použití hradel. Ta fungují na principech výrokové logiky, což je z hlediska experimentu zajímavé nejen pro ukázkou konkrétních elektro-součástek, ale i z matematického hlediska. Při rozmyšlení laboratoře bylo rozhodnuto, že z výukového hlediska by bylo vhodné experiment rozdělit na pět menších úloh, jejichž obtížnost se postupně zvyšuje. První tři úlohy jsou tak velice jednoduché a jejich základním úkolem je usnadnit studentům pochopení fungování hradel. Hlavním úkolem je zde správně vyplnit pravdivostní tabulky pro jednotlivé typy funkcí (AND, NAND, OR, NOR, XOR, XNOR, NOT a opakovač). Čtvrtá úloha je rozdělena na dvě části – v první části je studentovi představen R-S bistabilní klopný obvod a cílem je správně doplnit pravdivostní tabulku. Ve druhé části čtvrté úlohy je studentovi předložen integrovaný obvod s neznámou funkcí. Na základě pozorování funkčnosti experimentu a vyplnění pravdivostní tabulky by pak měl být student schopen odvodit, že se jedná o exkluzivní logický součet (XOR). V páté úloze je hlavním úkolem správně doplnit pravdivostní tabulku pro BCD dekodér a vyznačit rozsvícené diody sedmissegmentového panelu pro všechny kombinace. V rámci laboratorního protokolu byla vytvořena ještě šestá úloha, která však je pouze teoretická, ale odkazuje se na předchozí úlohy, a na jejímž základě by měl student prokázat, že zcela pochopil fungování hradel.

Na základě zkušeností získaných při návrhu a výrobě vzdálené laboratoře bylo možno učinit závěry týkající se vhodnosti využívání vzdálených laboratoří. Finanční analý-

za ukázala, že pro levnější experimenty, jako je např. experiment s hradly zkonstruovaný v rámci bakalářské práce, se nevyplatí konstruovat vzdálenou laboratoř. Je mnohem výhodnější takový experiment provést se studenty přímo na hodině, kde mohou být navíc ještě aktivizováni vyučujícím. Finanční nevýhodou vzdáleného experimentu jsou totiž kromě nákladů na samotnou konstrukci i náklady vyplývajícího z provozu laboratoře. Je nutno zajistit neustálý přívod elektrické energie (ideálně použít i další pomůcky pro ochranu laboratoře při náhodném kolapsu elektrické rozvodné sítě, čímž přibývají další náklady na pořízení) a případně se mohou objevit i náklady spojené s potřebou neustálého připojení k internetu, což by však v případě většiny vzdělávacích institucí neměl být výrazný problém. Nicméně pokud by byla takto vytvořena laboratoř, kde by byly výrazně vyšší pořizovací náklady na jednotlivé součásti (např. by byl využit mikroskop, jehož cena se může pohybovat mezi 50 – 100 tis. Kč), bylo by výhodné takový experiment zrealizovat. Provozní náklady by v tu chvíli nebyly v takovém nepoměru vůči nákladům na pořízení.

Pokud by byla laboratoř využívána jako nástroj k provádění laboratorních cvičení v rámci výuky, bylo by potřeba vyřešit i problém související s přístupem k ovládání laboratoře. Ve chvíli, kdy by experiment mohl být v daný čas ovládán pouze jedním studentem, ostatní studenti musí čekat. Pokud by laboratoř byla přístupná veřejně tak je nutno počítat s tím, že může být v kritický okamžik navíc blokována i osobami, které v daný okamžik nepotřebují experiment přímo ovládat. Z toho hlediska je pochopitelný krok mnoha institucí, které přístup k laboratořím povolují až na základě registrace, případně na základě požadavku uživatele pošlou generovaný jednorázový přihlašovací kód. Tím je částečně zajištěn přednostní přístup studentům dané instituce. Bylo by výhodné se v takovém případě zamyslet i nad možností vytvoření simulace daného laboratorního experimentu, který by bylo možno ovládat větším počtem uživatelů a kde by se v tu chvíli stala jedinou překážkou pro užívání neomezeným množstvím zájemců kapacita serveru. Zájemci o laboratorní cvičení by se tak nejprve mohli seznámit se simulací před tím, než by jim bylo umožněno samostatně ovládat experiment.

Překážkou pro výrobu laboratoře může být z hlediska vyučujícího i počáteční časová investice, protože je nutno nejen rozumět jevu prezentovanému pomocí laboratoře, často je nutno sestavit i pohyblivé zařízení (pojezdy, otočná zařízení, ...) a nezbytnou součástí je vyznat se v programování webových stránek a v psaní skriptu pro ovládání laboratoře – v případě bakalářské práce bylo třeba se alespoň rámcově zorientovat v JavaScriptu, kaskádových stylech, HTML, C++ a programovacím jazyku, který využívá Arduino. Po nabytí zkušeností s výrobou jedné laboratoře by výroba případné další laboratoře však již byla znatelně snazší.

Na základě zkušeností s výrobou experimentu lze tvrdit, že vzdálené laboratoře mohou být velice zajímavým názorným doplněním výuky, ale mají i své nevýhody a před případnou výrobou laboratoře je nutno pečlivě zvážit klady a zápory takového řešení z nejrůznějších hledisek.

Z výše uvedeného je patrné, že všechny cíle bakalářské práce byly splněny.

