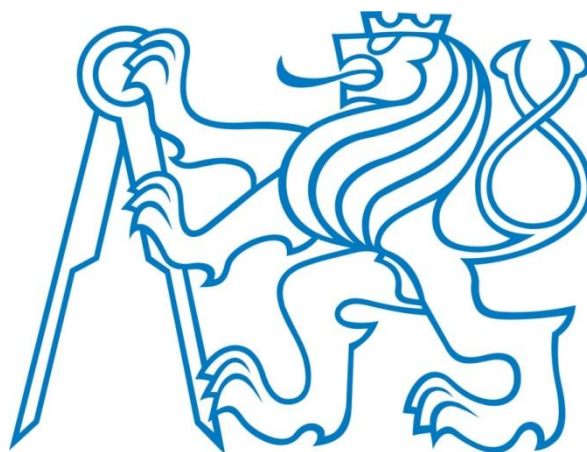


# České vysoké učení technické

## Fakulta elektrotechnická

### Víceúčelový laboratorní testovací přístroj na bázi mikrořadiče



Jiří Hladík, Jakub Halcin, Jan Staněk,  
Petr Makeš, Filip Škraňka, Adam Bařtipán

A3M99PTO – Práce v týmu a její organizace

Zadavatel: doc. Ing. Jan Fischer, CSc.

## Obsah

<b>1</b>	<b>ZADÁNÍ PROJEKTU .....</b>	<b>4</b>
1.1	Zadání od zadavatele .....	4
1.2	Cíl projektu .....	5
1.3	Mise .....	5
1.4	Vize.....	5
<b>2</b>	<b>SPECIFIKACE ŘEŠENÉHO PROBLÉMU.....</b>	<b>6</b>
2.1	Výchozí stav projektu.....	6
2.2	Budoucí stav projektu.....	6
2.3	Funkční a nefunkční požadavky .....	6
2.3.1	Funkční požadavky .....	7
2.3.2	Nefunkční požadavky .....	7
2.4	SMART .....	7
<b>3</b>	<b>POSTUP ŘEŠENÍ.....</b>	<b>9</b>
3.1	Zajištění projektu .....	9
3.1.1	Vývojový tým.....	9
3.2	Použitý hardware.....	10
3.3	Použitý software.....	10
3.4	SWOT analýza .....	10
3.5	Matice zodpovědnosti .....	12
3.6	Analýza rizik FMEA .....	13
3.7	Ganttův diagram.....	15
3.8	Kritická cesta projektu.....	16
<b>4</b>	<b>POROVNÁNÍ PLÁNU A SKUTEČNOSTI .....</b>	<b>18</b>
<b>5</b>	<b>ŘEŠENÍ PROBLÉMŮ .....</b>	<b>20</b>
5.1	Špatný odhad časové náročnosti .....	20
5.2	Rozdílné znalosti členů týmu.....	20
5.3	Kompilování na ADuC843.....	20
5.4	Nedostatky navržené architektury .....	21
5.5	Ztráta motivace.....	21
<b>6</b>	<b>ZÁVĚR .....</b>	<b>22</b>
<b>7</b>	<b>REFERENCE .....</b>	<b>24</b>

## Seznam obrázků

Obr. 1 - Předpokládaný časový plán v Genttově diagramu .....	15
Obr. 2 - Znázornění kritické cesty projektu .....	16
Obr. 3 - Zjednodušený model architektury .....	18
Obr. 4 - Okno osciloskopu v počítačové aplikaci (signál 50Hz na anténě) .....	19

## Seznam tabulek

Tab. 1 - Vyhodnocení SWOT analýzy .....	11
Tab. 2 - Definování matice zodpovědnosti za jednotlivé úkoly .....	12
Tab. 3 - Vyhodnocení rizik metodou FMEA .....	13

# 1 Zadání projektu

## 1.1 Zadání od zadavatele

Analyzujte potřebu měřicích a testovacích přístrojů při obvyklé laboratorní výuce na katedře měření a domácí přípravě studentů v předmětech A3B38MMP, A4B38NVS, A3M38VBM a A3M38AVS. Zjistěte, které funkce používaných měřicích přístrojů by bylo možno nahradit ve velmi zjednodušené formě mikrořadičem řady STM32Fxxx a mikrořadičem ADuC843.

Cílem projektu je vytvořit řešení, kdy pro některé laboratorní funkce bude možno místo standardních měřicích přístrojů (impulsní generátor, čítač, voltmetr, osciloskop, logický analyzátor) možno využít velmi jednoduchou desku s mikrořadičem napojeným na PC. To by v jednoduchých případech, především při domácí práci studentů v předmětech zaměřených na praktickou realizaci, mělo vyřešit potřebu měřicích přístrojů.

