

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

NÁVRH A REALIZACE PROGRAMU PRO MĚŘICÍ PŘÍSTROJ EXFO
FTB-5700

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

RADEK STRÍTESKÝ

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

NÁVRH A REALIZACE PROGRAMU PRO MĚŘICÍ PŘÍSTROJ EXFO FTB-5700

DESIGN AND REALISATION OF A PROGRAM FOR MEASURING EQUIPMENT EXFO FTB-5700

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

RADEK STŘÍTESKÝ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PETR MÜNSTER, Ph.D.

BRNO 2015



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Radek Stříteský

ID: 154876

Ročník: 3

Akademický rok: 2014/2015

NÁZEV TÉMATU:

Návrh a realizace programu pro měřicí přístroj EXFO FTB-5700

POKYNY PRO VYPRACOVÁNÍ:

Cílem bakalářské práce je návrh a realizace vlastního programu, který by umožňoval ovládání měřicího přístroje EXFO FTB-5700. Kromě automatického měření bude navržený software obsahovat i plánovač měření a modul pro vyhodnocení dlouhodobých měření.

DOPORUČENÁ LITERATURA:

[1] FILKA, M. Optoelektronika : Pro telekomunikace a informatiku. Vyd. 1. Brno : Centa, 2009. 369 s. ISBN 978-80-86785-14-1.

[2] COLLINGS, B., HEISMANN, F., LIETAERT, G. Reference Guides to Fiber Optic Testing, Vol. 2, JDSU : 2010, p. 146.

Termín zadání: 9.2.2015

Termín odevzdání: 2.6.2015

Vedoucí práce: Ing. Petr Münster, Ph.D.

Konzultanti bakalářské práce:

doc. Ing. Jiří Mišurec, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Hlavním cílem mé bakalářské práce je realizace programu pro automatizované měření disperze optických sítí. Práce v první části stručně vysvětluje optická vlákna, disperze a typy jejího měření. Je zde popsána chromatická (CD) a polarizační vidová (PMD) disperze. Dále je v práci popsáno programování pro Windows pomocí Windows 32 API. Jedná se zejména o popis vytvoření okna aplikace, práce klávesnice a myši ve Windows, zasílání zpráv proceduře okna a ovládací prvky ve Windows. V poslední části je popis realizace programu a podrobnější popis nejdůležitějších funkcí. Také se zde nachází výsledné grafy CD a PMD z testovacích automatizovaných měření.

KLÍČOVÁ SLOVA

optické vlákno, disperze, FTB 5700, Windows 32 API

ABSTRACT

Main goal of my Bachelor's thesis is to realize the program for the automatic measurement of optical networks. First part of this project describes the optical fiber, dispersion and types of measurement. There are described chromatic (CD) and polarization mode (PMD) of dispersion. There is described the programming for Windows by using Windows 32 API. Main part are description of creating window of application, work of keyboard and mouse in Windows, sending messages to window procedure and controlling parts in Windows. There is a description of the program and more detailed description of the most important functions in the last part. We can also find here the graphs CD and PMD from the testing automatic measurements.

KEYWORDS

optical fiber, dispersion, FTB 5700, Windows 32 API

STRÍTESKÝ, Radek *Návrh a realizace programu pro měřicí přístroj EXFO FTB-5700*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2015. 47 s. Vedoucí práce byl Ing. Petr Münster, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Návrh a realizace programu pro měřicí přístroj EXFO FTB-5700“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu práce panu Ing. Petrovi Münsterovi, Ph.D. za velmi užitečnou metodickou pomoc a cenné rady při zpracování bakalářské práce.

Brno

.....

(podpis autora)



Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
<http://www.six.feec.vutbr.cz>

PODĚKOVÁNÍ

Výzkum popsany v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno

.....

(podpis autora)



EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI



OBSAH

Úvod	11
1 Optická vlákna	12
1.1 Druhy optických vláken	12
1.1.1 Jednovidová	13
1.1.2 Mnohavidová s konstantním indexem lomu	13
1.1.3 Mnohavidová s gradientním indexem lomu	13
1.2 Disperze v optických vláknech	13
1.2.1 Chromatická disperze CD	14
1.2.2 Polarizační vidová disperze PMD	15
1.3 Měření disperze optických vláken	15
1.3.1 Měření chromatické disperze	15
1.3.2 Měření polarizační vidové disperze	16
2 Měřicí přístroj	18
2.1 Přístroj FTB 200	18
2.2 Modul FTB 5700	18
3 Metody automatizovaného ovládání programu	20
3.1 Ovládání pomocí klikání myši	20
3.2 Ovládání pomocí API	21
3.3 Přístup k programu pomocí Windows 32	21
3.3.1 Handle	21
3.3.2 Vlastní okno aplikace	21
3.3.3 Práce myši ve Windows	24
3.3.4 Práce klávesnice ve Windows	26
4 Praktická část	29
4.1 Popis metody řešení	29
4.1.1 Platforma měřicího přístroje FTB-200	29
4.1.2 Výběr metody řešení	29
4.1.3 Grafické prostředí programu	30
4.2 Realizace programu	30
4.2.1 Realizace Win32 Aplikace	30
4.2.2 Ovládání měřicího programu	34
4.3 Automatizované měření disperzí	38
4.3.1 Měření optické trasy	38
4.3.2 Výsledky a vyhodnocení	39

5 Závěr	42
Literatura	43
Seznam symbolů, veličin a zkratk	44
Seznam příloh	45
A Spuštění měření	46
B Obsah přiloženého CD	47

SEZNAM OBRÁZKŮ

1.1	Struktura optického vlákna	12
1.2	Průběh chromatické disperze	14
1.3	Metoda fázového posunu	15
1.4	Metoda zpoždění impulsů	16
1.5	Metoda interferometrická	17
2.1	Měřicí platforma FTB 200 (převzaté z www.profiber.eu)	18
2.2	Grafické rozhraní programu pro měření disperzí optických vláken . . .	19
3.1	Prostředí programu Auto Mouse Clicker	20
4.1	Vzhled programu	33
4.2	Hlavní funkce měření	35
4.3	Struktura oken standardního programu měření	36
4.4	CSV soubor s výpisem dat z HTML souborů	39
4.5	Zapojení měření s přístrojem EXFO FTB - 200 (EXFO převzaté z www.profiber.eu)	39
4.6	Chromatická disperze vlákna - smotané vlákno	40
4.7	Chromatická disperze vlákna - reálné vlákno	40
4.8	Polarizační vidová disperze - smotané vlákno	41
4.9	Polarizační vidová disperze - reálné vlákno	41

SEZNAM TABULEK

3.1	Identifikátory pro typy „handle“	21
3.2	Vysvětlení předpon konstant identifikátorů	22
3.3	Zprávy vyvolané kliknutím myši	25
3.4	Virtuální kódy klávesnice	28
3.5	Vysvětlení předpon konstant identifikátorů	28

ÚVOD

Cílem bakalářské práce je návrh a realizace programu, který by umožňoval automatické ovládání měřícího programu na přístroji EXFO FTB 200. Platforma FTB 200 s modulem FTB 5700 obsahuje program pro měření chromatické a polarizační vidové disperze optického vlákna. Standardní program na přístroji neumožňuje automatické ovládání a neobsahuje plánovač úloh. Plánovač úloh je vhodný pro několikanásobné měření disperze určitého optického vlákna v předem určených časových intervalech. Disperze, zvláště polarizační vidová disperze, je dynamicky se měnící hodnota vlivem okolního prostředí.

V první části bakalářské práce se čtenář seznámí s obecnou charakteristikou optického vlákna. Vysvětlení chromatické a polarizační vidové disperze a její vlivy na přenos optickým vláknem. Je zde uvedeno několik metod pro měření disperze.

Další část práce obsahuje popis měřícího přístroje s měřícím modulem. Je zde uveden stručný popis přístroje FTB 200 a modulu FTB 5700 pro měření chromatické a polarizační vidové disperze.

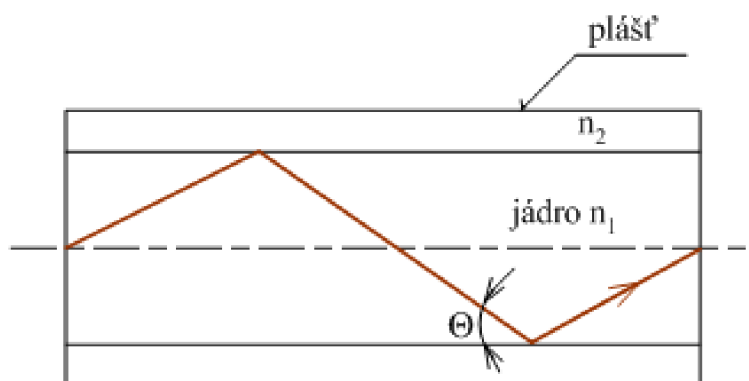
Dále je v bakalářské práci uvedeno několik metod řešení automatického ovládání programu. Jsou zde obecně popsány metody pro implementaci automatického ovládání měřícího přístroje. Podrobně tu je popsán přístup k programu pomocí Windows 32. Je zde popsáno okno aplikace ve Windows. Vysvětleny tu jsou zprávy myši a klávesnice určené pro určitou aplikaci.

Poslední a nejdůležitější částí práce je návrh a realizace programu, který ovládá měření. Nachází se zde popis vybrané metody ovládání měření, problémy s vytvořením grafického programu pro měřící přístroj, samotná realizace automatizovaného ovládání a testovací automatizované měření optických vláken.

1 OPTICKÁ VLÁKNA

Optické vlákno je dielektrická struktura většinou válcové symetrie. Přenos informace světlovodem umožňuje světelný paprsek. Nositelem signálu jsou neutrální fotony, které na sebe vzájemně nepůsobí. Při přenosu nevznikají žádná elektrická ani magnetická pole. Optický spoj je tvořen modulovaným zdrojem záření, optickým prostředím a přijímačem záření. Oblast optického záření sahá od 100 nm do 1 mm. Pro optický přenos informace má užší význam oblast mezi 0,4 až 1,7 μm . V této oblasti mají materiály používané pro výrobu vláken nejmenší útlum. Přenos informace optickým vláknem je založena na totálním odrazu na rozhraní dvou optických prostředí s rozdílným indexem lomu. Vlákno je tvořeno válcovým dielektrickým jádrem s indexem lomu n_1 a dielektrickým pláštěm s indexem lomu n_2 (obr. 1.1). Platí, že $n_1 > n_2$. U paprsků, které vstupují do jádra pod menším úhlem než θ podle rovnice 1.1, dochází na rozhraní jádro - plášť k totálnímu odrazu [3].

$$\cos \theta = \frac{n_2}{n_1} \quad (1.1)$$



Obr. 1.1: Struktura optického vlákna

1.1 Druhy optických vláken

Světlovody rozdělujeme podle materiálu a technologie přenosu. Nejčastěji se pro výrobu světlovodů používá křemenné sklo, které má nejmenší útlum. Podle technologie přenosu je dělíme na jednovidová a mnohavidová. Mnohavidová optická vlákna dále dělíme na vlákna s konstantním indexem lomu jádra a pláště a na vlákna gradientní s proměnným indexem lomu.