Seznam použité literatury

- 1 **VANĚČEK, David a kol.** *Didaktika technických odborných předmětů*. Praha: České vysoké učení technické v Praze, 2016. ISBN 978-80-01-05991-3.
- 2 **MAŇÁK, Josef a Vlastimil ŠVEC.** *Výukové metody*. Brno: Paido, 2003. 219 s. ISBN 80-7315-039-5.
- 3 **PRŮCHA, Jan, Eliška WALTEROVÁ a Jiří MAREŠ.** *Pedagogický slovník*. 7., aktualiz. a rozš. vyd. Praha: Portál, 2013. ISBN 978-80-262-0403-9.
- 4 **ČERVENKOVÁ, Iva.** 2013. *Výukové metody a organizace vyučování*. [online]. [cit. 2019-02-17]. Dostupné z: <http://projekty.osu.cz/svp/opory/pdf-cervenkova-vyukove-metody-a-organizace-vyucovani.pdf>
- 5 **BENDL, Stanislav, KUCHARSKÁ, Anna a kol.** *Kapitoly ze školní pedagogiky a školní psychologie*. Praha: Univerzita Karlova v Praze, 2008. ISBN 978-80-7290-366-5.
- 6 **LERNER, Isaak Jakovlevič.** *Didaktické základy metod výuky*. Praha: SPN, 1986.
- 7 **KLEMENT, Milan, Jiří DOSTÁL a Jiří KLEMENT.** *Metody realizace a hodnocení výuky* [online]. Druhé vydání. Olomouc, 2014 [cit. 2019-02-09]. ISBN 978-80-87658-21-5. Dostupné z: <https://docplayer.cz/16959753-Metody-realizace-a-hodnoceni-interaktivni-vyuky.html>
- 8 **MAŇÁK, Josef.** *Nárys didaktiky*. 1. vyd. Brno: PdF MU, 1990. ISBN 80-210-0210-7.
- 9 **SLAVÍK, Jan a Jaroslav NOVÁK.** *Počítač jako pomocník učitele: efektivní práce s informacemi ve škole*. Praha: Portál, 1997. Pedagogická praxe. ISBN 80-7178-149-5.
- 10 *Kužel z zkušenosti (Cone Of Experience)*. Edgar Dale [online]. 2007 [cit. 2019-02-17]. Dostupné z: http://it.pedf.cuni.cz/strstud/edutech/2006__Dale__Ovsenak/cone_of_learning.html
- 11 **DOSTÁL, Jiří.** *Experiment jako součást badatelsky orientované výuky*. Trendy ve vzdělávání, 2013, 1.1: 9-19. ISSN 1805-8949.
- 12 **LOJOVÁ, Gabriela.** *Styly a strategie ve výuce cizích jazyků*. Vyd. 1. Praha: Portál, 2011. ISBN 978-80-7367-876-0.
- 13 **DUNN, Ritta, DUNN, Kenneth.** *Teaching elementary students through their individual learning styles: practical approaches for grades 3-6*. Allyn & Bacon, 1992. ISBN 978-0205132219.
- 14 **DUNN, Ritta, DUNN, Kenneth.** *Teaching secondary students through their individual learning styles: practical approaches for grades 7-17*. Allyn & Bacon, 1993. ISBN 978-0205133086.
- 15 **MAREŠ, Jiří.** *Styly učení žáků a studentů*. Praha: Portál, 1998. 240 s. ISBN 80-7178-246-7.

- 16 **HULANOVÁ, Lenka.** *Internetová kriminalita páchaná na dětech: psychologie internetové oběti, pachatele a kriminality.* 1. vyd. Praha: Triton, 2012. 217 s. ISBN 978-80-7387-545-9.
- 17 Český statistický úřad [online]. 2019 [cit. 2019-03-24]. Dostupné z: <https://www.czso.cz/csu/czso/domov>
- 18 **SVOBODA, Emanuel a Růžena KOLÁŘOVÁ.** *Didaktika fyziky základní a střední školy: vybrané kapitoly.* Praha: Karolinum, 2006. ISBN 8024611813.
- 19 **LORENZOVÁ, Jitka.** *Etické hodnoty a normy v současných proměnách školy.* In DOROTÍKOVÁ, S. (ed.) *Profesní etika učitelství.* Praha: Pedagogická fakulta UK, 2003, s. 78-105. ISBN 80-7290-102-8.
- 20 **PETTY, Geoffrey.** *Moderní vyučování.* 3. vyd. Praha: Portál, 2004. ISBN 80-7178-978-X.
- 21 **BERTRAND, Yves.** *Soudobé teorie vzdělávání.* Praha: Portál, 1998. Studium (Portál). ISBN 80-7178-216-5.
- 22 **BROM, Pavel.** *Integrovaný e-learning v oblasti kvantové fyziky a optiky.* Praha, 2018. Disertační práce. Univerzita Karlova, Matematicko fyzikální fakulta. Vedoucí práce Doc. RNDr. František Lustig, CSc.
- 23 *PhET: Free online physics, chemistry, biology, earth science and math simulations* [online]. Colorado: University of Colorado Boulder [cit. 2019-03-24]. Dostupné z: <https://phet.colorado.edu/>
- 24 *Trendy ve vzdělávání* [online]. Olomouc: Pedagogická fakulta Univerzity Palackého v Olomouci [cit. 2019-03-24]. Dostupné z: <https://tvv-journal.upol.cz/>
- 25 **LUSTIG, František.** *Školní fyzikální laboratoře současnosti a budoucnosti* [online]. 2004 [cit. 2019-03-24]. Dostupné z: http://www.ises.info/old-site/clanky_pdf/Lustig%20didfyz%202004.pdf
- 26 **Lustig, František, Brom, Pavel, Kuriščík, Pavel, Dvořák, Jiří.** „Hands-on-Remote“ Laboratories. Proceedings of the 15th International Conference on Remote Engineering and Virtual Instrumentation (REV 2018), March 21-23 2018, University of Applied Science Dusseldorf (HSD), Germany, 2018. Str. 446-455. [online] http://www.ises.info/old-site/clanky_pdf/Hands-on-remote_Laboratory_REV2018_1201.pdf [cit. 20. 7. 2018]
- 27 *IPv6 a IPv4.* Avonet [online]. [cit. 2019-03-24]. Dostupné z: <https://avonet.cz/25080-ipv6-a-ipv4>
- 28 **Bližňák, Michal a kol.** *Virtuální laboratoř pro vývoj aplikací s mikroprocesory a FPGA.* Vyd. 1. Brno: CERM, 2011. 81 s. ISBN 978-80-7204-754-3.
- 29 *Remote-LAB GymKT* [online]. Klatovy: Gymnázium Jaroslava Vrchlického v Klatovech [cit. 2019-03-24]. Dostupné z: <http://remote-lab.fyzika.net/>
- 30 *iSES – Internet School Experimental System* [online]. [cit. 2019-02-27]. Dostupné z: <http://www.ises.info/index.php/cs/>
- 31 *Arduino - Home* [online]. [cit. 2019-03-17]. Dostupné z: <https://www.arduino.cc/>