Definujte jednotlivé přístrojové funkce a jejich požadované parametry s ohledem na jejich realizovatelnost mikrořadičem. Realizujte dané přístroje ve verzi s řadiči s jádrem ARM Cortex ve formě STM32F0xx, STM32F1xxxxa STm32F3xxx a dále s řadiči jádrem 8051 ve formě ADuC843. Orientujte se na dostupné desky, např. STM32F4 Discovery kit, STM32F3Discovery kit, které budou napojeny na PC s využitím vnitřního rozhraní USB, a desek STM32F0 Discovery kit a STMVL Discovery, které budou napojeny s využitím UART a externího převodu na rozhraní RS232. Přístroje se budou v jednodušší formě ovládat a nastavovat znakově pomocí terminálového programu běžícího na PC, na něj budou připojeny prostřednictvím rozhraní RS232 nebo rozhraní USB, případně též pomocí externího převodníku USB/RS232. Ve složitější formě vyžadující zobrazení zaznamenaného průběhu (funkce osciloskop nebo logický analyzátor) vytvořte potřebnou aplikaci pro PC. Navrhněte a realizujte doplňkové obvody, které zajistí ochranu vstupů a

výstupů mikrořadiče při jejich nesprávném připojení na nehodná napětí. Výsledky řešení v jednotlivých etapách ověřte ve spolupráci se studenty v návaznosti na souběžně běžící výuku předmětu A3B38MMP. Vytvořte příslušnou WWW stránku, kde bude možno najít vytvořené programy i popisy jejich využití na daných deskách. Zvažte využití stránek wiki

[http://neuron.feld.cvut.cz/publicwiki/Hlavn%C3%AD\\_strana](http://neuron.feld.cvut.cz/publicwiki/Hlavn%C3%AD_strana)

pro ukládání mezivýsledků práce jednotlivých členů týmu i zadavatele projektu pro jejich vzájemnou informovanost o postupu prací.

## **1.2 Cíl projektu**

Cílem tohoto projektu je připravit zařízení schopná svými vlastnostmi nahradit běžně používané laboratorní přístroje na cvičeních organizovaných katedrou měření. Součástí projektu je analýza vhodných funkcí, prozkoumání možností řešení v rámci omezení daných zadáním, následná implementace.

## **1.3 Mise**

Výsledek projektu poskytne základ platformy, na které bude možné vytvářet cenově dobře dostupné přístroje, které se svými vlastnostmi budou podobat drahým laboratorním přístrojům. Studenti budou s jistými omezeními využívat při řešení semestrálních prací v domácích podmínkách osciloskop, logický analyzátor, generátor nebo jednoduchý voltmetr. Koncept poskytne dostatečné možnosti pro budoucí rozvoj a umožní navázat dalším projektům podobného charakteru.

## **1.4 Vize**

Navázání jiného budoucího týmu na stávající projektovou dokumentaci. Po vyjasnění zájmů katedry možné využití třetími stranami. Ideálně prosazení jako konkurenceschopný opensource projekt. Propagace katedry měření.

## **2 Specifikace řešeného problému**

### **2.1 Výchozí stav projektu**

Projekt našeho týmu vzniká tzv. „na zelené louce“. Nenavazuje na předchozí projekt, ani se nemůže opírat o předem zpracovanou analýzu. K dispozici máme sadu vývojových kitů zapůjčených zadavatelem projektu a možnost využít vývojový software licencovaný v rámci MSDNAA.

Při řešení můžeme využít know-how, které přináší znalosti jednotlivých členů týmu. Jejich potenciál a rozsah však není zprvu zcela jasný. Stejně tak nelze přesně odhadnout časové možnosti jednotlivých členů a sílu jejich motivace se na projektu aktivně podílet.

### **2.2 Budoucí stav projektu**

Poznáme znalosti a schopnosti jednotlivých členů týmu. Následně jim budou přiřazeny vhodné prostředky pro práci a provedeme rozložení projektu na dílčí procesy. Řízením a vzájemnou komunikací pak dosáhneme stanoveného cíle. Výsledný produkt projektu - multiplatformní firmware - bude dokončen na základě paralelně probíhajících procesů, za které zodpovídají jednotliví členové týmu.

### **2.3 Funkční a nefunkční požadavky**

Zadavatel našeho projektu je odborníkem a dobře se orientuje v dané problematice. Z tohoto důvodu má vcelku ucelenou představu o tom, co by výsledné zařízení mělo dělat a jakými prostředky by se mělo jeho požadavků dosáhnout.

V rámci projektu může být komunikace se zadavatelem na odborné úrovni přínosem, zmenšuje se tím oblast případných nedorozumění. Na projektového manažera však klade vyšší nároky, protože kromě řízení projektů musí být také odborníkem (což není v praxi pravidlem) nebo musí mít kvalitní odborníky k dispozici přímo při vyjednávání se zadavatelem. Zde je shrnut seznam požadavků vyjednaných se zadavatelem, jež jsou kladeny na funkčnost a provoz výsledného zařízení.

### **2.3.1 Funkční požadavky**

- Osciloskop se vzorkovací frekvencí v rozsahu desítek až tisíců ksps
- 8 kanálový logický analyzátor, je-li to možné
- Voltmetr, je-li to možné
- Generátor signálu, je-li to možné
- PC GUI aplikace ve stylu Digilent [1]

### **2.3.2 Nefunkční požadavky**

- Implementace firmware na procesorech STM32Fxxx
- Implementace firmware na procesorech 8051 a ADuC843
- Komunikace s PC prostřednictvím RS232 a USB VCP
- Studentuvzdornost
- Kompatibilní s ovládáním z terminálu

## **2.4 SMART**

Na samém počátku projektu jsme si položili základní otázky, které se dotýkají samé podstaty naší činnosti. Pro kvalitativní zhodnocení požadavků zadavatele jsme použili standardizovanou metodu SMART.

Díky SMART jsme si ujasnili, co chceme v rámci projektu dělat, do kdy musí být činnost hotova, jakým způsobem budeme sledovat úspěšnost naší činnosti a především zda je vůbec možné požadovaného výsledku dosáhnout.