1.1.1 Jednovidová

Jednovidová vlákna mají malý průměr jádra a umožňují přenos pouze jediného (základního) vidu. U těchto vláken lze dosáhnout nižších hodnot útlumu. Malý průměr jádra je naopak nevýhodou pro navázání světla do vlákna. Nejčastěji se používají pro dálkové přenosy [3].

1.1.2 Mnohavidová s konstantním indexem lomu

Tyto vlákna se vyznačují jednoduchou výrobou a manipulací. Jejich nevýhodou je naopak větší útlum, disperze a malá přenosová kapacita. Vlákna tohoto typu jsou nejvíce používaná na krátké vzdálenosti [3].

1.1.3 Mnohavidová s gradientním indexem lomu

Vlákna mnohavidová s gradientním indexem lomu se vyznačují změnou indexu lomu v průřezu jádra. Maximální hodnota indexu lomu je v ose jádra a směrem od osy se zmenšuje. Vlákno s gradientním indexem lomu se vyznačuje menší disperzí, menším útlumem a složitější výrobou. Využívají se pro středně velké přenosy, s výhodou pro multiplexní přenosy [3].

1.2 Disperze v optických vláknech

Disperze patří k nejdůležitějším parametrům optických vláken, určuje šířku pásma a tím i přenosovou rychlost. Disperze vln v optických vláknech způsobuje zkreslení přenášeného signálu, které je definováno jako rozdíl šířky impulsu v polovině výšky na konci a na začátku vlákna.

Materiálová disperze je způsobena kmitočtovou závislostí na indexu lomu vlákna. Ve světlovodu se materiálová disperze kombinuje s disperzí vlnovodovou, která je ovlivněna geometrickými a materiálovými parametry světlovodu. Výsledný účinek materiálové a vlnovodové disperze bývá označován jako chromatická disperze.

Polarizační vidová disperze (PMD) je dalším faktorem ovlivňující kvalitu přenosu optickým vlákem. Hlavní příčinou PMD je asymetrie optického vlákna. Polarizační vidová disperze je proměnná v čase a závisí na celé řadě faktorů jako je použitý materiál, instalace kabelu a vnější vlivy.

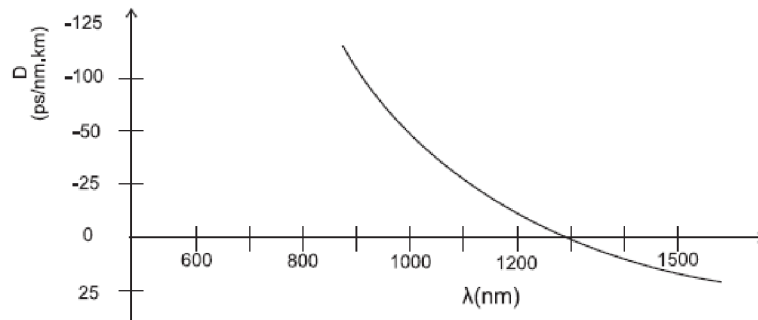
Dále existuje vidová disperze, která se vyskytuje pouze u mnohavidových vláken. Disperze je způsobena rozdílnou rychlostí šíření jednotlivých vidů [4].

1.2.1 Chromatická disperze CD

Se vzrůstající potřebou přenosových kapacit a používáním hustého vlnového multiplexu DWDM (Dense Wavelength Division Multiplex) bylo potřeba začít měřit chromatickou disperzi. Při DWDM se najednou šíří složky signálu s rozdílnou vlnovou délkou jinou rychlostí. Tím se nám budou některé složky vstupního impulsu signálu při průchodu optickým vláknem zpožďovat. Impuls se začne časově roztahovat a zdeformovaný impuls bude zasahovat do vedlejších bitových mezer a dojde ke zkreslení přenášené informace. Velikost disperze optického vlákna udává koeficient chromatické disperze D , definované vztahem 1.2 [4].

$$D(\lambda) = \frac{dt_g(\lambda)}{d(\lambda)}, \quad (1.2)$$

kde t_g je celkové zpoždění signálu při průchodu vláknem. Hodnota koeficientu udává rozšíření impulsu v ps, zdroje záření se spektrální pološířkou 1 nm, po průchodu vláknem 1 km. Obvyklý průběh chromatické disperze je uveden na obr. 1.2. Technologicky se dá připravit vlákno, aby se chromatická disperze v oblasti kolem 1,55 μm blížila k nule. Tyto případy vláken DS (Dispersion Shifted) se dělí na vlákna s posunutou disperzní charakteristikou a plochou disperzní charakteristikou [4].



Obr. 1.2: Průběh chromatické disperze

Se stále používanějším DWDM nastal problém u starších položených vláken, jak kompenzovat chromatickou disperzi. Nejpoužívanějším řešením je použít DCF (Disperzion Compensation Fiber) s vysokou hodnotou záporné chromatické disperze. Kompenzace trasy se provádí napojením tohoto vlákna na konci trasy [10].

1.2.2 Polarizační vidová disperze PMD

PMD (Polarization Mode Dispersion) je zkreslení jednotlivého impulsu procházejícího ve dvou na sebe kolmých polarizačních rovinách. Tento jev je způsobený kruhovou nesymetrií optického vlákna, která vzniká při montáži nebo výrobě vlákna. Polarizační vidová disperze na krátkých vzdálenostech přibližně do 10 km je definovaná vztahem 1.3 [4].

$$PMD = \frac{\Delta\tau}{l}, \quad (1.3)$$

kde l - délka trasy a $\Delta\tau$ - zpoždění signálu. Pro větší vzdálenosti se šíří nelineárně, definované vztahem 1.4 [4].

$$PMD = \frac{\Delta\tau}{\sqrt{l}}, \quad (1.4)$$

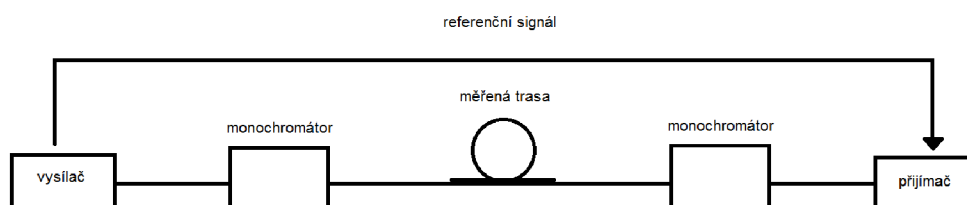
1.3 Měření disperze optických vláken

1.3.1 Měření chromatické disperze

Metoda fázového posuvu a diferenciálního posuvu

Metoda fázového posuvu je uvedena jako referenční metoda pro měření chromatické disperze. Schéma metody je uvedeno na obr. 1.3. Metoda spočívá v porovnávání změřené fáze na konci trasy s fází vstupního signálu. Informace o vstupní fázi je k detektoru přenášena pomocí referenční trasy (např. jiné vlákno v kabelu). K měření se používá modulovaný zdroj záření na několika vlnových délkách. Jako detektor je použit například vektorvoltmetr.[5]

V praxi se více využívá metoda diferenciálního fázového posuvu. Metoda je založena na měření rozdílu fází signálu na různých vlnových délkách. Referenční trasa odpadá, přesto se používají dvě vlákna. Jedno jako měřené a druhé jako komunikační, po kterém ovládáme vysílač.[5]



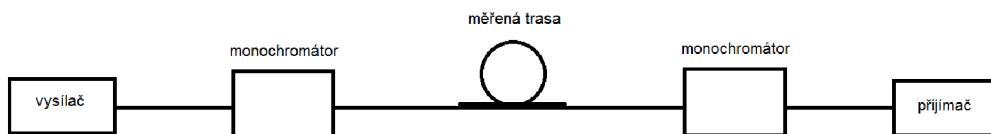
Obr. 1.3: Metoda fázového posuvu

Metoda interferometrická

Metoda vhodná pro měření krátkých úseků vláken o celkové chromatické disperzi nejvýše jednotek ps, vyžívá se především pro laboratorní měření. Interferometrická metoda je založena na interferenci mezi zářením z měřeného a referenčního vlákna, pomocí které se zjišťuje fázový rozdíl obou signálů. [5]

Metoda zpoždění impulsů v časové oblasti

Do trasy jsou poslány impulzy o různých vlnových délkách v přesných rozestupech. Zpoždění vlivem chromatické disperze se určuje z porovnání rozestupů impulsů na vstupu a na výstupu vlákna. Schéma metody je uvedeno na obr 1.4. Přesnost měření je dána měřicí soustavou a samotnou chromatickou disperzí. Impulzy se navzájem vzdalují a hlavně roztahují, což způsobuje problematičtější stanovení polohy impulsů. [5]



Obr. 1.4: Metoda zpoždění impulsů

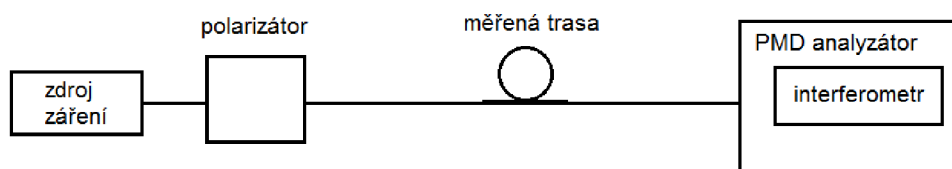
1.3.2 Měření polarizační vidové disperze

Polarizační vidová disperze se začala měřit až při nástupu vyšších přenosových rychlostí (od 10 Gbit/s). Jedná se o jev výrazně slabší v porovnání s chromatickou disperzí. Naopak je to jev zcela náhodný, který závisí i na montáži a vlivu okolního prostředí. Jev proto nejde správně spočítat a musí se měřit od stádia výroby až po uvedení optické trasy do provozu. Nutnost měření platí především pro starší položená vlákna. Polarizační vidovou disperzi jde velmi obtížně, draze a ne zcela správně kompenzovat.

Metoda interferometrická

Tato metoda je založena na interferenci nízkokoherentního optického záření. Tento signál vstupuje přes polarizátor do měřené optické trasy. Na konci trasy je použit Michelsonův interferometr. Metoda je zobrazena na obr. 1.5. Interferometr se skládá z polarizátoru, dvou zrcadel (jedno pohyblivé), interferometru a detektoru. Optické

záření se v interferometru rozdělí do dvou větví. Jedna větev je zakončena pevným zrcadlem a druhá pohyblivým zrcadlem, které mění fázový posun signálů. Odraženým signálem pomocí interference na detektoru zjišťujeme polarizační vidovou disperzi. Metoda je rychlá, odolná vůči vibracím a umožňuje měřit PMD od 0,1 ps až 100 ps. [5]



Obr. 1.5: Metoda interferometrická

Metoda P-OTDR

Metoda P-OTDR (Polarisation OTDR) využívá metody zpětného rozptylu OTDR. Metoda je založena na vysílání signálu do optického vlákna a ze zpětného rozptýleného záření vlivem Rayleighova zpětného rozptylu. Pro analýzu PMD potřebujeme informace o lokálním dvojlomu a vazební délce. Použijeme metodu skenování vlnové délky nebo metodu stupně polarizace DOP. [6]

2 MĚŘÍCÍ PŘÍSTROJ

Měřicí univerzální platforma FTB 200 je produktem firmy EXFO. EXFO poskytuje servis pro provozovatele bezdrátových a pevných sítí a výrobců zařízení v telekomunikačním průmyslu. Firma nabízí inovační řešení pro správu a údržbu telekomunikačních sítí.