- 32 *PSPad - free unicode vývojářský textový editor určený pro HTML, PHP, XHTML, JavaScript, ASP, Perl, C a mnoho dalších jazyků* [online]. [cit. 2019-03-20]. Dostupné z: <http://www.pspad.com/cz/>
- 33 *Fritzing* [online]. [cit. 2019-03-20]. Dostupné z: <http://fritzing.org/home/>
- 34 *Arduino-shop.cz: VELKOOBCHOD, MALOOBCHOD S ARDUINEM* [online]. [cit 2019-03-20]. Dostupné z: <https://arduino-shop.cz/>
- 35 *GM electronic | elektronické součástky, komponenty | GM electronic, spol. s.r.o.* [online]. [cit 2019-03-20]. Dostupné z: <https://www.gme.cz/>
- 36 **LUSTIG, František.** *Vzdálené experimenty s Arduinem, s USB a COM měřicími přístroji - Internetové školní experimentální studio „iSES Remote Lab SDK“.* In: ROSECKÝ, Jan. *Počítač ve škole 2016 – sborník příspěvků.* Nové Město na Moravě: Gymnázium Vincence Makovského se sportovními třídami, 2016. ISBN 978-80-905765-6-8.

Seznam obrázků

Obrázek 1 Vztah mezi vyučováním a učením [2]	7
Obrázek 2 Vztahy mezi výukovou metodou, strategií učení/vyučování a stylem učení/vyučování [2].....	7
Obrázek 3 Model pyramidy učení dle S. Shapira [1].....	8
Obrázek 4 Členění metod podle komunikačních faktorů [2].....	9
Obrázek 5 Daleho kužel zkušenosti	11
Obrázek 6 Model vztahů mezi motivací, záměry, průběhem a výsledky učení [13]	13
Obrázek 7 Množství jednotlivců používajících internet [17].....	15
Obrázek 8 Použití vybraných informačních technologií studenty [17].....	16
Obrázek 9 Studenti využívající internet k vybraným vzdělávacím aktivitám [17].....	16
Obrázek 10 Rozdělení experimentů podle Broma [22]	19
Obrázek 11 Oficiální loga systému iSES a prototypovací platformy Arduino [30, 31]...	23
Obrázek 12 Pohled na systém iSES [30].....	24
Obrázek 13 Základní princip fungování vzdálené laboratoře	25
Obrázek 14 Stažení příkladů pro iSES Remduino Lab z webových stránek tvůrců [30]	26
Obrázek 15 Umístění souborů příkladu iSES Remduino Lab	27
Obrázek 16 Nastavení image__serveru.....	28
Obrázek 17 Nastavení MeasureServeru	29
Obrázek 18 Spuštění webového rozhraní pomocí adresy „localhost“	29
Obrázek 19 Princip základních funkcí v iSES Remduino Lab.....	30
Obrázek 20 Stažení Arduino IDE z webových stránek vývojářů.....	32
Obrázek 21 Instalace programu Arduino IDE.....	33
Obrázek 22 Instalace knihovny Remduino	34
Obrázek 23 Instalace ovladačů pro klonu Arduina	35
Obrázek 24 Nainstalovaný správný ovladač klonu Arduina.....	36
Obrázek 25 Nahrávání skriptu do Arduina.....	37
Obrázek 26 Vývojový diagram skriptu pro Arduino	38
Obrázek 27 Nastavení zvýrazňovačů v programu PSPad.....	41
Obrázek 28 Ukázka užití kaskádových stylů na zobrazení čísla úlohy.....	46
Obrázek 29 Vizuální podoba spouštěcího tlačítka	47
Obrázek 30 Schémata všech úloh vytvořená v softwaru Fritzing	49
Obrázek 31 Příprava úloh experimentu od návrhu v softwaru pro reálné sestavení... 49	
Obrázek 32 Repasovaný počítač využitý jako server.....	50
Obrázek 33 Klon Arduina Una [34].....	50
Obrázek 34 Hradla a BCD dekodér v provedení DIP14 a DIP16 [35]	51
Obrázek 35 LED dioda a sedmissegmentový LED panel [35].....	51
Obrázek 36 Rezistor [35]	51
Obrázek 37 Webová kamera Genius FaceCam 1000X.....	52
Obrázek 38 Servo motor [34]	52
Obrázek 39 Propojovací materiál [35].....	53
Obrázek 40 Design webové stránky vzdálené laboratoře.....	57