Specifický: Vytvoření univerzálního měřicího zařízení využitelného při výuce

Měřitelný: Splnění specifikací plynoucích ze zadání

Adekvátní: Nahrazení laboratorních přístrojů v domácích podmínkách

Realizovatelný: Úroveň znalostí týmu umožňuje dokončení zařízení

Terminovaný: 4.6.2013 funkční projekt dle zadání



## 3 Postup řešení

### 3.1 Zajištění projektu

#### 3.1.1 Vývojový tým

Pracovní tým je formován ze stávajícího kolektivu studentů oboru Kybernetika a robotika - Senzory a přístrojová technika. Tím je dáno poměrně jednostranné zaměření a teoreticky velmi podobná úroveň znalostí a schopností.

Minimální rozmanitost členů týmu má přímý dopad na jeho strukturu. Projektový manažer je zde současně HW i SW specialistou. Má největší spektrum zkušeností, ale (teoreticky) se nezajímá o detaily jednotlivých architektur a pouze deleguje úkoly mezi jednotlivé vývojáře a programátory.

Vedoucí vývojář udává pod vedením projektového manažera směr, kterým se budou orientovat ostatní aktivně programující členové týmu. Měl by respektovat jejich problémy a poskytnout jim adekvátní prostředky k jejich řešení.

Jiří Hladík – Projektový manažer

- Organizace činností, kontrola plnění plánu, komunikace se zadavatelem projektu, návrh SW architektury projektu, programátor C#

Jakub Halcin – Programátor

- Vývojář STM32F407, programátor MCU a C#, návrh architektury pro PC aplikaci, analýza projektu

Adam Bařtipán – Specialista MCU

- Vývojář STM32F303, programátor MCU

Jan Staněk – Vedoucí vývojář firmware

- Vývojář STM32F100, programátor MCU, návrh vyšších vrstev v MCU

Filip Škraňka – Programátor

- Vývojář STM32F051, programátor MCU, návrh komunikačního protokolu

Petr Makeš – Programátor

- Vývojář ADuC843, programátor MCU

### **3.2 Použitý hardware**

V rámci projektu jsou k dispozici pouze vývojové discovery kity společnosti STMicroelectronics s procesory STM32F051 [2], STM32F100 [3], STM32F303 [4] a STM32F407 [5], dále kit s procesorem Analog Devices ADuC843. Žádné další hardwarové prostředky nemáme k dispozici a nemáme ani prostředky na jejich pořízení.

### **3.3 Použitý software**

Firmware pro procesory určené pro výše zmíněné mikrokontrolery je možné vytvořit ve zdarma dostupné verzi vývojového prostředí Keil, která zvládá kompilovat kód s omezením do velikosti 32 kB. K vytvoření počítačové aplikace bude použito prostředí Microsoft Visual Studio 2010, jež je studentům přístupné v rámci licence MSDNAA.

### **3.4 SWOT analýza**

Při zahájení řízení projektu bylo třeba získat komplexní nadhled nad vlivy, jež budou působit uvnitř týmu a rozhodovat o rychlosti a úspěšnosti našeho postupu. Pokusili jsme se tedy o analýzu typu SWOT, která dává do souvislosti ovlivnění možných rizik a příležitostí skrze silné a slabé stránky našeho týmu a projektu.

		Silné stránky					Slabé stránky				
		Atraktivita projektu	Materiálové zajištění projektu	Silné Know-how	Dlouhodobá spolupráce se zadavatelem	Budoucí využitelnost projektu	Rozsah projektu	Část týmu v pracovním poměru	Jednostranné zaměření členů týmu	Nedostatek času	
Využití stávajících standardů	Příležitosti	1	1	6	2	5	3	1	-2	-7	10
Možnost odborné konzultace		4	6	2	9	4	-4	-2	-5	-4	13
Překrývání projektů		2	0	3	2	-3	5	0	-3	3	10
Časové vyčerpání	Rizika	4	0	4	-3	0	-7	-6	0	-9	-17
Demotivace členů týmu		9	3	6	2	7	-6	-9	-7	-7	-2
Nesprávné pochopení požadavků zadavatele		-1	4	-5	8	3	-6	-2	-6	-5	-10
		19	14	16	20	16	-15	-18	-23	-29	

Tab. 1 - Vyhodnocení SWOT analýzy

Výchozí položky SWOT analýzy byly vytvořeny na základě brainstormingu. Zaměřujeme se zde na vlastnosti čistě vývojářského charakteru, ale dotýkáme se i personálních nebo manažerských. Příležitosti a hrozby pak odrážejí vlivy, které se reálně mohou podepsat na fungování našeho týmu.

Z vytvořené SWOT analýzy vyplývá zásadní nepříznivá skutečnost. Kombinace nedostatku času na projekt, který je omezen počtem týdnů semestru a částečným pracovním úvazkem členů týmu, tvoří dvojici slabých stránek s vysokou váhou, kterou nemůžeme opomíjet.

Je nutné provést opatření proti souvisejícím rizikům, která však nelze eliminovat zcela. Tím se snadno můžeme dostat do bodu, kdy budou někteří členové týmu časově přetíženi a vyčerpáni.

Rizikem s vysokou vahou, které lze eliminovat protiopatřeními, je nesprávné pochopení požadavků (respektive představ) zadavatele. Toto lze vyřešit agilní spoluprací na projektu, kdy bude mít zadavatel přístup k výsledkům jednotlivých procesů v rámci projektu během celého jejich běhu a bude mít možnost se k nim vyjádřit. Současně se bude aktivně účastnit kontrolních dní.

