

Použitie knižnice Atmel Software Framework pre tvorbu embedded aplikácií

Peter Stanko

Bakalárska práca
2015



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

nascannované zadání s. 1

nascannované zadání s. 2

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

ABSTRAKT

Práca sa zaoberá medzi užívateľmi rozšírenou knižnicou Atmel Software Framework. Cieľom bol popis knižnice pre mikropočítače firmy Atmel, voľba vhodného mikropočítača a vývojového prostredia na jej demonštráciu. Následne návod na použitie, po predstavení a oboznámení sa s používaním knižnice vytvorenie ukázkových aplikácií. Posledným cieľom bolo záverečné zhodnotenie výhod a nevýhod použitia tejto knižnice pre programovanie mikropočítačov.

Na začiatok je v práci rozoberaná chronológia vývoju mikropočítačov. Od ich počiatkov až po aktuálne používané mikropočítače v praxi. Práca sa venuje moderným mikropočítačom od firmy Atmel, ich základným rozdelením a stručným popisom.

Nasleduje výber mikrokontroléra a vývojového prostredia, v ktorom bude práca a samotne ASF rozoberané. Popis knižnice obsahuje rozdelenie na funkcie a možnosti jej využitia s konkrétnym zvoleným mikrokontrolérom.

Výsledkom práce je po rozobraní knižnice uvedenie a popísanie dvoch praktických príkladov jej využitia. K záveru je uvedené zhodnotenie práce z danou knižnicou a vývojovým prostredím.

Kľúčová slova: mikropočítač, mikrokontrolér, rozhranie pre programovanie aplikácií (API),

ABSTRACT

The submitted thesis deals with a commonly used library "Atmel Software Framework". The aim of the thesis was to make a description of the library for microcomputers from a company Atmel, to demonstrate its function on a suitable computer and in an development tool. This is followed by a manual, after presentation and familiarisation with the library, creation of demo applications. The last but not least aim was to make a final assessment of pros and cons of the library for microcomputers programming.

At the beginning, there is a chronological development of microcomputers presented. From their origins to microcomputers currently used in practice. The thesis deals with modern microcomputers from a company Atmel, their basic categories and with their brief description.

Afterwards, a microcontroller and an development tool were chosen where the thesis and the ASF itself are analyzed.

The description of the library contains division into functions and options of its application with a concrete microcontroller. After analysing the library, the result of the thesis shall be introduction and description of two practical examples of its application.

In the conclusion, there is an assessment of the work with the library and development tool and alternatives of options to develop a program for microcomputers.

Key words: microcomputer, microcontroller, Application programming interface (API)

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD	10
I TEORETICKÁ ČÁST	11
1 DEFINÍCIA A ROZDELENIE MIKROPOČÍTAČOV	12
1.1 PRINCÍP FUNGOVANIA MIKROPROCESOROV	12
1.1.1 Štruktúra mikroprocesorov.....	13
1.2 ROZDELENIE MIKROPOČÍTAČOV A MIKROKONTROLÉROV	13
1.2.1 Mikropočítač	13
1.2.1.1 Štruktúra mikropočítačov	14
1.2.2 Mikrokontrolér	14
1.3 VYUŽITIE MIKROPOČÍTAČOV	14
2 HISTÓRIA	16
2.1 NULTÁ GENERÁCIA	16
2.2 PRVÁ GENERÁCIA	17
2.3 DRUHÁ GENERÁCIA	17
2.4 TRETIA GENERÁCIA	17
2.5 ŠTVRTÁ GENERÁCIA	17
2.6 PIATA GENERÁCIA	17
3 STRUČNÝ PREHĽAD MODERNÝCH MIKROPOČÍTAČOV	18
3.1 MIKROKONTROLÉRY AVR (ATMEL).....	18
3.1.1 Trieda tinyAVR:.....	19
3.1.2 Trieda AVR:.....	19
3.1.3 Trieda megaAVR:	19
3.1.4 Trieda XmegaAVR:	20
3.2 MIKROPROCESORY S ARCHITEKTÚROU ARM OD FIRMY ATMEL	20
3.2.1 ARM – based Flash MCU	20
3.2.2 ARM – based embedded MPU	20
II PRAKTICKÁ ČÁST	21
4 VÝBER VHODNÉHO MIKROPOČÍTAČA A VÝVOJOVÉHO POSTERDIA	22
5 ATMEL SOFTWARE FRAMEWORK (ASF)	23
5.1 ADC – ANALOG TO DIGITAL CONVERTOR	24
5.2 CALENDAR SERVICE	24
5.3 DELAY ROUTINES	26
5.4 GENERIC BOARD SUPPORT	27
5.5 GPIO – GENERAL PURPOSE INPUT / OUTPUT	27
5.6 IOPORT – INPUT / OUTPUT POT CONTROLLER	27
6 POSTUP POUŽÍVANIA ASF	30
6.1 VYTVORENIE NOVÉHO PROJEKTU	30
6.2 KOMUNIKÁCIA S VÝVOJOVOU DOSKOU – ARDUINO UNO	33
7 UKÁŽKOVÉ APLIKÁCIE VYTVORENÉ POMOCOU ASF	36