Obrázek 41 Výroba jednotlivých úloh.....	58
Obrázek 42 Výroba otočného systému pro webovou kameru.....	58
Obrázek 43 Výstavba vzdálené laboratoře	59
Obrázek 44 Lokální IP adresa nalezená přes příkazový řádek.....	60
Obrázek 45 Nastavení lokální IP adresu počítače v demilitarizované zóně	61

Seznam tabulek

Tabulka 1 Komplexní porovnání reálného a vzdáleného experimentu	62
Tabulka 2 Cenová kalkulace reálného experimentu na hodině	63
Tabulka 3 Cenová kalkulace pořizovacích nákladů vzdáleného experimentu.....	64

PŘÍLOHA

Laboratorní cvičení s hradly

Návod k vypracování vzdáleného experimentu

Ovládání vzdálené laboratoře s hradly zahrnuje přepínání úloh pomocí posuvníku, změnu logické hodnoty u čtyřech vstupů a vizuální odečítání logických hodnot na výstupech (viz obrázky níže). Označení LED diod je možné vyčíst ze schématu zapojení a odpovídá označení, které se nachází v pravdivostní tabulce zkoušeného logického členu. LED diody jsou zapojeny anodami k vstupům a výstupům hradel, takže jejich rozsvícení označuje logickou 1. Sedmisegmentový LED panel z 5. úlohy má opačné zapojení diod, proto se LED diody rozsvítí při logické 0.

Vzdálená laboratoř s hradly

Přepínání úloh

3

www.ises.info

5. XOR

6. NOT

Vstup A: On Off Stav: 1

Vstup B: On Off Stav: 0

Vstup C: On Off Stav: 0

Vstup D: On Off Stav: 0

Protokol k experimentu

ISES Remote Lab SDK Hands-on-Remote Laboratories

Kliknutím na šipku nebo tažením jezdce posuvníku lze přepínat úlohy

Podle rozsvícených LED diod lze doplnit pravdivostní tabulku

Stavy vstupů jsou označeny

Klikáním na tlačítka je možné měnit logické hodnoty vstupů

Kliknutím na odkaz dojde otevření protokolu v novém okně prohlížeče

Exkluzivní logický součet (XOR)

A (LED 2)	B (LED 1)	Y (LED 3)
0	0	0
0	1	1
1	0	0
1	1	1

fritzing

Poznámka: Vlivem přenosových možností internetu může u experimentu docházet ke zpoždění mezi vykonanou akcí a pozorovatelným výsledkem. Zpoždění může být v závislosti na připojení i v řádu sekund, buďte proto, prosím, trpěliví.

1. úloha

Na základě pozorování vzdáleného experimentu doplňte pravdivostní tabulky konjunktoru a Shefferovy funkce.

Schéma vnitřního zapojení integrovaného obvodu SN74LS08

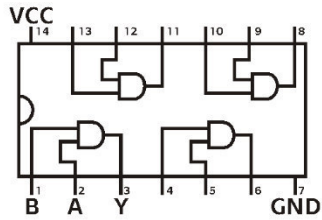
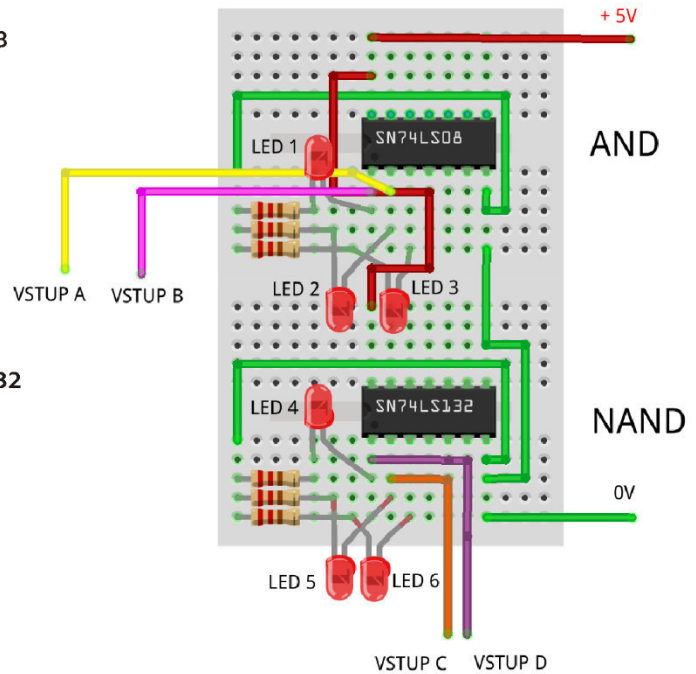
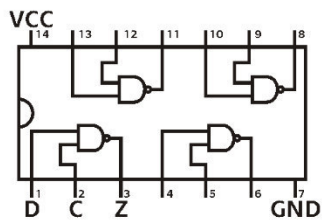