Náš projekt má také celou řadu silných stránek. Mezi hlavní bychom zařadili schopnost dohodnout se se zadavatelem, jelikož s ním mají členové týmu dřívější zkušenosti. Klíčovou vlastností je silné know-how, jehož nositeli jsou jednotliví členové týmu, kteří aplikují své znalosti a zkušenosti z dřívější praxe. Příležitostí je také možnost konzultovat případné problémy zdarma s pracovníky fakulty.

### 3.5 Matice zodpovědnosti

		Jiří Hladík		Jakub Halcin		Adam bařtípán		Jan Staněk		Petr Makeš		Filip Škraňka	
1 - vykonává	2 - zodpovídá	1	2	1	2	1	2	1	2	1	2	1	2
Koordinace projektu		x			x				x				
Komunikace se zadavatelem		x			x				x				
Dokončení projektu		x			x								
STM32F051			x						x			x	
STM32F100			x					x					
STM32F303			x			x			x				
STM32F407			x	x					x				
ADuC843			x							x			
Návrh SW architektury		x							x				
Návrh komunikace			x									x	
Implementace osciloskopu			x					x					
Implementace generátoru			x					x					
Implementace log analyzátoru			x			x							
Implementace čítače									x	x			
Implementace GPIO									x			x	
PC aplikace		x		x					x		x		x

Tab. 2 - Definování matice zodpovědnosti za jednotlivé úkoly

### 3.6 Analýza rizik FMEA

Nejen na začátku, ale také v průběhu našeho projektu jsme se snažili předvídat možné rizikové situace a přijímat proti nim adekvátní opatření. Ke shrnutí této činnosti jsme se rozhodli použít nástroj FMEA a to ve formě tabulky. Ta zahrnuje rizika různorodého charakteru, která pocházejí jednak ze situace uvnitř pracovního týmu, ale také z vnějšího prostředí.

Základním problémem je skutečnost, že i při nejlepší vůli nelze předvídat všechna rizika, která mohou nastat. Seznam je výsledkem brainstormingu a pozorování vývoje rizik v průběhu práce na projektu. Hodnocení jsme se snažili stanovit objektivně.

<b>P - pravděpodobnost Z - závažnost D - detekovatelnost</b>	<b>P</b>	<b>Z</b>	<b>D</b>	<b>RPN</b>	<b>Způsob eliminace rizika</b>	<b>Původ</b>
Neudělení zápočtu	2	10	1	20	Zaměření se nejdříve na minimální požadavky pro úspěšné absolvování	Tým, požadavky předmětu
Nedodržení specifikací projektu	5	7	5	175	Ponechat rezervy ve schválených specifikacích vůči výsledkům, jež jsme schopni dosáhnout	Tým, vyšší moc
Nepochopení požadavků zadavatele	2	6	10	120	Konzultace sporných bodů zadání se zadavatelem	Tým, zadavatel
Stupňování požadavků ze strany dodavatele v průběhu	7	4	1	28	Po volbě zadavatele a projektu bez možnosti snížení rizika	Zadavatel
Vytvoření skryté návrhové chyby	10	7	10	700	Vývoj zásadních částí posuzují min. dvě nezávislé osoby	Programátoři
Špatný návrh komunikačního protokolu	5	4	7	140	Vymezit dostatek času a uvolnit několik vývojářů, brainstorming nad způsoby selhání	Programátoři
Vyčerpání možností free licence vývojového programu	7	9	2	126	Průběžná optimalizace kódu, zvážit možnost nákupu licence	Tým, zadavatel
Zranění klíčového člena týmu	1	10	10	100	Není možné snížit riziko	Člen týmu
Časové vyčerpání členů týmu	10	5	2	100	Pokusit se práci co nejlépe rozložit podle potřeb jednotlivých členů	Vyšší moc
Možná demotivace členů týmu	8	10	2	160	Pokusit se zajistit základní potřeby členů týmu, v rámci PTO však nelze	Koncepce předmětu

Tab. 3 - Vyhodnocení rizik metodou FMEA

Při pohledu na FMEA analýzu dojdeme rychle k závěru, že se zde vyskytuje mnoho rizik, která jsou akceptovatelná. Jen minimum je potřeba bezprostředně řešit. Rizika mají rozdílný stupeň detekovatelnosti, což se v praxi projevuje tím, že můžeme projev rizika mnohdy zjistit až zpětně. Jedním z takových rizik je možnost špatného pochopení požadavků zadavatele projektu.