7.1	APLIKÁCIA – BLIKANIE LED	36
7.2	AD PREVODNÍK	38
8	VÝHODY A NEVÝHODY ATMEL SOFTWARE FRAMEWORK.....	40
	ZÁVĚR	41
	SEZNAM POUŽITÉ LITERATURY.....	42
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	43
	SEZNAM OBRÁZKŮ	44
	SEZNAM TABULEK.....	45
	SEZNAM PŘÍLOH.....	46

ÚVOD

V dobe, kde využívanie elektroniky je dennou súčasťou, sú mikropočítače, ako ich riadiace prvky súčasťou veľkého počtu zariadení. Ich použitie začína od malých USB kľúčov cez mikrovlnné rúry, práčky, mobilné telefóny. Výhodou a hlavným dôvodom je ich kompaktnosť a vysoká univerzálnosť. Základom mikropočítača je mikroprocesor vnútorná pamäť ako aj vstupy a výstupy pre periférie. To ho predurčuje na samostatné fungovanie bez potreby ďalších doplnkov.

Mikropočítače pracujú na základe inštrukcií. Tieto inštrukcie sú programované najčastejšie v jazyku C. Pri starších mikropočítačoch bol využívaný jazyk assembler. V prípade programovania mikropočítačov bez využitia dodatočných knižníc je nutné poznať registre mikroprocesora a správny prístup do pamäte a periférií. Úlohou bakalárskej práce je priblížiť možnosť využitia knižníc na uľahčenie a zrýchlenie písania kódu – programu pre mikropočítače.

Knižnice Atmel Software Framework je dielom poprednej firmy Atmel v oblasti výroby mikropočítačov. Obsahuje funkcie, ktoré sú, už predprogramované pre konkrétne mikropočítače. Tým odpadne starosť pri priradení jednotlivých registrov pre správne fungovanie mikropočítača.

Po krátkom zasvätení do témy mikropočítačov, ich vývoju či histórie, bude vybraný konkrétny mikropočítač. Na tomto mikropočítači demonštratívne bude popísaná knižnica ASF a jej funkcie. Na záver zhodnotenie práce s ňou a porovnanie s inými vývojovými riešeniami.

I. TEORETICKÁ ČÁST

1 DEFINÍCIA A ROZDELENIE MIKROPOČÍTAČOV

Mikropočítače môžeme rozdeliť na niekoľko základných funkčných častí. Je to mikroprocesor, pamäť programu, pamäť dát, periférne obvody. Periférne obvody tvoria veľmi rozmanitú skupinu a ich skladba je závislá na aplikácií mikropočítača. Vzájomné spojenie jednotiek je v mikropočítači zásadne sprostredkované sústavou zberníc. [1]