Schéma vnitřního zapojení integrovaného obvodu SN74LS132



fritzing

Konjunktor (AND)



A (LED 2)	B (LED 1)	Y (LED 3)
0	0	0
0	1	0
1	0	0
1	1	1

Shefferova funkce (NAND)



C (LED 5)	D (LED 4)	Z (LED 6)
0	0	1
0	1	1
1	0	1
1	1	0

2. úloha

Na základě pozorování vzdáleného experimentu doplňte pravdivostní tabulky disjunktory a Pierovy funkce.

Schéma vnitřního zapojení integrovaného obvodu SN74LS32

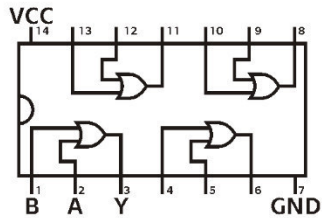
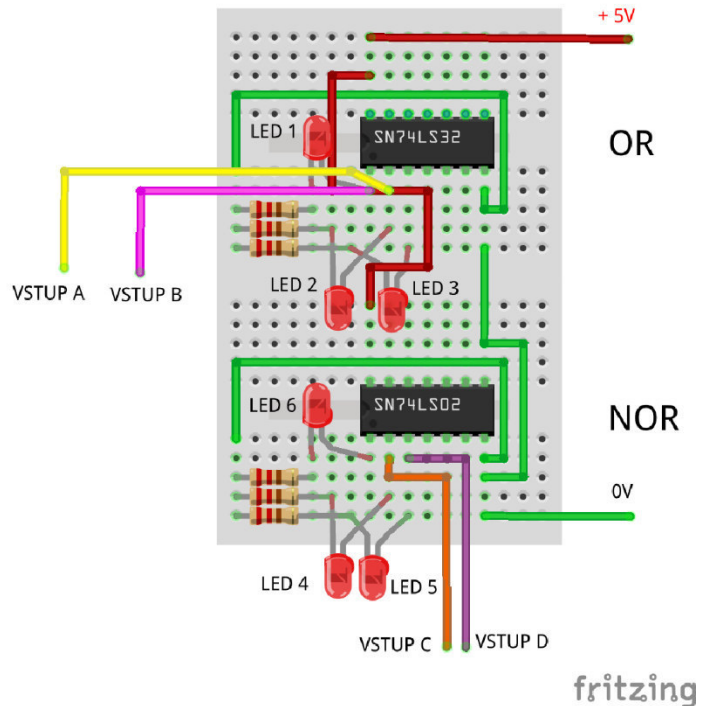
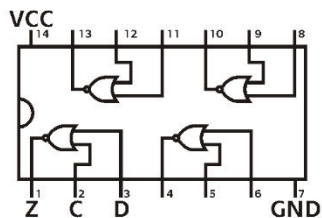


Schéma vnitřního zapojení integrovaného obvodu SN74LS02



Disjunktory (OR)



A (LED 2)	B (LED 1)	Y (LED 3)
0	0	0
0	1	1
1	0	1
1	1	1

Piereova funkce (NOR)



C (LED 4)	D (LED 5)	Z (LED 6)
0	0	1
0	1	0
1	0	0
1	1	0

3. úloha

Na základě pozorování vzdáleného experimentu doplňte pravdivostní tabulky exkluzivního logického součtu a invertoru. Pravdivostní tabulky negovaných funkcí doplňte na základě nabytých vědomostí z úloh 1-3.

Schéma vnitřního zapojení integrovaného obvodu SN74LS86A

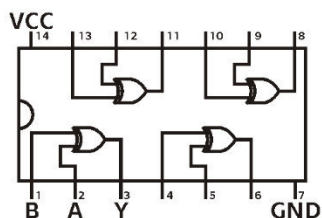
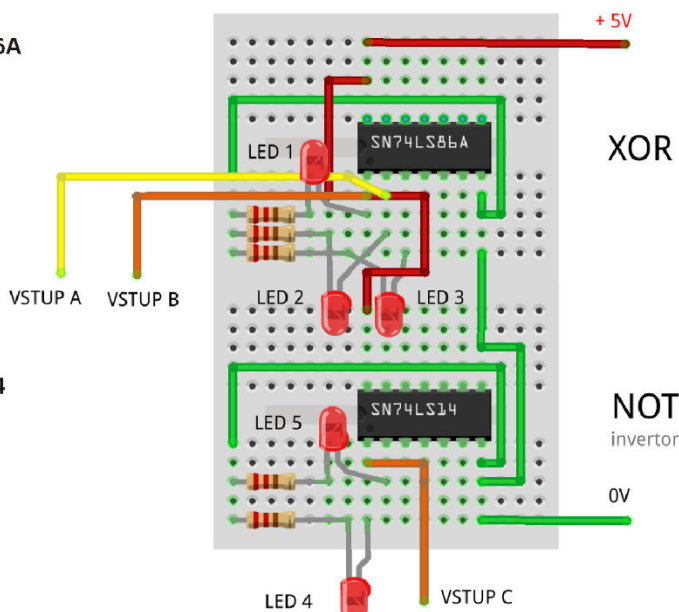
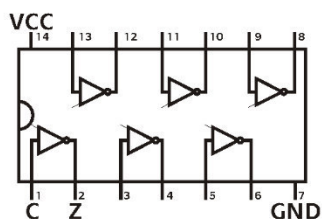