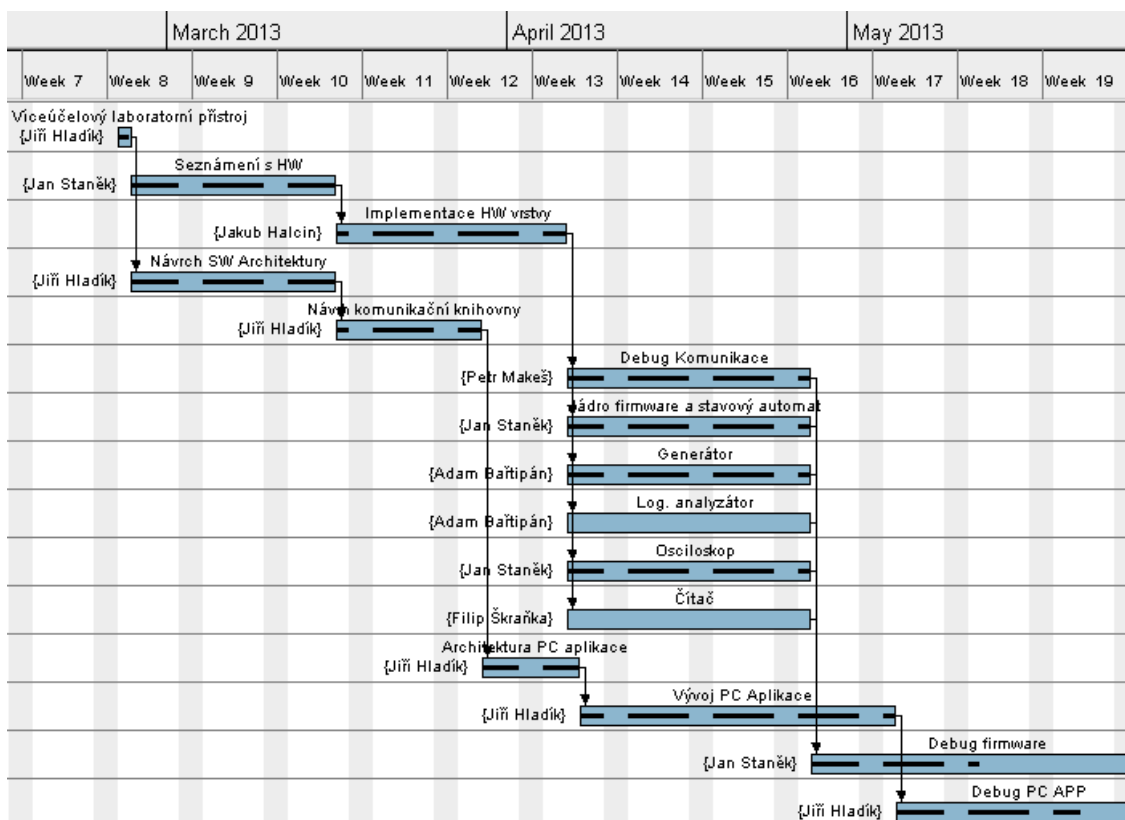
Na základě této analýzy jsme zintenzivnili komunikaci se zadavatelem v oblasti požadavků. Došlo ke zdůkladnění kontrol vyvíjeného kódu a také k úpravám časového plánu. Práci v rámci projektu jsme se rozhodli rozdělit takovým způsobem, aby výpadek jednoho člena neměl fatální dopad na celý výsledek projektu.

Jistým ústupkem rizikům, která vyplývají z nedostatku času, je fakt, že náš projekt nemůže obsahovat nic na víc. Zaměřili jsme se proto na skutečně důležité vlastnosti vyplývající ze zadání a na spolehlivou funkčnost. Některé optimalizační techniky a uživatelsky přívětivé prvky musely zůstat stranou.

### 3.7 Ganttův diagram

K plánování prací jsme použili rozšířený Ganttův diagram, který je doplněn o návaznosti mezi dílčími úkony.

Plánování časového rozvrhu prací bylo vcelku složité, jelikož jsme se museli vyrovnat s omezenou délkou semestru a náročností úkolu. Z diagramu je patrné, že jsme semestr rozdělili na čtyři větší etapy - analýzu potřebnou pro zahájení projektu, práci na nejnižších HW vrstvách, práci na vyšších vrstvách a PC aplikaci. V závěru semestru jsme se pak měli věnovat odladění a údržbě vytvořeného SW.



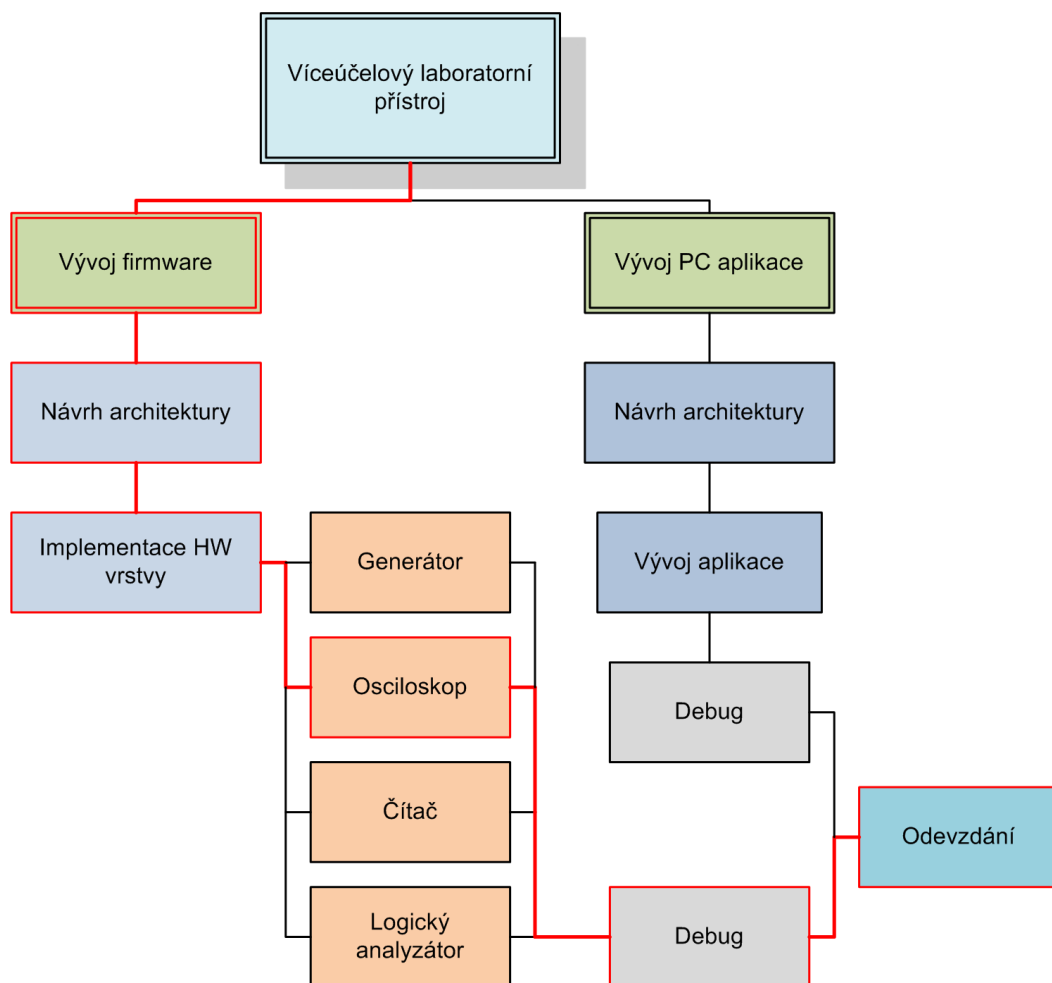
Obr. 1 - Předpokládaný časový plán v Ganttově diagramu