2.1 Přístroj FTB 200

Platforma FTB 200 poskytuje testování optických a ethernetových sítí. Umožňuje vložení více jak 30 modulů pro různé testování sítí. Platforma poskytuje vysoký výpočetní výkon pro pokročilé testovací aplikace. Měřicí přístroj umožňuje přenos dat do cloudu nebo do připojeného externího úložiště [1].

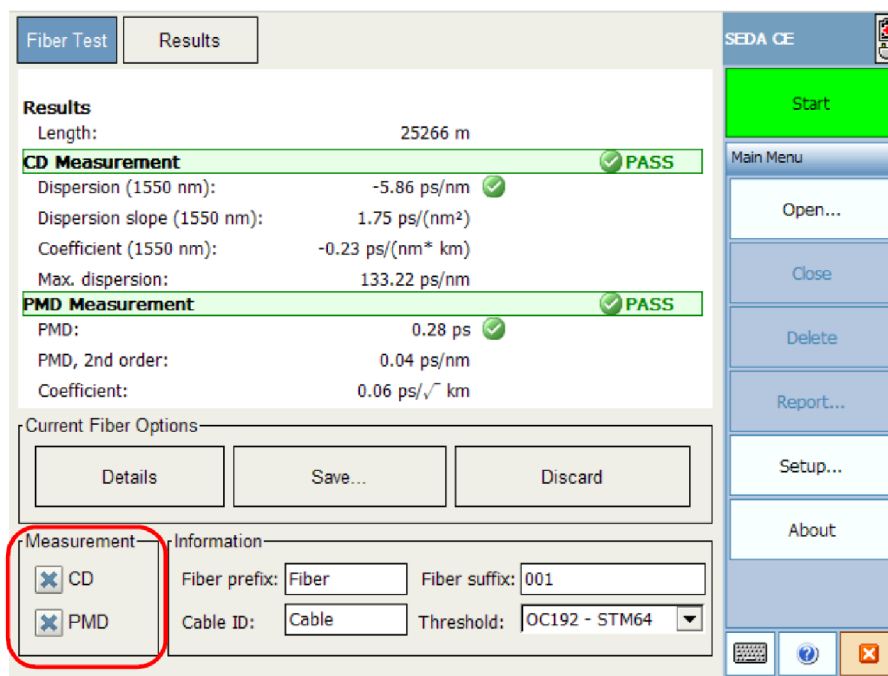


Obr. 2.1: Měřicí platforma FTB 200 (převzaté z www.profiber.eu)

2.2 Modul FTB 5700

Modul FTB umožňuje měření chromatické a vidové polarizační disperze. Kombinace CD a PMD do jednoho usnadňuje technikům práci a čas. FTB 5700 je speciálně navržen pro dnešní vysokorychlostní sítě a umožňuje plně automatizované měření s vyhodnocením naměřených hodnot. Nevýhodou je absence plánovače úloh pro

opakované měření optické trasy. Přístroj vykonává měření z jednoho konce vlákna s využitím Fresnelovu odrazu na konci vlákna. Modul umožňuje měřit vlákna do délky 120 km. Na obr. 2.2 je znázornění prostředí aplikace pro měření CD a PMD [2].



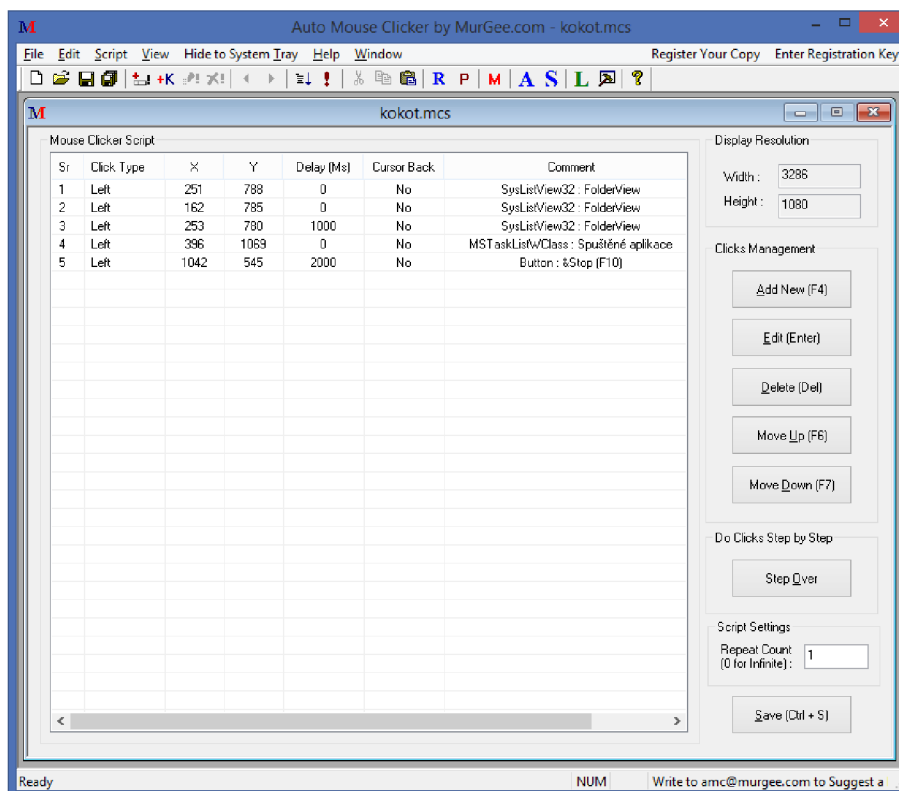
Obr. 2.2: Grafické rozhraní programu pro měření disperzí optických vláken

3 METODY AUTOMATIZOVANÉHO OVLÁDÁNÍ PROGRAMU

Automatizované ovládání programu lze řešit několika způsoby. Níže je popis metod a jejich výhody a nevýhody.

3.1 Ovládání pomocí klikání myši

Na automatizované ovládání myši existuje mnoho volně dostupných programů (např. Auto Mouse Clicker). Většina těchto programů umožňuje automatické nahrávání událostí myši a klávesnice nebo ruční zápis posloupnosti událostí. Volitelně se znamenávají tyto události: kliknutí myši, pohyb myši, zpoždění mezi jednotlivými událostmi a vstupy z klávesnice. Prostředí programu Auto Mouse Clicker se záznamem událostí je na obr. 3.1. Na klávesové zkratky aplikace pro měření CD a PMD nereaguje. Hlavní nevýhodou automatizovaného klikání je nemožnost dynamicky měnit posloupnost událostí při měření. V další části je nastíněno několik metod pro automatické ovládání programu.



Obr. 3.1: Prostředí programu Auto Mouse Clicker

3.2 Ovládání pomocí API

API (Application Programming Interface) označuje rozhraní pro programování aplikací. Jde o sbírku procedur, funkcí a tříd, které může programátor používat. Pomocí API lze například přistupovat k operačnímu systému, běžící aplikaci, připojenému zařízení a webové aplikaci. API nám umožňuje snáze a rychleji komunikovat a vyměňovat data s aplikacemi druhých stran.

3.3 Přístup k programu pomocí Windows 32

Operační systém je definován svým API. Obsahuje všechna volání funkcí, které může aplikace běžící na operačním systému žádat. Obecně platí, že Windows API zůstává konzistentní od Windows 1.0. Jedinými změnami jsou rozšíření a přechod z 16bitové na 32bitovou (64bitovou) architekturu procesorů. Windows API definuje mnoho funkcí pro grafiku (GDI), zvuk, ovládací zařízení, síť a mnoho dalších.

3.3.1 Handle

Ve Windows je handle ukazatel na ukazatel, ukazuje na adresu uloženou v tabulce Windows. Tento způsob adresování je ve Windows nezbytný, protože správce paměti často přesouvá objekty v paměti. Pokud by aplikace pracovala pouze s ukazateli pro přístup k objektům, výsledkem by byly neplatné ukazatele. Z toho důvodu Windows používá handle. Jedná se jednoduše o číslo, které odkazuje na objekt. Skutečná hodnota handle je pro nás nevýznamná, avšak modul Windows přesně ví, jak jej použít. Handle jsou ve Windows použity velmi často, hodně funkcí vrací handle a mnoho dalších funkcí použije handle jako vstupní parametr. Rozlišujeme tři hlavní identifikátory handlu popsané v tab 3.1. Dále se můžeme setkat s HICON (handle ikony), HCURSOR (handle kurzoru myši) a HBRUSH (handle grafického štětce) [9].

Tab. 3.1: Identifikátory pro typy „handle“

Identifikátor	Význam
HINSTANCE	Handle „instance“ - tedy programu samotného
HWND	Handle okna
HDC	Handl kontextu zařízení

3.3.2 Vlastní okno aplikace

Nejnápadnější okna ve Windows jsou okna aplikací, která mohou obsahovat záhlaví, panel nástrojů, posuvníky, seznamy, tlačítka, textová pole a jiné. Každý z těchto

objektů je oknem [8].

Názvy identifikátorů

Názvy identifikátorů jsou psány velkými písmeny a jsou definovány v hlavičkových souborech Windows. Začínají dvou nebo třípísmennou předponou, za kterou následuje podtržítka. Vysvětlení významu předpon vysvětluje tab. 3.2. Téměř každá číselná konstanta ve Windows má identifikátor [8].

Tab. 3.2: Vysvětlení předpon konstant identifikátorů

Předpona	Konstanta
CS	Volba stylu třídy
CW	Možnosti pro vytváření okna
DT	Možnosti pro psaní textu
IDI	Identifikační číslo (ID) ikony
IDC	Identifikační číslo (ID) kurzoru
MB	Možnosti dialogu se zprávou
SND	Možnosti zvuku
CWM	Zprávy Windows
WS	Styl okna

Registrace třídy okna

Vytvořené okno je vždy založeno na třídě okna. Třída okna určuje proceduru, která zpracovává zprávy určené oknu. Na jedné třídě může být založeno více prvků. Při vytvoření okna se definují další charakteristiky oken, které jsou pro okno jedinečné. [7]

Vytvoření okna

Třída okna definuje obecné charakteristiky okna, funkcí *CreateWindow* zadáme podrobnější informace o oknu. Informace předávané funkcí *RegisterClass* jsou zadány v datové struktuře, informace předané *CreateWindow* jsou zadány jako samostatné parametry funkce. Zde je volání funkce: [7]

```
hwnd = CreateWindow (szAppName,           //název třídy okna
                    TEXT ("Pokus"),       //záhlaví okna
                    WS_OVERLAPPEDWINDOW, //styl okna
                    CW_USEDEFAULT,       //výchozí pozice x
                    CW_USEDEFAULT,       //výchozí pozice y
                    CW_USEDEFAULT,       //výchozí velikost x
                    CW_USEDEFAULT,       //výchozí velikost y
```

```

NULL,           //handle rodičovského okna
NULL,           //handle nabídky okna
hInstance,     //handle instance programu
NULL);         //parametry pro vytvoření

```

Zobrazení okna

Po dokončení funkce *CreateWindow* vznikne ve Windows nové okno. Tím Windows alokoval část paměti pro uložení informace o okně. Později tuto paměť najde pomocí handle okna. Pro zobrazení okna na obrazovce je potřeba vyvolat další dvě funkce. První z nich je *ShowWindows*. Prvním parametrem je handle vytvořeného okna, druhým parametrem potom hodnota *iCmdShow*. Druhý parametr určuje typ zobrazení okna, tedy zda normálně, minimalizovaně nebo maximilizovaně. Druhou funkcí je *UpdateWindow*, která okno na displeji zobrazí [7].