Schéma vnitřního zapojení integrovaného obvodu SN74LS14



fritzing

Exkluzivní logický součet (XOR)



A (LED 2)	B (LED 1)	Y (LED 3)
0	0	0
0	1	1
1	0	1
1	1	0

Negovaný XOR (XNOR)



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

Invertor (NOT)



C (LED 4)	Z (LED 5)
0	1
1	0

Repeater (opakovač)



C	Z
0	0
1	1

4. úloha

a) Na základě pozorování vzdáleného experimentu doplňte pravdivostní tabulku bistabilního klopného obvodu. Zaměřte při tom pozornost na pořadí zapínání a vypínání vstupů.

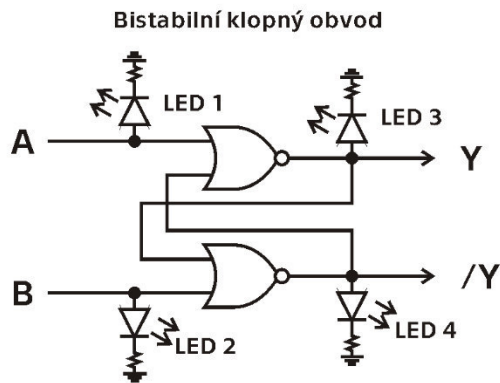
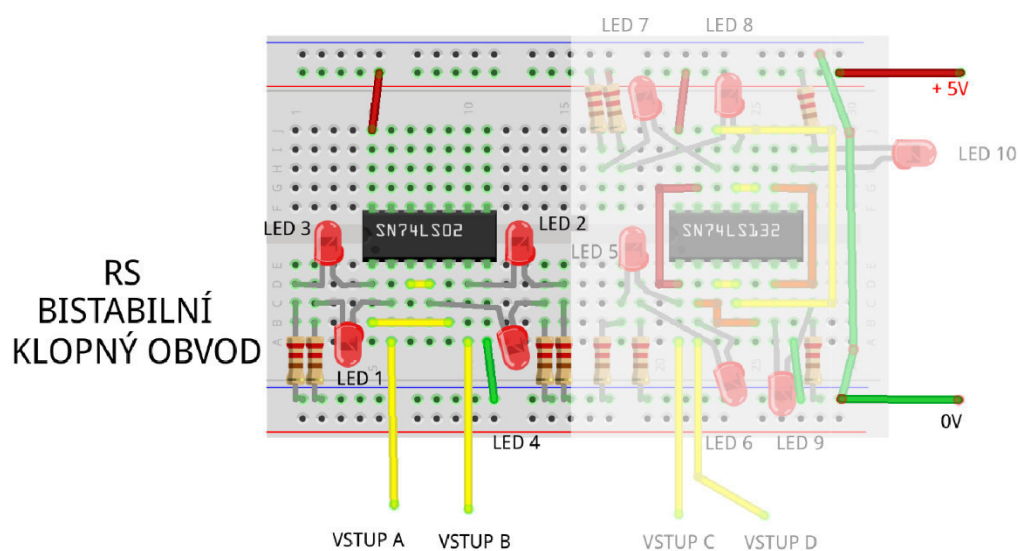
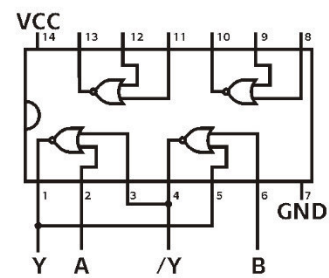