Řešení přidělení časových kvant ke konkrétním úkolům jsme prováděli na základě předchozích zkušeností s vývojem. Odhad náročnosti jsme nebyli schopni zpřesnit na dny, vzhledem k náročnosti jednotlivých úkonů jsme se proto rozhodli pro plánování v týdnech. V časovém plánu jsme si přesto nemohli dovolit rezervy, jaké bychom považovali za optimální.

Na začátku každého týdne pak měla proběhnout pracovní porada za přítomnosti zadavatele. Schůzky dílčích částí týmu se měly řešit individuálně.

### 3.8 Kritická cesta projektu

Dekompozici na jednotlivé dílčí úkony v rámci projektu jsme vyřešili pomocí WBS. Pro účely této zprávy byla však upravena, jelikož se svým rozsahem nehodí. Zvolili jsme poněkud nestandardní tvar tak, aby v něm bylo možné znázornit kritickou cestu projektu.



Obr. 2 - Znáznornění kritické cesty projektu

I z takto upravené WBS je patrné, že projektového manažera zajímá dosažení cíle – tedy dokončení firmware se schopnostmi laboratorního přístroje. Řídí přitom především členy týmu, kteří určují směr vývoje firmware a aplikace pro PC.



Vedoucí vývojář má poté prostor pro vlastní kreativitu, přičemž plní úkoly zadané manažerem projektu. Vývojáři navrhují vícevrstvou architekturu a tento koncept dodávají programátorům. Programátoři jsou dobře obeznámeni s jednotlivými HW vrstvami, na kterých bude koncept implementován. Mají tuto implementaci na starosti.

Pro splnění cíle projektu tak, aby zadavatel neuplatňoval sankce, je nutné vytvořit alespoň osciloskop na několika z vybraných vývojových kitů a projekt v řádném termínu odevzdat. PC aplikace není kritickým místem, jelikož už samotný firmware musí být navržen tak, aby byl schopen zpracovat příkazy terminálové komunikace. Vyvíjená PC aplikace pak může být v nouzi nahrazena aplikací třetí strany.

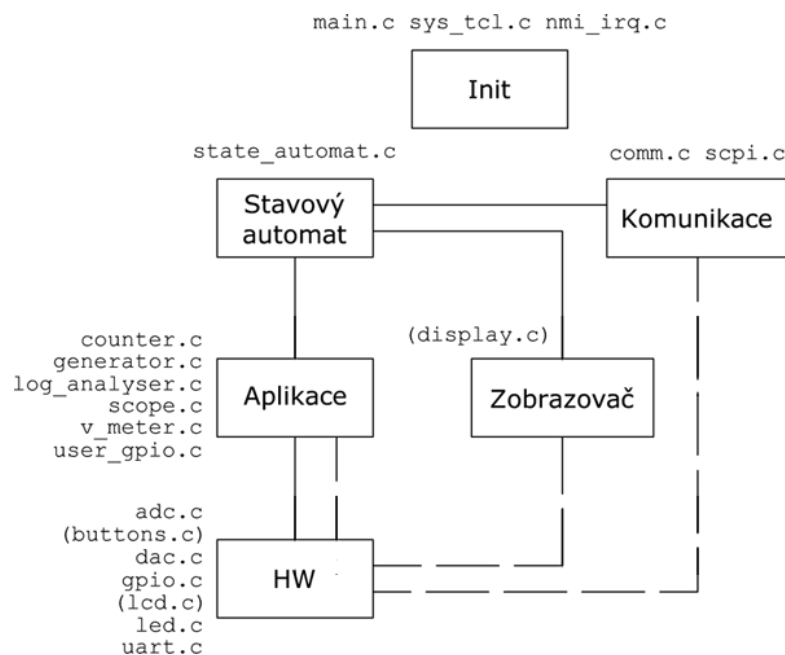
## 4 porovnání plánu a skutečnosti

Z počátku semestru jsme se jako tým dohodli na pravidelných schůzkách jednou týdně v pozdních odpoledních hodinách. Tím se zabezpečilo, že čas použitelný pro poradu nebude výrazně omezen.