Smyčka zpráv

Po zobrazení okna na displeji se musí program připravit na čtení vstupu z klávesnice a myši od uživatele. Windows pro každý program udržuje frontu zpráv. Když uživatel vyvolá událost klávesnicí nebo myší, Windows tuto událost převede na zprávu a uloží do fronty zpráv programu. Program následně vybírá zprávy z fronty následovně:

```

While (GetMessage (&msg, NULL, 0, 0))
{
    TranslateMessage (&msg);
    DispatchMessage (&msg);
}

```

Proměná *msg* je struktura typu *MSG*, která vypadá následovně:

```

typedef struct tagMSG
{
    HWND    hwnd;
    UINT    message;
    WPARAM  wParam;
    LPARAM  lParam;
    DWORD   time;
    POINT   pt;
}

```

Voláním funkce *GetMesseage* vyzvedneme zprávu ze smyčky zpráv. Parametry krom prvního jsou nastaveny na *NULL* nebo *0*, což značí, že program vyžaduje všechny zprávy pro všechna okna vytvořená programem. [7]

Struktura zprávy *MSG* je následující:

- *hwnd* - Handle okna, kterému je zpráva směřována.
- *message* - Identifikátor zprávy, který číselně označuje zprávu.

- *wParam* - 32bitový parametr zprávy, význam a hodnota závisí na typu zprávy.
- *lParam* - 32bitový parametr, který taky závisí na konkrétní zprávě.
- *time* - Čas, kdy byla zpráva umístěna do fronty zpráv.
- *pt* - Souřadnice myši v čase, kdy byla zpráva umístěna do fronty zpráv.

Procedura okna

Registrace okna, vytvoření okna, zobrazení okna a vyzvedávání zpráv byla jen režié. Skutečná činnost programu začíná až v proceduře okna. Procedura okna určuje, jak bude okno reagovat na vstup od uživatele. Vždy je přiřazena určité třídě okna, kterou voláme funkcí *RegisterClass*. Procedura okna má následující tvar:

```
LRESULT CALLBACK WndProc (HWND hwnd, UINT message, WPARAM wParam, LPARAM lParam)
```

Parametry funkce jsou shodné s prvními čtyřmi položkami ve struktuře *MSG*. Pokud program vytvoří více oken založených na jedné třídě, *hwnd* udává aktuální okno, které má obdržet zprávu. Procedury okna nevolají programy přímo, jsou volány samostatně z Windows. Program může nepřímo volat procedury okna pomocí funkce *SendMessage*. [7]

3.3.3 Práce myši ve Windows

Když uživatel Windows pohne myší, Windows přesune na displeji malou bitmapu, takzvaný kurzor. Přesné místo na obrazovce určuje „kurzor myši“. Kurzor má jeden bod zvaný jako „horký bod - hot spot“. Tento bod určuje přesnou polohu na obrazovce. Windows podporuje několik druhů kurzorů. Nejpoužívanější kurzorem je skloněná šipka (*IDC_ARROW*), který má horký bod na špičce šipky. Dále se často používají kurzory typu kříž (*IDC_KROS*) a přesýpací hodiny (*IDC_WAIT*). Programátoři aplikací si také mohou navrhnout svoje vlastní kurzory myši. Výchozí kurzor pro určité okno je zadán při definici struktury třídy okna, například: [7]

```
wndclass.hCursor = LoadCursor (NULL, IDC_ARROW) ;
```

Pomocí tlačítek myši můžeme provádět následující činnosti:

- *Clicking* - Stisknutí a uvolnění tlačítka.
- *Double-clicking* - Stisknutí a uvolnění tlačítka myši dvakrát za sebou.
- *Dragging* - Pohyb myši při současném přidržení tlačítka.

Zprávy myši

Procedura okna dostává zprávy od myši, když myš přejede přes okno nebo uživatel uvnitř okna klikne. Zprávy jsou předány, i když okno není aktivní nebo nemá fokus pro vstup. Přesunutí myši v klientské oblasti okna vyvolá zprávu *WM_MOUSE_MOVE* pro proceduru okna. Když je kliknutí myši provedeno uvnitř klientské oblasti okna, procedura okna dostane zprávy uvedené v tab. 3.3 [7].

Tab. 3.3: Zprávy vyvolané kliknutím myši

Tlačítko	Při stisku	Při uvolnění	Při druhém stisku
Levé	WM_LBUTTONDOWN	WM_LBUTTONUP	WM_LBUTTONDBLCLK
Prostřední	WM_MBUTTONDOWN	WM_MBUTTONUP	WM_MBUTTONDBLCLK
Pravé	WM_RBUTTONDOWN	WM_RBUTTONUP	WM_RBUTTONDBLCLK

U těchto zpráv obsahuje *lParam* polohu myši. Souřadnice x se nachází ve spodním slově parametru, naopak souřadnice y v horním slově parametru. Souřadnice myši vzhledem k hornímu rohu okna můžeme vyčíst pomocí následujících maker:

```
x = LOWORD (lParam) ;
y = HIWORD (lParam) ;
```

Hodnoty *wParam* určují stav kláves *Shift* a *Ctrl* a tlačítek myši.

```
MK_LBUTTON      Levé tlačítko je stisknuto.
MK_MBUTTON      Prostřední tlačítko je stisknuto.
MK_RBUTTON      Pravé tlačítko je stisknuto.
MK_SHIFT        Klávesa Shift je stisknuta.
MK_CONTROL      Klávesa CTRL je stisknuta.
```

S klikáním myši mohou nastat zvláštní situace. Například procedura okna dostane zprávu *WM_LBUTTONUP*, aniž by dříve dostala zprávu *WM_LBUTTONDOWN*. K tomu dojde v případě přesunutí stisknutého tlačítka a uvolnění v našem sledovaném okně. Naopak může procedura okna dostat zprávu *WM_LBUTTONUP* bez předchozí zprávy *WM_LBUTTONDOWN*. [7]

Existují dvě výjimky z uvedených příkladů:

- Procedura okna může myš uvěznit v okně. Díky uvěznění myši dostává procedura okna zprávy dále i po přesunutí myši z klientské oblasti okna.
- Když se nachází na obrazovce modální systémové okno ¹. nebo modální systémové okno se zprávou.

Dvojklik myši

Dvojklik myši můžeme definovat jako dvě kliknutí myši v blízkém prostoru a určitém intervalu. Blízký prostor můžeme definovat následovně: široký jako průměrná šířka znaku v textu a vysoký jako polovina výšky znaku. Interval mezi kliknutími se nazývá „rychlost dvojitého kliknutí“, můžeme ji měnit v ovládacích panelech Windows.

Aby procedura okna dostávala zprávy poklepání myši, musíme zadat identifikátor *CS_DBLCLKS* ve struktuře třídy okna před voláním funkce *RegisterClass* [7].

```
wndclass.style = CS_HREDRAW | CS_VREDRAW | CS_DBLCLKS) ;
```

¹Modální systémová okna zamezují uživateli přepnout se do jiného okna.

Při nezadání identifikátoru *CS_DBLCLKS* do struktury třídy dostane procedura okna následující zprávy:

```
WM_LBUTTONDOWN  
WM_LBUTTONUP  
WM_LBUTTONDOWN  
WM_LBUTTONUP
```

Po přidání stylu *CS_DBLCLKS* do struktury třídy, dostane procedura následující zprávy:

```
WM_LBUTTONDOWN  
WM_LBUTTONUP  
WM_LBUTTONDOWNBLCLK  
WM_LBUTTONUP
```

Zprávy po dvojím kliknutí myši se zpracovávají snadněji, pokud první kliknutí provádí dílčí část akcí, které provádí dvojklik myši.

3.3.4 Práce klávesnice ve Windows

Klávesnice má dlouhou historii počínaje Remingtonovým psacím strojem z roku 1874. První programátoři používali klávesnici pro děrování děrných štítků a později k ovládání terminálu pro ovládání velkých sálových počítačů. V dnešní době se klávesnice značně liší, obsahují funkční klávesy, klávesy pro pohyb kurzoru a obvykle samostatnou číselnou klávesnici [7].

Vstup z klávesnice je předáván pomocí zpráv do procedury okna aplikace Windows. V operačním systému Windows existuje celkem 8 různých zpráv, přibližně polovinu z nich reálně využíváme při programování aplikací pro Windows. Můžeme například ignorovat stisk kláves, které ovládají systémové funkce. Systémové funkce většinou zahrnují klávesu *Alt* [7].

Fokus okna

Klávesnice u osobních počítačů musí být sdílena mezi všemi běžícími aplikacemi ve Windows. Klávesnice musí být sdílena také mezi všemi okny určité aplikace.

Program pro vyzvedávání zpráv z fronty zpráv využívá struktura *MSG*. Ve struktuře položka *hwnd* určuje *handle* okna, které má dostat zprávu. Pomocí funkce *patchMessage* ve smyčce zpráv jsou zprávy zasílány proceduře určitého okna.

Zpráva přijde proceduře okna, které má právě vstupní fokus. Okno se vstupním fokusem může být aktivní okno nebo potomek okna (dceřiné okno) nebo dceřiné okno aktivního dceřiného okna a tak dále. Aktivní okno je vždy na nejvyšší úrovni (*handle* jeho rodičovského okna je *NULL*). Dceřiné okno může mít vstupní fokus

pouze pokud je potomkem aktivního okna. Vstupní fokus dceřiných oken poznáme pomocí blikajícího kurzoru nebo tečkované čáry po obvodu.

Pomocí zachytávání zpráv `WM_SETFOCUS` a `WM_KILLFOCUS` procedura okna dokáže určit, zda její okno má vstupní fokus. Zpráva `WM_SETFOCUS` oznamuje, že dostává vstupní fokus a zpráva `WM_KILLFOCUS` oznamuje, že vstupní fokus ztrácí [7].

Zprávy vyvolané stiskem kláves

Po stisku klávesy uloží Windows zprávu `WM_KEYDOWN` nebo `WM_SYSKEYDOWN` do fronty zpráv okna, které má právě vstupní fokus. Po uvolnění klávesy uloží Windows zprávu `WM_KEYUP` nebo `WM_SYSKEYUP` do fronty zpráv okna. Zprávy obvykle přicházejí v párech. Při stisknutí klávese posílají Windows posloupnost zpráv `WM_KEYDOWN` (`WM_SYSKEYDOWN`) a po uvolnění klávesy jednu zprávu `WM_KEYUP` (`WM_SYSKEYUP`). Čas, kdy byla klávesa stisknuta, můžeme určit voláním funkce `GetMessageTime` [7].

Systémové zprávy se týkají stisků kláves, které jsou nejdůležitější pro samotný Windows. Zprávy `WM_SYSKEYDOWN` a `WM_SYSKEYUP` se převážně generují pro klávesy stisknuté převážně v kombinaci s klávesou *Alt*. Tyto zprávy se starají například přepínání aktivního okna (*ALT + TAB*) nebo ukončení aplikace (*Alt + F4*) [7].