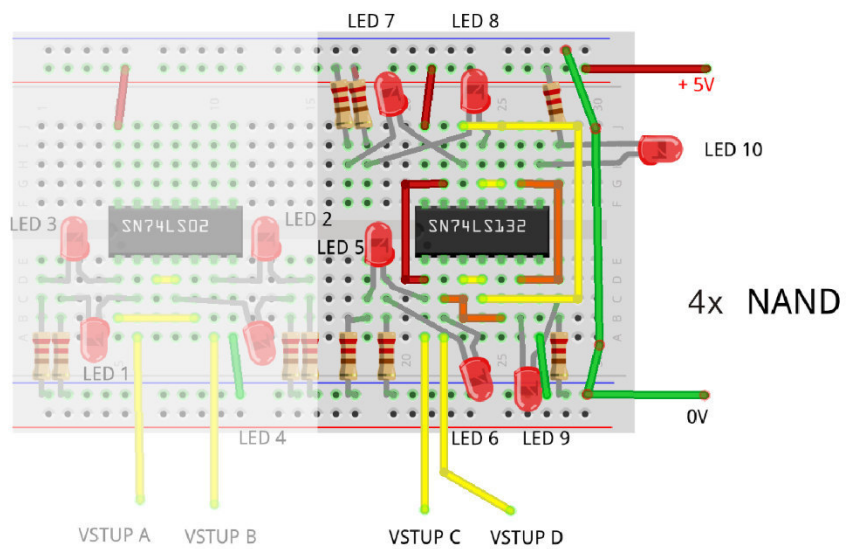
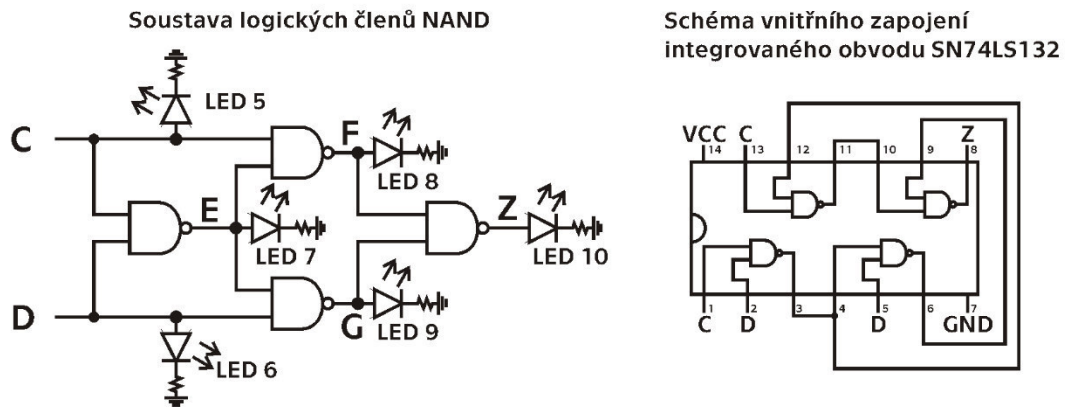
Schéma vnitřního zapojení integrovaného obvodu SN74LS02



fritzing

A (LED 1)	B (LED 2)	Y (LED 3)	/Y (LED 4)
0	0	Y_{n-1}	$/Y_{n-1}$
0	1	1	0
1	0	0	1
1	1	0	0

b) Na základě pozorování vzdáleného experimentu doplňte pravdivostní tabulku následujícího zapojení. O jakou funkci se jedná?



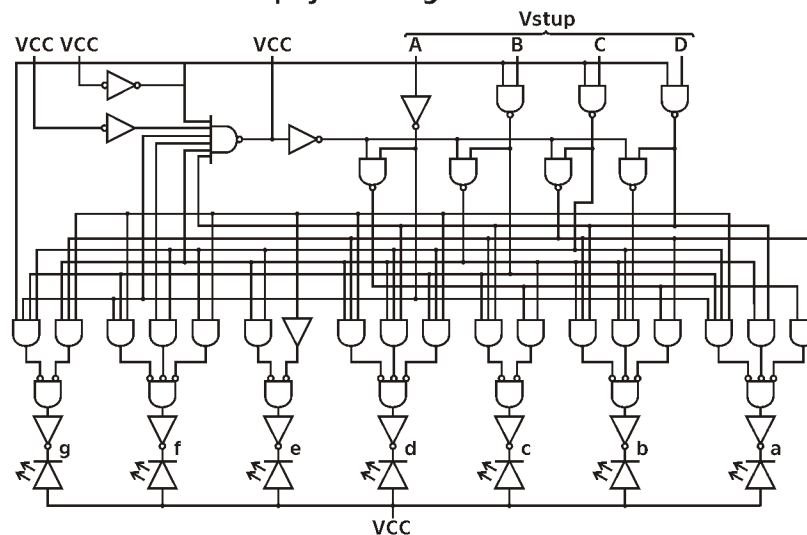
C (LED 5)	D (LED 6)	E (LED 7)	F (LED 8)	G (LED 9)	Z (LED 10)
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	0

Jedná se exkluzivní logický součet (XOR).

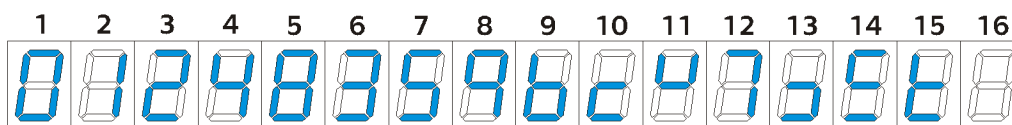
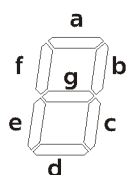
5. úloha

Na základě pozorování vzdáleného experimentu doplňte pravdivostní tabulku BCD_to_7-segment dekodéru (SN74LS47) a vyznačte rozsvícené diody u všech kombinací. Pozor, jedná se o sedmsegmentový panel se společnou anodou, tzn. že se LED segment rozsvítí, pokud dekodér na odpovídajícím pinu vyhodnotí logickou 0.