Jako centrální uložení vhodné pro výměnu dat nejrůznějšího charakteru bylo zřízeno SVN na code.google [6] a pro sdílení informací o projektu, zápisů z rady a další dokumentace byla zřízena stránka na Neuron wiki na katedře měření [7]. Tím bylo vytvořeno zázemí pro předávání informací a kooperaci členů týmu. Projekt mohl být zahájen.

Prvotní snahou bylo vytvoření vlastního návrhu architektury firmware pro požadované procesory. Rychle se ukázalo, že nikdo z týmu nemá pro tento návrh dostatečné zkušenosti. Rozsah projektu se ukázal jako téměř neřešitelná překážka.

Tuto skutečnost jsme nepodcenili a raději absolvovali několik konzultací s programátorem z externí firmy. Ten nám pomohl ujasnit si strukturu potřebných funkcí a způsob vhodný způsob implementace. I přes včasný zásah a důkladnou přípravu trvalo vytvoření architektury téměř měsíc. Projekt se pak dostal do skluzu.



Obr. 3 - Zjednodušený model architektury

Na obrázku lze vidět zjednodušený model architektury. Nejnižší vrstva ovládá přímo hardware a odlišuje se vždy podle konkrétní varianty procesoru, pro který je určena. Pro jednoduchost obsahuje její popis přímo názvy ovládaných periférií.

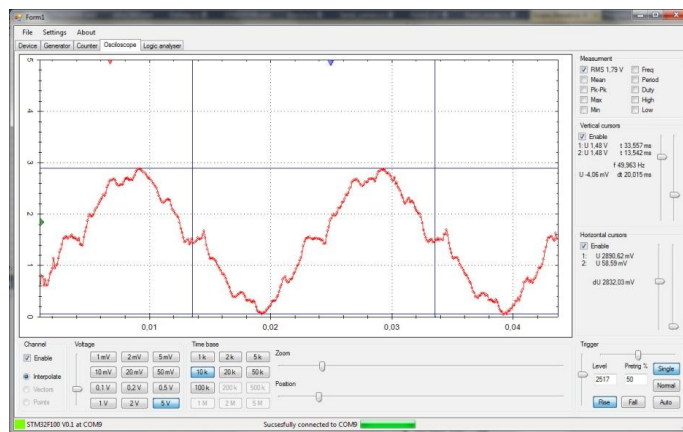
Nadřazenou vrstvu tvoří implementované funkce. Tato část je pro všechny použité procesory společná a každá dílčí část povětšinou ovládá pouze jedinou nižší vrstvu. Například osciloskop ovládá pouze AD převodníky.

Nejvyšší vrstvou je stavový automat, který řídí všechny implementované funkce a vykonává příchozí příkazy. Takto navržená architektura má výhodu snadné rozšiřitelnosti a velké přehlednosti.

Po vytvoření architektury měly začít programovací práce naplno. Vlivem pracovního vytížení všech členů týmu nepokračovaly práce předpokládanou rychlostí. Objevily se komunikační problémy při práci více členů na společném kódu. Jeden člen naprogramoval část kódu a druhý jej nemohl přeložit, protože byl napsán pro jiný procesor, nebo byl v nevhodné vrstvě programu.

Jako protiopatření byla zavedena kontrola napsaného kódu a část týmu se začala scházet na programovacích schůzkách, kde lze komunikovat přímo a tedy řešit problémy s kompilací ihned. Tato opatření zajistila zrychlení postupu v projektu a brzy se podařilo oživit osciloskop na procesoru SM32F100. Další procesory následovaly, ale na implementaci všech funkcí už nezbyl čas.

Během této doby současně probíhal návrh a vývoj počítačové aplikace. Ten proběhl spíše podle očekávání. Výsledkem je aplikace s uživatelsky přívětivým GUI.



Obr. 4 - Okno osciloskopu v počítačové aplikaci (signál 50Hz na anténě)

## 5 Řešení problémů

Vzhledem k velmi malým zkušenostem s řízením a realizací takto rozsáhlého projektu došlo v průběhu jeho vypracování k několika problémům, jež jsme museli řešit.

### 5.1 Špatný odhad časové náročnosti

Na počátku realizace projektu stálo tvrzení zadavatele, že projekt je pro tým o velikosti šesti lidí snadný. Nicméně se ukázalo, že komplexnost projektu je velmi rozsáhlá a vyžaduje přesný návrh. Například bylo potřeba přesně specifikovat možnost souběhu jednotlivých podporovaných funkcí, způsoby implementace funkcí tak, aby na všech procesorech mohlo být vše stejné atd.

Návrh softwarové architektury tedy zabral mnoho času. Poté již nezbyval čas na přesné splnění zadání, proto bylo zvoleno pokračování projektu po kritické cestě s tím, že až bude funkční osciloskop, bude možné začít vyvíjet další části, což by díky kvalitnímu návrhu architektury neměl být problém.

### 5.2 Rozdílné znalosti členů týmu

Z počátku realizace bylo rozhodnuto, že každý bude zodpovědný za svou procesorovou větev projektu, aby implementace hardwarové vrstvy proběhla v co nejkratším čase. Tento plán rozdělit jednotlivé procesory mezi členy týmu se ukázal jako nevhodný, protože kvůli rozdílným znalostem nebylo možné docílit dokončení implementace hardwarové vrstvy ve stejný okamžik. Zkušenější vývojáři tedy dokončili svou práci dříve a přesunuli se na další část projektu. Příležitostně se vraceli pomoci ostatním kolegům s jejich procesorem.