Virtuální kódy kláves

Virtuální kód klávesy je uložen v parametru *wParam* u zpráv `WM_KEYDOWN`, `WM_SYSKEYDOWN`, `WM_KEYUP` a `WM_SYSKEYUP`. Virtuální kód určuje, která klávesa byla stisknuta nebo uvolněna, mají nejčastěji názvy začínající na *VK_*. Kódy jsou definované v hlavičkovém souboru `WINUSER.H`. Tabulka 3.4 ukazuje názvy s číselnými údaji a klávesu z kompatibilní klávesnice IBM, která tomuto kódu odpovídá. Tabulka dále ukazuje, zda klávesa je potřeba, aby Windows fungoval správně.

V tab 3.5 jsou nejpoužívanější virtuální kódy kláves, jedná se o virtuální kódy kláves s písmeny a číslicemi na hlavní klávesnici [7].

Tab. 3.4: Virtuální kódy klávesnice

Desítková soustava	Hex	Identifikátor	je potřeba?	IBM-kompatibilní klávesnice
8	08	VK_BACK	•	Klávesa Backpace
9	09	VK_TAB	•	Klávesa Tab
12	0C	VK_CLEAR		Klávesa 5 na číselné klávesnici s vypnutým NumLock
13	0D	VK_RETURN	•	Klávesa Enter
16	10	VK_SHIFT	•	Klávesa Shift
17	11	VK_CONTROL	•	Klávesa Ctrl
18	12	VK_MENU	•	Klávesa Alt
20	14	VK_CAPITAL	•	Klávesa CapsLock
27	1B	VK_ESCAPE	•	Klávesa Esc
32	20	VK_SPACE	•	Mezerník

Tab. 3.5: Vysvětlení předpon konstant identifikátorů

Desítková soustava	Hex	Identifikátor	je potřeba?	IBM-kompatibilní klávesnice
48_57	30_39	Žádný	•	0 až 9 na hlavní klávesnici
65_90	41_5A	Žádný	•	A až Z

4 PRAKTICKÁ ČÁST

Praktická část bakalářské práce se zabývá automatizovaným ovládáním měřícího přístroje EXFO FTB - 200. Měřící přístroj měří chromatickou s polarizační vidovou disperzí pomocí dodávaného programu výrobcem, který umožňuje vždy pustit pouze jedno měření disperzí. Měření disperzí je vhodné měřit v delším časovém úseku po několika dílčích měřeních pro lepší vyhodnocení výsledků chromatické a polarizační vidové disperze. Automatizované ovládání přístroje umožní efektivnější využití stávajícího měřícího přístroje FTB - 200 a přiblížení přístrojům z vyšších řad, které v základu obsahují plánovač úloh měření. V první části je řešena problematika řešení a návrh metody řešení. Další část obsahuje návrh a realizaci programu, který ovládá stávající program od výrobce. Poslední část této kapitoly je věnována reálnému měření disperzí optických vláken pomocí automatizovaného měření a vyhodnocení dlouhodobých výsledků.

4.1 Popis metody řešení

Nejvíce časově náročné na této práci bylo vybrání metody ovládání a vypořádání se se všemi nástrahami přístroje. Je zde popsán výběr metody, programovacího jazyka a potíže s nasazením grafické aplikace na přístroj.

4.1.1 Platforma měřícího přístroje FTB-200

Na měřícím přístroji běží operační systém XP. Jako úložiště slouží 8 GB flash disk. Disk je rozdělen na systémový a datový oddíl. Na systémovém disku je nainstalovaný pouze operační systém XP, všechny další programy a data se nacházejí na datovém oddílu. Na systémový oddíl nelze doinstalovat další programy větší než cca 150 MB pro nedostatečné místo na disku. Toto omezení nám ztěžuje nasazení grafické aplikace (nemožnost nainstalovat Microsoft .NET Framework).

4.1.2 Výběr metody řešení

Nejlepší volbou pro automatizované měření v přístroji EXFO by bylo naprogramovat celý měřící program znovu. Překážkou tu je absence API od výrobce pro měřící modul FTB-5700. Výrobce EXFO nemá zveřejněný API na své webové stránce a na email nereagoval. Problém by se dal vyřešit zpětným inženýrstvím, což by bylo velice časově náročné a složité. Ovládání pomocí automatického ovládání myši je velice jednoduché, ale nejde vůbec předvídat chování programu.

Byla vybrána metoda ovládání standardního programu dalším programem vytvořeným v rámci bakalářské práce. Pomocí vytvořeného programu přistupujeme k standardnímu programu prostřednictvím knihovny Windows API. Tímto způsobem je možné ovládat program intuitivně a předcházet nastávajícímu chodu měření. Program byl psán ve vývojovém prostředí Microsoft Visual Studio 2010 s využitím programovacího jazyka Visual C++.

4.1.3 Grafické prostředí programu

V předcházející semestrální práci bylo ovládání programu řešeno pomocí konzolové aplikace. Pro snazší ovládání měřicího programu byl vytvořen grafický program. Pro nemožnost využití Microsoft .NET frameworku byl zhotoven program jen pomocí Windows API. Musíme zde řešit registraci třídy okna a následné vytvoření okna, smyčku zpráv, proceduru okna a zpracování zpráv.

4.2 Realizace programu

Je zde popsán návrh a realizace programu, dále jsou poté vyjmenovány a vysvětleny nejdůležitější části a jednotlivé funkce z programu. Celý program včetně zdrojových kódů je přiložen na CD.

4.2.1 Realizace Win32 Aplikace

Celý program je složený ze tří základních stavebních prvků, které jsou nezbytně nutné pro chod nejjednoduššího programu ve Windows. Hlavní části programu jsou:

- smyčka zpráv,
- registrace třídy okna a vytvoření okna,
- procedura okna a zpracování zpráv.

Následně jsou základní části kódu programu vloženy do práce a stručně popsány. Vzhled výsledné aplikace je uveden na obr. 4.1.

Smyčka zpráv

Smyčka zpráv je umístěna uvnitř hlavní funkce našeho programu WinMain. Celý náš program běží v téhle jediné smyčce zpráv. Funkce *GetMessage* vezme zprávu z fronty zpráv aplikace a odešle ji k dalšímu zpracování. Jedinou výjimkou je zpráva *WM_QUIT*, při které funkce *GetMessage* vrátí 0 a smyčka zpráv je ukončena. Posléze dojde k ukončení celé funkce WinMain a tím pádem celé aplikace.

Uvnitř smyčky zpráv jsou funkce *TranslateMessage* a *DispatchMessage*. První funkce překládá virtuální kódy zpráv klávesnice a generuje další znakové zprávy. Funkce *DispatchMessage* odešle zprávu do procedury příslušného okna.

```
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInst,
                  LPSTR lpCmdLine, int nShow)
{
    if ( !InitApp() )
        return FALSE;
    g_hInstance = hInstance;
    while ( GetMessage(&msg, NULL, 0, 0) )
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    return msg.wParam;
}
```

Registrace třídy okna

V našem programu máme zaregistrovanou jedinou vlastní třídu okna, kterou využívá hlavní okno aplikace. Další prvky naší aplikace jsou tlačítka, která využívají systémovou třídu. Tuto třídu nemusíme ani nemůžeme vytvářet.

Níže je uvedena registrace třídy, která definuje určité vlastnosti pro následně vytvořené hlavní okno aplikace.

```
WNDCLASSEX wc;
wc.cbSize = sizeof(WNDCLASSEX);
wc.cbClsExtra = 0;
wc.cbWndExtra = 0;
wc.hbrBackground = (HBRUSH)(COLOR_WINDOW + 1);
wc.hCursor = LoadCursor(NULL, IDC_ARROW);
wc.hIcon = LoadIcon(NULL, IDI_APPLICATION);
wc.hIconSm = LoadIcon(NULL, IDI_APPLICATION);
wc.hInstance = g_hInstance;
wc.lpfnWndProc = WindowProcMain;
wc.lpszClassName = _MainClassName;
wc.lpszMenuName = MAKEINTRESOURCE(IDR_MENU1);
wc.style = CS_HREDRAW | CS_VREDRAW;
if ( !RegisterClassEx(&wc) )
    return FALSE;
```

Pomocí funkce *CreateWindowsEx* si vytvoříme hlavní okno aplikace založené na třídě vytvořené v předešlém kroku. Při vytváření okna jsou definované další vlastnosti.

```
//hlavní okno
g_hwndMain = CreateWindowEx(0, _MainClassName,
    _AppName,
    WS_OVERLAPPEDWINDOW | WS_VISIBLE,
    100, 100, 400, 550,
    NULL, NULL, g_hInstance, NULL);
if ( g_hwndMain == NULL )
    return FALSE;
```

Dále jsou vytvářena okna založena na systémové třídě `BUTTON`. Zde je příklad vytvoření tlačítka `Start` pro spouštění měření.

```
// button Start
g_hwndStart = CreateWindowEx(0,
    TEXT("BUTTON"),
    TEXT("START"),
    WS_CHILD | WS_VISIBLE | BS_PUSHBUTTON,
    10, 310, 75, 25,
    g_hwndMain,
    (HMENU)NULL,
    g_hInstance,
    NULL);
if ( g_hwndStart == NULL )
    return FALSE;
```

Procedura okna

Zprávy, na které chceme reagovat, zachytáváme v proceduře okna pomocí příkazu `switch`. Všechny zprávy (včetně těch, na které reagujeme) posíláme pomocí funkce `DefWindowProc` k výchozímu zpracování. Ve zkrácené proceduře okna z našeho programu rozlišujeme zprávy `WM_COMMAND` a `WM_DESTROY`. Pomocí filtru zpráv `WM_COMMAND` reagujeme na stisk tlačítek. Po doručení zprávy `WM_DESTROY` vložíme do smyčky zpráv zprávu `WM_QUIT`, kterou realizujeme pomocí funkce `PostQuitMessage`. Touto zprávou ukončíme celý program.

```
LRESULT CALLBACK WindowProcMain(HWND hwnd, UINT uMsg, WPARAM wParam,
    LPARAM lParam)
{
    switch ( uMsg )
    {
    case WM_COMMAND:
        if ( lParam == (LPARAM)g_hwndStart){
            hlavniFunkce();
        }
        break;
    case WM_DESTROY:
        PostQuitMessage(0);
        break;
```