Schéma vnitřního zapojení integrovaného obvodu SN74LS47

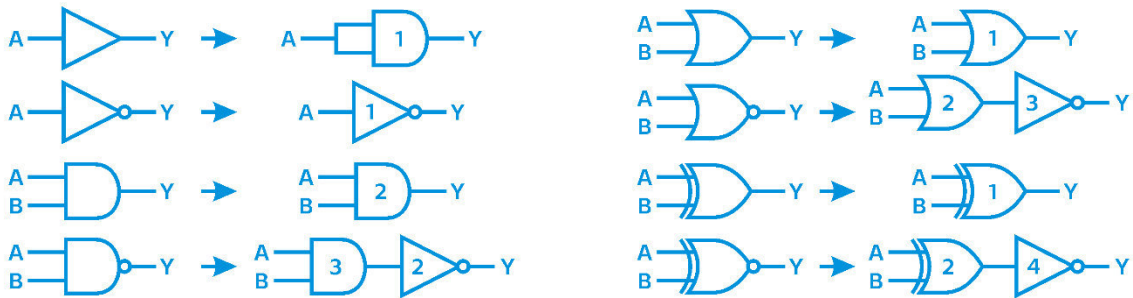


	A	B	C	D	a	b	c	d	e	f	g	Číslo
1	0	0	0	0	0	0	0	0	0	0	1	0
2	0	0	0	1	1	0	0	1	1	1	1	1
3	0	0	1	0	0	0	1	0	0	1	0	2
4	0	1	0	0	1	0	0	1	1	0	0	4
5	1	0	0	0	0	0	0	0	0	0	0	8
6	0	0	1	1	0	0	0	0	1	1	0	3
7	0	1	0	1	0	1	0	0	1	0	0	5
8	1	0	0	1	0	0	0	1	1	0	0	9
9	0	1	1	0	1	1	0	0	0	0	0	6
10	1	0	1	0	1	1	1	0	0	1	0	-
11	1	1	0	0	1	0	1	1	1	0	0	-
12	0	1	1	1	0	0	0	1	1	1	1	7
13	1	0	1	1	1	1	0	0	1	1	0	-
14	1	1	0	1	0	1	1	0	1	0	0	-
15	1	1	1	0	1	1	1	0	0	0	0	-
16	1	1	1	1	1	1	1	1	1	1	1	-



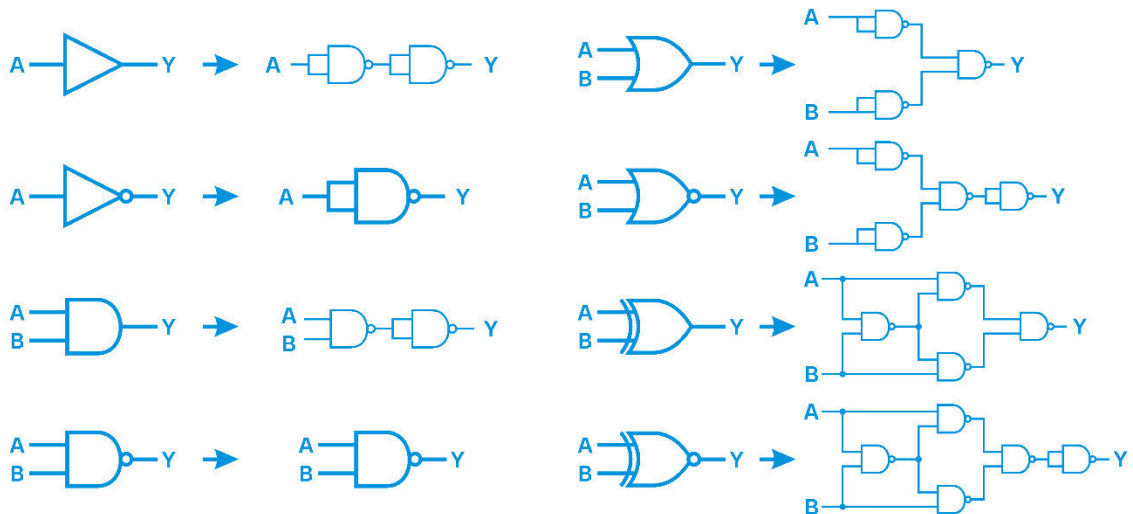
6. úloha - výzva

a) V prvních třech úlohách jsou zhotoveny pravdivostní tabulky 8 typů logických členů. Každý typ logického členu lze vytvořit pomocí jiných logických členů. Jak byste zapojili všechny zkoumané typy logických členů jen pomocí 4 součástek, přičemž každá součástka musí obsahovat jiný typ členu. Jedna součástka obsahuje více logických členů (6x invertor; 4x AND, 4x NAND, 4x OR, 4x NOR, 4x XOR nebo 4x XNOR).



AND (3x), Invertor (4x), OR (2x), XOR (2x)

b) Jak by vypadalo zapojení předchozí úlohy, kdyby byl požadavek na minimalizaci typů hradel? Kolik typů hradel a kolik součástek by bylo nutné využít?



Každé hradlo zvlášť => 22x NAND => 6x integrovaný obvod

Funkce a její negace dohromady => 13x NAND => 4x integrovaný obvod

c) Co by se stalo, kdyby se oba vstupy bistabilního klopného obvodu ze 4. úlohy naráz přepnuly z logické 1 na logickou 0?

Ideální stav: Výstupy začnou kmitat.

Realita: Náhodně se jeden z výstupů překlápí na 1 a druhý na 0.

Evidence výpůjček

Prohlášení:

Dávám svolení k půjčování této bakalářské práce. Uživatel potvrzuje svým podpisem, že bude tuto práci řádně citovat v seznamu použité literatury.

Jméno a příjmení: Pavel Žák

V Praze dne: 02. 05. 2019

Podpis:

Jméno	Oddělení/ Pracoviště	Datum	Podpis