### 5.3 Kompilování na ADuC843

Po naprogramování jádra firmware se ukázalo, že procesor ADuC843 s jádrem 8051 není schopen kompilovat složitější struktury s několika ukazateli za

sebou. Kompilátor navíc nezahlásil žádné varování a hledání těchto chyb bylo velmi náročné. Chvillemi bylo nutné na vyřešení situace uvolnit až čtyři členy našeho týmu.

Celý problém se nakonec podařilo odstranit několika workarouny. Pro budoucí pokračování v projektu byla tato opatření v kódu řádně okomentována a vysvětlena.

## **5.4 Nedostatky navržené architektury**

Jelikož návrh architektury pro firmware procesoru probíhal v časovém stresu, došlo k zanesení několika chyb. Tyto chyby byly většinou typu chybějící prototyp funkce nebo duplicitní názvy struktur. Nejednalo se tedy o chyby, kvůli kterým by musel být udělán rozsáhlý zásah do celé architektury, a stačilo provést pouze drobné úpravy.

## **5.5 Ztráta motivace**

Kvůli časové náročnosti projektu a velmi malým pokrokům v rané části realizace došlo u některých členů týmu ke snížení chuti k práci na projektu. S tímto problémem se dá jen velmi těžko bojovat. Nikdo z týmu nemá manažerské schopnosti k udržení motivace a ani možnost nějak na ostatní zatlačit a donutit je k práci.

Řešením bylo neustávající nasazení ostatních členů týmu. Když členové bez motivace viděli pokrok, a že projekt někam vede, motivaci opět alespoň částečně získali.

## 6 Závěr

Zpětně lze konstatovat, že projekt, do kterého jsme se pustili, byl velmi komplikovaný a náročný. Ačkoliv jsme měli hned od počátku představu, jak budeme dílčí problémy řešit, skutečné komplikace přinášejí detaily. Vývojářskou práci nelze nazvat zcela exaktní a i zdánlivě malý problém může vývojový projekt na dlouhou dobu pozdržet.

Získané zkušenosti poukazují na skutečnost, že pro úspěch projektu je klíčová fakticky pouze jediná oblast - lidé. Úsilí nutné k zajištění materiálových zdrojů je naprosto zanedbatelné vůči úsilí, které je nutno vkládat do správné motivace a řízení členů týmu. Do této oblasti spadají také zadavatelé projektu, kteří mohou na manažera působit tlakem na změnu specifikací v průběhu projektu a nepřímo se snažit ovlivňovat chod projektu.

Při práci jsme měli možnost pozorovat, jak na náš tým působí známá Maslovova pyramida potřeb. Schopnost efektivní práce týmu do značné míry narušovalo zázemí, které mají jednotliví členové. V poměru samostatně výdělečných členů jsme nad průměrem ostatních týmů PTO, což se nutně projevilo v motivaci některých pracovat bez ohodnocení na časově náročných úkonech.

Důvody jsou ale zřejmé - těžko lze někoho motivovat seberealizací a uznáním, pokud je toto činěno na úkor zajištění jeho základních potřeb nutných k životu. Ukázalo se, že negativní způsoby motivace fungují ve vývojářském kolektivu jen krátce. Jsou buďto ignorovány nebo situaci ještě zhoršují.

Je potřeba dodat, že skutečná náročnost projektu, se od původně předpokládané lišila také nemožností zabránit zadavateli kladení upřesňujících požadavků po uzavření zadání. V reálné situaci bychom zadavatele odradili od podobných aktivit

zavedením vícenákladů nebo posunem předpokládaného časového horizontu. Nic takového však v rámci projektu v PTO učinit nelze.

Přesto, že projekt provázelo mnoho problémů a nebyly implementovány všechny funkce, minimální požadavky zadavatele byly v dostatečné kvalitě naplněny. Výsledkem projektu je tedy funkční firmware procesorů s implementovaným osciloskopem a generátorem. Dále pak počítačová aplikace umožňující jednoduché připojení a ovládání kitů. Takto uzavřený projekt lze již použít při výuce a práci studentů na semestrálních pracích v rámci cvičení příštího semestru.

## 7 Reference

[1] **Digilent.** Analog Discovery. [Online] [Citace: 24. 5 2013.]  
<http://www.digilentinc.com/Products/Detail.cfm?NavPath=2,842,1018&Prod=ANALOG-DISCOVERY>.

[2] **STMicroelectronics.** Discovery kit for STM32 F05x. [Online]  
<http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/PF253215>.

[3] **STMicroelectronics** . Discovery kit for STM32F100. [Online]  
<http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/PF250863>.

[4] **STMicroelectronics.** Discovery kit for STM32F303. [Online]  
<http://www.st.com/web/en/catalog/tools/PF254044>.

[5] **STMicroelectronics** . Discovery kit for STM32 F4 series. [Online]  
<http://www.st.com/web/en/catalog/tools/FM116/SC959/SS1532/PF252419>.

[6] **Code Google.** PTO38TEAM SVN code sharing for laboratory multipurpose device . [Online] <http://code.google.com/p/pto38team/>.

[7] **Neuron wikipedia.** Wiki pro podporu projektů laboratoře videometrie . [Online] <http://neuron.feld.cvut.cz/publicwiki/PTO-2013>.