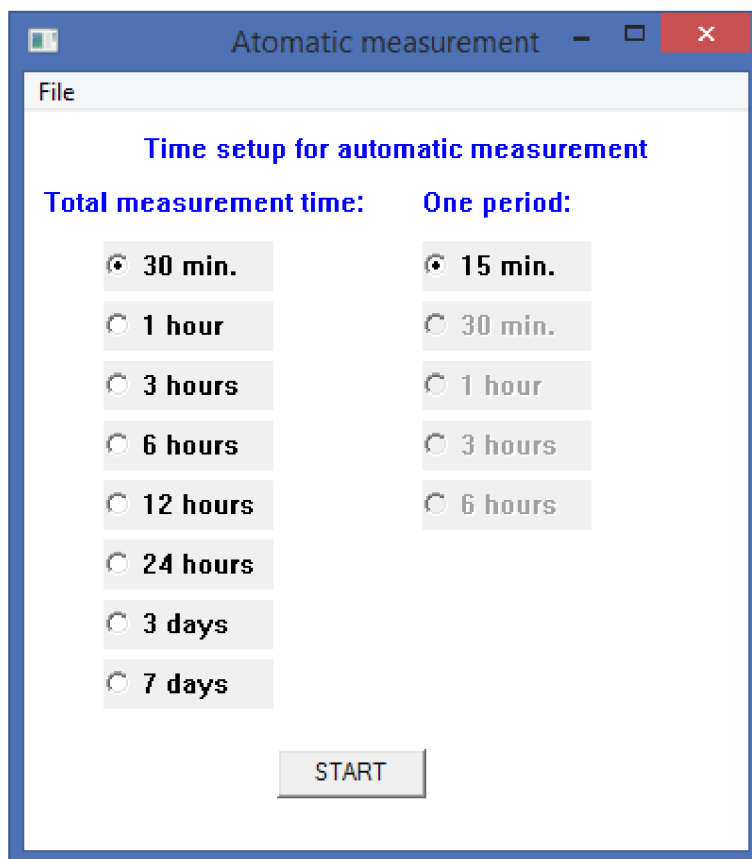
```

}
return DefWindowProc(hwnd, uMsg, wParam, lParam);
}

```

Popis ovládacích prvků programu

Na obr. 4.1 je vzhled vytvořeného programu pro automatizované měření. Nastavení automatizovaného ovládání je velice snadné a intuitivní. Před začátkem měření je třeba v levém sloupci vybrat celkovou dobu měření a v pravém sloupci čas opakování dílčích měření. Při spuštění programu jsou standardně vybrané první možnosti z časové nabídky měření. U programu nelze vybrat nelogické časy měření (celkovou dobu měření menší než čas opakování). V každém sloupci jsou vždy aktivní pouze prvky, které jsou vhodnou kombinací k vybranému prvku v druhém sloupci. Program dále obsahuje hlavní nabídku s nápovědou k měření. Nakonec měření spustíme tlačítkem START, čímž se hlavní měřící program přepne do popředí a spustí se automatizované měření.



Obr. 4.1: Vzhled programu

4.2.2 Ovládání měřicího programu

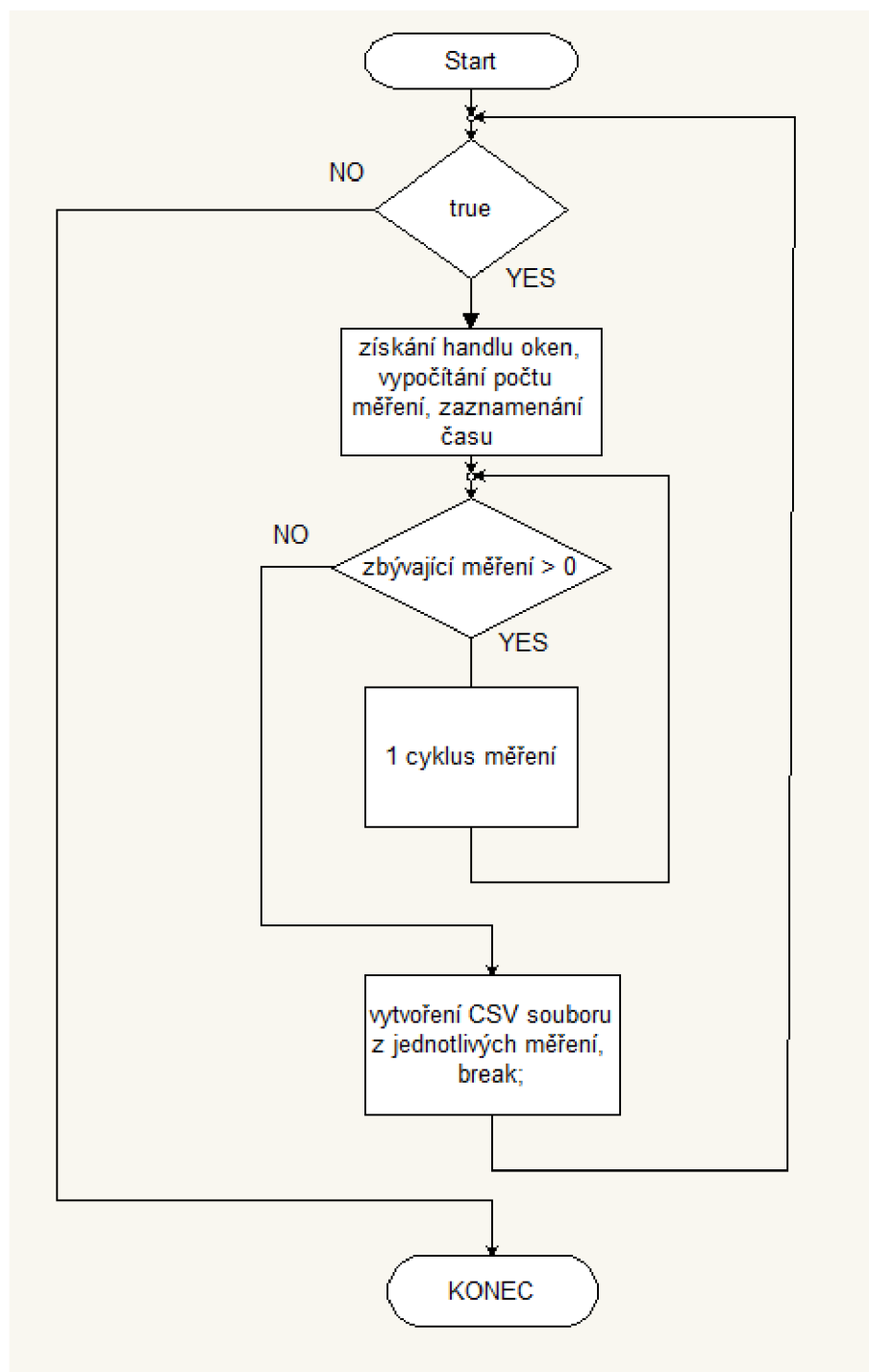
Ovládání je realizováno pomocí zasílání zpráv stávajícímu programu pro měření disperzí pomocí naší aplikace. Před začátkem automatizovaného měření je třeba spustit program „Compact Toolbox“ a spustit modul SEDA pro měření disperzí. Dále vybereme možnost měření CD, PMD nebo obě měření najednou viz obr.2.2. Následně spustíme naši aplikaci, vybereme celkovou dobu měření v levém sloupci a dobu jednoho dílčího měření ve sloupci pravém a nakonec klikneme na tlačítko START. Tím se spustí hlavní funkce programu *hlavníFunkce*, která se stará o celé ovládání měření. Nejdůležitější části funkce a její vnořené funkce si popíšeme.

Hlavní funkce ovládání měření

Hlavní funkce programu se stará o ovládání měření disperzí. Vývojový diagram funkce je na obr.4.2. Celý Kód je vložen do cyklu *while* s podmínkou *true*, tento cyklus slouží pouze pro okamžité ukončení funkce při chybě ovládání pomocí příkazu *break*. Funkce nejdříve zjistí *handles* všech prvků, pomocí kterých budeme ovládat měření. Dále se uloží systémový čas začátku měření pro synchronizaci celého měření podle definovaných časů měření uživatelem.

Dále funkce obsahuje cyklus *while*, který se stará o ovládání měření. Vnořená funkce *start* se stará o spuštění a určení konce jednotlivého měření. Po skončení měření se musí uložit binární data u udělat report do html souboru. O tuto problematiku se starají funkce *Save* a *Report*. Následuje cyklus *while* pro čekání na další spuštění měření. Synchronizace je prováděna pomocí porovnání časů v sekundách. Porovnává se aktuální čas s časem začátku měření, který je navýšen o určitý časový interval.

```
while(kolik > 0){
    start();
    save();
    if(report())break;
    //casovani
    if(kolik !=1){
        while(true){
            Sleep(1000);
            t_ted=time(NULL);
            if((t_start+merS*pocet)<t_ted) break;
        }
    }
    kolik --;
    pocet ++;
}
```

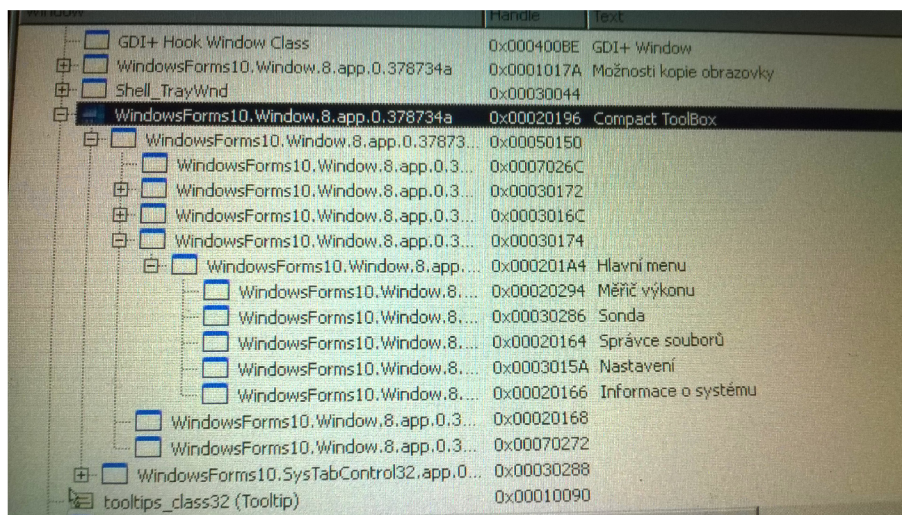


Obr. 4.2: Hlavní funkce měření

Na konci hlavní funkce je vložena funkce *zapis()*, která se stará o výpis naměřených výsledků z jednotlivých měření do jednoho CSV souboru. Dále si popíšeme jednotlivé funkce, které se starají o ovládání měření.

Získání handle oken

Nejdříve si pomocí programu Window Detective zjistíme strukturu měřícího programu. Struktura programu je na obr. 4.3, kde můžeme vidět název třídy, číslo handle a pojmenování prvku. Pro získání handle hlavního okna použijeme funkci *FindWindows*, která má dva parametry. První parametr je název třídy okna, druhým parametrem je název okna. Funkce vrátí aktuální handle okna.



Obr. 4.3: Struktura oken standardního programu měření

Další handle dceřiných prvků hlavního okna získáme pomocí funkce *FindWindowEx*, která má 4 vstupní parametry. Třetí a čtvrtý parametr je shodný s funkcí *FindWindows*. Prvním parametrem je handle rodičovského okna a druhým parametrem je handle dceřiného okna.

Z důvodu stejných tříd oken v celém programu a absence názvů dceřiných oken není přesně definované, na jaký handle nám funkce vrátí hodnotu. Potom je návratová hodnota handle vždy na první okno na dané úrovni. Pomocí funkce *GetWindow* najdeme handle na další okna ve stejné úrovni. Funkce má dva parametry. Prvním z nich je handle ze stejné úrovně, druhý parametr označuje jaké okno hledáme. Můžeme se posunout na první, poslední, následující nebo předcházející index okna v dané úrovni. V ukázce z programu viz níže vidíme použití funkcí *FindWindow*, *FindWindowEx* a *GetWindow*. Dále je použita funkce *ShowWindow* pro maximalizování měřícího programu a funkce *SetForegroundWindow* vloží podproces, který vytvořil zadané okno do popředí a aktivuje okno. Aktivní okno potřebujeme pro ovládání měření.

```
h_seda = FindWindow(NULL, L"SEDA");  
if(NULL != h_seda){  
    ShowWindow(h_seda, SW_MAXIMIZE);  
}
```

```

SetForegroundWindow(h_seda);

h_results = FindWindowEx(h_seda, NULL, L"WindowsForms10.Window.8.app.0.378734a",
                        L"Results");
if (h_results==NULL){
    MessageBox(NULL, L"Error handle Results.", _T("Chyba"), MB_OK);
    return 1;
}
h_pMenu= GetWindow(h_results,GW_HWNDNEXT);
if (h_pMenu==NULL){
    MessageBox(NULL, L"Error handle pMenu.", _T("Chyba"), MB_OK);
    return 1;
}
}
}

```

Spouštění měření

O spuštění měření se stará funkce *start*, která nejdříve pošle zprávy *PostMessage* handlu tlačítka START. Zprávy s druhým parametrem *WM_LBUTTONDOWN* a *WM_LBUTTONUP* vykonají stisk tlačítka a měření je spuštěno. Tlačítko je v průběhu měření přejmenováno na Stop, konec jednotlivého měření poznáme pomocí přejmenování tlačítka zpět na Start. Text z tlačítka zjistíme zasláním zprávy *SendMessage* s parametrem *WM_GETTEXT*. Tato funkce je umístěna v nekonečné smyčce *while*, při přejmenování tlačítka zpět dojde k ukončení cyklu.

```

void start(void){
    Sleep(1000);
    PostMessage(h_start, WM_LBUTTONDOWN, 0,0);
    PostMessage(h_start, WM_LBUTTONUP, 0, 0);
    Sleep(5000);
    TCHAR a~[10];
    char xx [10] = "Start";
    USES_CONVERSION;
    TCHAR* b = A2T(xx);
    while(true){
        SendMessage(h_start, WM_GETTEXT, sizeof(a), LPARAM(a));
        if(_tcscmp(a, b)==0) break;
        Sleep(1000);
    }
}
}

```

Uložení měření

Každé měření je nutné uložit jako binární data a posléze vytvořit HTML soubor z binárního souboru. Při ukládání prvního měření je vytvořena nová složka, do které se ukládají všechny naměřené výsledky. Celé ukládání souborů je řešeno zasíláním

zpráv tlačítkům a pomocí funkce *keybd_event*, která simuluje stisk kláves na klávesnici. Pomocí virtuálních kláves je například řešeno otevírání již vytvořené složky.

Výpis naměřených hodnot do CSV souboru

Z každého HTML souboru se vypisuje do CSV souboru datum, čas a naměřené výsledky. V HTML souboru se vyhledávají určité jedinečné řetězce a následně se vkládají do CSV souboru. Funkce pro vytvoření CSV souboru rozpoznává typ měření disperze (CD, PMD, obě najednou) a podle typu pojmenovává výsledný soubor s výsledky. Níže je zkrácený výpis pro vypisování datu měření do CSV souboru. Program hledá v souboru řetězec „dat“, který se v souboru vyskytuje právě jeden. Následně poskočí o 162 znaků vpřed (v html souboru tagy formátování stránky) a vypíše datum do souboru. Ukázka výsledného CSV souboru je na obr. 4.4.

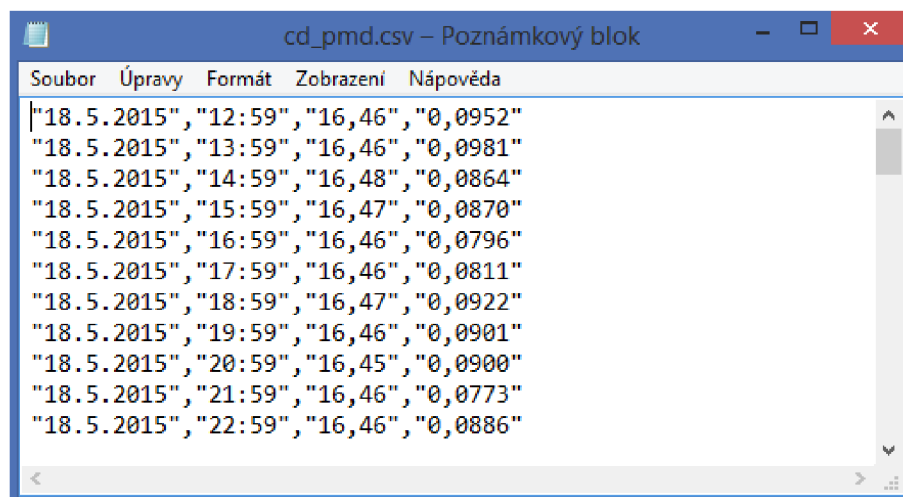
```
//zkrácený výpis kódu
FILE *fhtml,*fcsv;
fhtml = fopen(html, "r");
fcsv=fopen(csv,"w");
...
//datum
if (z=='d' && zz=='a' && zzz=='t'){
    for( m; m <162;m++){
        zzz = fgetc(fhtml);
    }
    fputc('"',fcsv);
    for(n;n<9;n++){
        zzz = fgetc(fhtml);
        fputc(zzz, fcsv);
    }
    fputc('"',fcsv);
    fputc(',')fcsv);
}
...
```

4.3 Automatizované měření disperzí

Automatizované měření bylo testováno v celém průběhu realizace programu. Program je plně optimalizovaný na dlouhodobá měření disperzí.

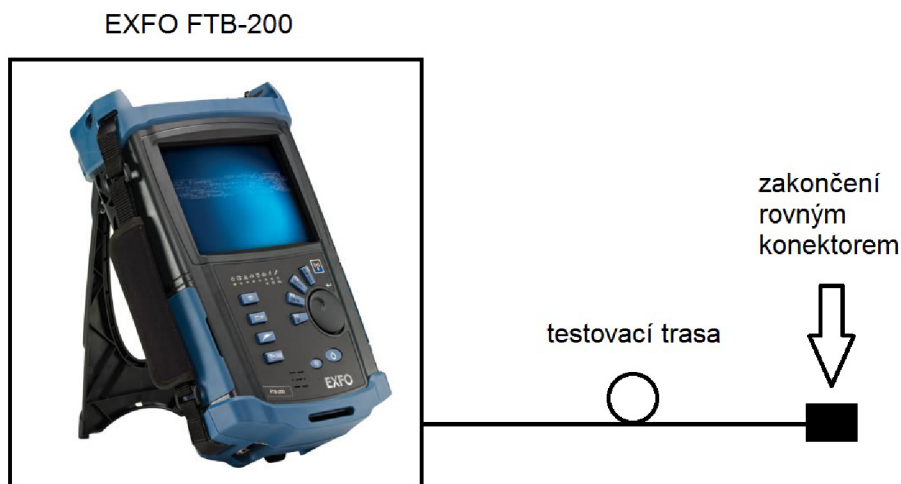
4.3.1 Měření optické trasy

V laboratoři byly provedeny dva třídní testy měření disperzí. Zapojení optické trasy s měřícím přístrojem je na obr. 4.5. První měření bylo provedeno na vlákne smotaném v laboratoři. Měření bylo nastavené s opakováním každých 15 min, tedy



Obr. 4.4: CSV soubor s výpisem dat z HTML souborů

celkem 288 dílčích měření. Druhé testovací měření bylo nastaveno s opakováním každou hodinu a bylo provedeno na optickém testovacím vlákně mezi VUT a Masarykovou univerzitou. Měření bylo prováděno každou hodinu.

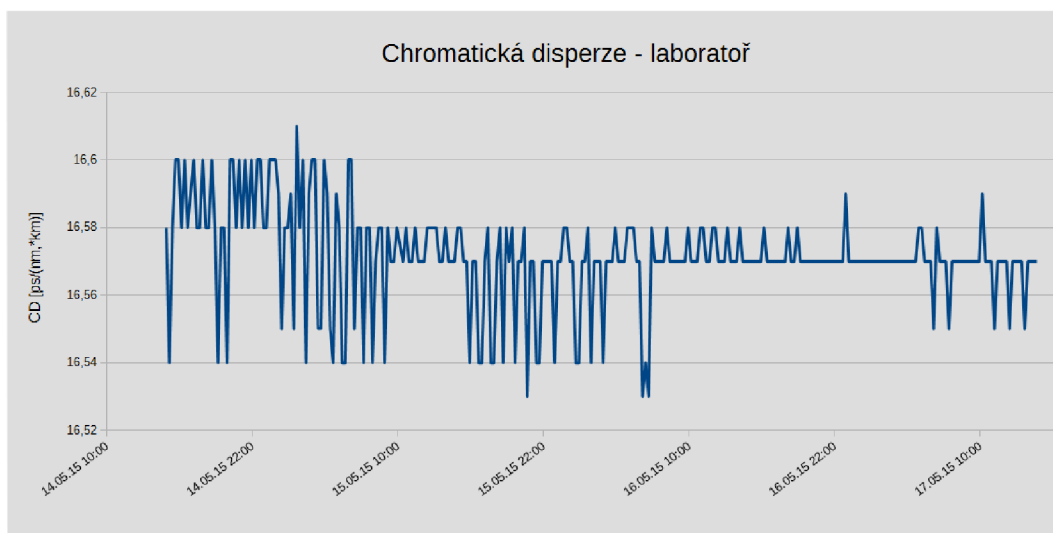


Obr. 4.5: Zapojení měření s přístrojem EXFO FTB - 200 (EXFO převzaté z www.profiber.eu)

4.3.2 Výsledky a vyhodnocení

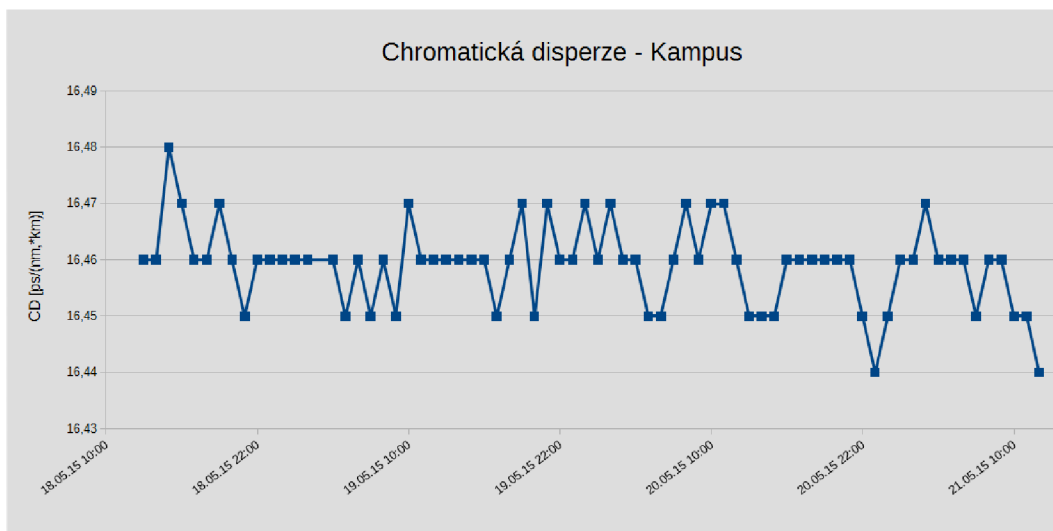
Z každého měření disperzí je vytvořen graf chromatické a polarizační vidové disperze v závislosti na čase.

Na obr. 4.6 je graf chromatické disperze optického vlákna měřené na cívce v laboratoři. Hodnoty kolísají mezi hodnotami 16,53 ps/(nm*km) a 16,61 ps/(nm*km), kolísání se vyskytuje hlavně v první polovině časového intervalu. Druhá polovina měření je stálejší, zřejmě způsobeno menšími otřesy v laboratoři o víkendu.



Obr. 4.6: Chromatická disperze vlákna - smotané vlákno

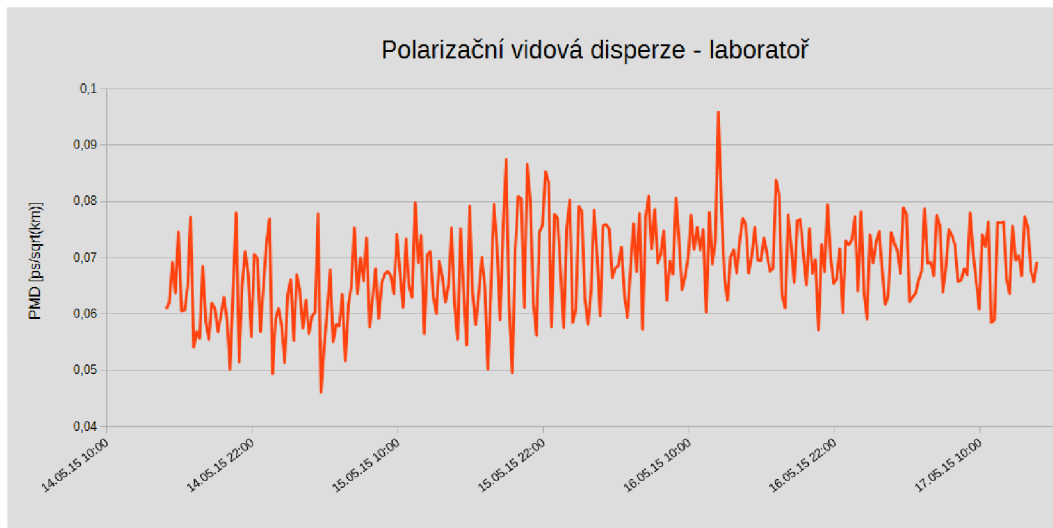
Na dalším grafu na obr. 4.7 je chromatická disperze optického vlákna, které je položeno mezi univerzitami v Brně. Naměřené koeficienty CD jsou v celém průběhu přibližně konstantní.



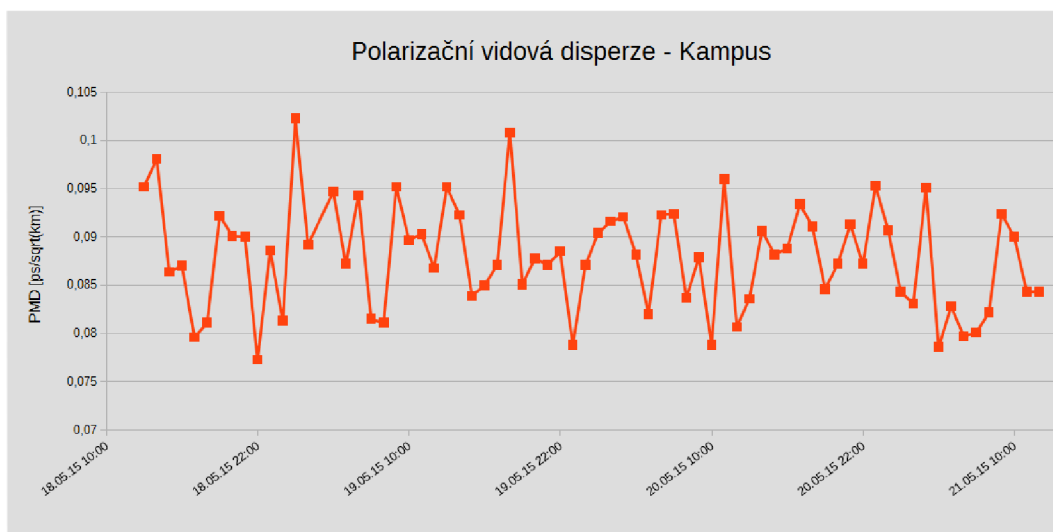
Obr. 4.7: Chromatická disperze vlákna - reálné vlákno

Koeficient polarizační vidové disperze v obou měřeních kolísá. Na rozdíl od chromatické disperze je PMD jev nestálý. Vliv na výsledek může mít porušená geometrie,

vnitřní pnutí nebo teplota vlákna. Graf polarizační vidové disperze vlákna umístěného v laboratoři je na obr. 4.8 a vlákna položeného mezi univerzitami je na obr. 4.9.



Obr. 4.8: Polarizační vidová disperze - smotané vlákno



Obr. 4.9: Polarizační vidová disperze - reálné vlákno

5 ZÁVĚR

V bakalářské práci jsem se zabýval návrhem a realizací programu pro automatické ovládání měřícího programu v přístroji EXFO. V první části jsem se zabýval optickými vlákny a popisem chromatické a polarizační vidové disperze. V této části se nachází také metody jejího měření.

Dále jsem řešil, jak automatizované ovládání řešit. V úvahu přicházelo několik variant. Nejpreciznější metodou je vytvořit celý měřící program založený na API k měřicímu modulu. Bohužel výrobce API neposkytuje ani na vyžádání. Naopak nejjednodušší metodou je automatizované klikání kurzorem myši na předem definované místo na obrazovce. Velkou nevýhodou je předem určený statický čas mezi klikáním a velká pravděpodobnost narušení automatizovaného ovládání jinou vyskakovací aplikací. Nakonec byla vybrána metoda ovládání měřícího programu pomocí dalšího programu.

Výsledkem bakalářské práce je zhotovení programu, který ovládá stávající měřící program. Program zjišťuje čísla handlů oken a zasílá jim zprávy s příkazy. Dále program vyvolává virtuální stisk kláves pro usnadnění ovládání měření. Výsledkem automatizovaného měření je složka s binárními a html soubory naměřených dat. Navíc program vytvoří z html souborů jeden CSV soubor s hodnotami a časem ze všech měření.

Poslední část bakalářské práce je věnována testovacímu automatizovanému měření disperzí. Měření není prováděno za účelem zkoumání naměřených dat disperzí optických vláken, ale za účelem demonstrace funkčnosti programu. V prvních zkušebních testech automatizovaného měření docházelo k chybám, které byly následně odstraňovány. Program v aktuální verzi běží bezchybně.

LITERATURA

- [1] EXFO Electro-Optical Engineering *Inc.*: *FTB-200 Compact Platform* [online], [cit. 8. 12. 2014]. Dostupné z URL: <http://www.exfo.com/Documents/TechDocuments/Specification_Sheets/EXFO_spec-sheet_FTB-200-v10_en.pdf>.
- [2] EXFO Electro-Optical Engineering *Inc.*: *FTB-5700 Single-Ended Dispersion Analyzer* [online], [cit. 8. 12. 2014]. Dostupné z URL: <http://www.exfo.com/Documents/TechDocuments/Specification_Sheets/EXFO_spec-sheet_FTB-5700-v12_en.pdf>.
- [3] Filka, M. *Přenosová media. Skripta. VUT FEKT, Brno 2012*
- [4] Filka, M. *Optoelektronika pro telekomunikace a informatiku. Vyd. 1. Brno: Miroslav Filka, 2009, 369 s. ISBN 978-80-86785-14-1.2*
- [5] HÁJEK, M., HOLOMEČEK, P. *Měření chromatické a polarizační vidové disperze jednovidových optických tras* [online], [cit. 5. 12. 2014]. Dostupné z URL: <<http://www.mikrokom.eu/skoleni/pdf/CD-PMDcablex2002.pdf>>.
- [6] HÁJEK, M., HOLOMEČEK, P. *POTDR - lokalizace úseků se zvýšenou hodnotou PMD v optických trasách* [online], [cit. 6. 12. 2014]. Dostupné z URL: <<http://www.mikrokom.eu/skoleni/pdf/potdr.pdf>>.
- [7] PETZOLD, Charles *Programování ve Windows: legendární publikace o programování Win32 API. Vyd. 1. Praha: Computer Press, 1999, xxxiii, 1216 s. ISBN 80-722-6206-8.*
- [8] BOHÁČ, A. *Win32 API : kompletní reference programátora pro Windows 95 a Windows NT : průvodce vývojáře. 1. vyd. Brno: Unis publishing, 1997, 669 s. ISBN 80-860-9706-4.*
- [9] MATOUŠEK, David. *Visual C 6.0: vývojové prostředí. 1. vyd. Překlad Alois Boháč. Praha: BEN, 2003, 359 s. ISBN 80-730-0130-6.*
- [10] KARASEK, Miroslav a Tomas ZAHRADKA. *Disperze signálu v optických vláknových vlnovodech. Vyd. 1. Praha: Academia, 1990, 130 p. Studie CSAV, 1990, c. 8. ISBN 80-200-0216-2.*

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

API	Application Programming Interface
EXFO	EXperts in Fibre Optics
CD	Chromatic Dispersion
CSV	Comma-separated values
DCF	Dispersion Compensating Fiber
DS	Dispersion Shifted
DOP	Degree Of Polarization
DWDM	Dense wavelength division multiplexing
HTML	Hyper Text Markup Language
IBM	International Business Machines Corporation
OTDR	optical time-domain reflectometer
PMD	Polarization Mode Dispersion
POTDR	Polarization OTDR
t_g	celkové zpoždění signálu
l	délka vlákna
D	disperze
n_1, n_2	index lomu světla
ϕ	mezní úhel
λ	vlnová délka
$\Delta\tau$	zpoždění signálu

SEZNAM PŘÍLOH

A Spuštění měření	46
B Obsah přiloženého CD	47

A SPUŠTĚNÍ MĚŘENÍ

Měření je možné vyzkoušet na přístroji FTB - 200 v laboratoři optických sítí. Pro spuštění programu je nutné zkopírovat knihovnu „msvcr100d.dll“ do system32 nebo do kořenového adresáře programu. Před spuštěním automatizovaného měření je nutné spustit program dodávaný výrobcem a vybrat typ měření (CD a PMD) disperze optického vlákna. Dále v našem vytvořeném programu vybrat celkovou dobu měření a čas opakování jednotlivých dílčích měření. Tlačítkem Start se spustí měření, tím se hlavní měřicí program maximalizuje a nastaví do popředí. Výsledkem automatizovaného měření jsou binární a html soubory z dílčích měření. Soubory se nacházejí ve vytvořené složce pojmenované „Auto. mereni dd. mm. rrrr hh mm ss“ v adresáři „D:/Documents and Settings/Operator/My Documents/CDPMD“. Výsledný CSV soubor se souhrnem naměřených hodnot se ve vytvořené složce nachází též.

B OBSAH PŘILOŽENÉHO CD

Na přiloženém CD se nachází bakalářská práce, spustitelný program, knihovna potřebná ke spuštění programu a složka se zdrojovými kódy. Aplikace byla testována ve Visual Studio 2010.