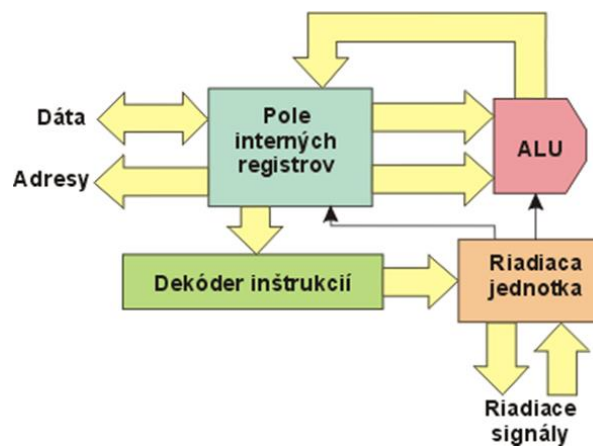
1.1 Princíp fungovania mikroprocesorov

Základnými logickými funkciami sú logický súčet, logický súčin a negácia. V číslicovej elektronike sú tieto funkcie realizované logickými členmi AND, OR a NOT, poprípade ich modifikáciami NOR a NAND. Základným stavebným prvkom logických členov sú polovodičové tranzistory. Vhodnou kombináciou logických členov je možné vytvárať zložitejšie logické funkcie. Mikroprocesor je kompaktnou súčiastkou zlučujúcou v sebe systém logických členov, ktoré realizujú zložitú logickú funkciu. Daná funkcia má viacero vstupných a výstupných premenných v podobe logických signálov. Činnosť mikroprocesora je riadená na základe inštrukcií tvoriacich program. Rovnako ako pri logických členoch je základným stavebným prvkom vnútornej elektronickej štruktúry mikroprocesora polovodičový tranzistor. [2]

1.1.1 Štruktúra mikroprocesorov

Nezávisle od vyššie spomenutých architektúr mikroprocesor obsahuje tieto základné prvky (Obr. 1). [4]

- ALU
- pole interných registrov
- dekodér inštrukcií
- riadiaca jednotka
- zbernice



Obr. 1 Schéma mikroprocesora [4]

1.2 Rozdelenie mikropočítačov a mikrokontrolérov

Na základe dodatočných periférnych zariadení, ktoré pridáme k mikroprocesoru môžeme rozdeliť:

1.2.1 Mikropočítač

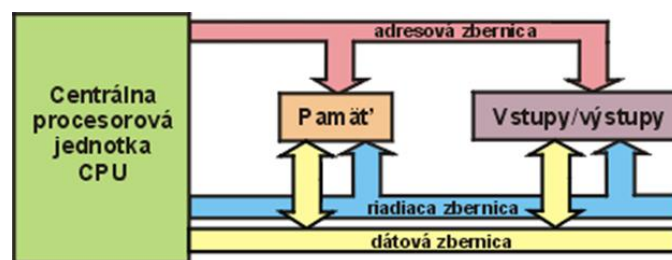
Jedno čipový mikropočítač - vznikne ak sa mikroprocesor doplní o periférne obvody (pamäte, obvody vstupov / výstupov a iné rozhrania) a spolu sú umiestnené do jedného puzdra. Jedno čipový mikropočítač disponuje určitou univerzálnosťou a možnosťou modifikovať jeho funkciu na základe programu, prípadne rozšírením o ďalšie periférne obvody. [5]

1.2.1.1 Štruktúra mikropočítačov

Mikropočítač rovnako ako bežný počítač pozostáva z troch základných blokov integrovaných do jedného celku:

- mikroprocesor – nazývaný aj centrálna procesorová jednotka (CPU)
- pamäť – systém dátovej a programovej pamäte
- vstupy / výstupy – umožňujú pripojiť doplnujúce zariadenia (ďalšiu pamäť a podobne)

Tieto bloky sú navzájom prepojené systémom zberníc (Obr. 2). [4]



Obr. 2 Schéma mikropočítača [4]

1.2.2 Mikrokontrolér

Jedno čipový mikropočítač určený pre špecifickejšie nasadenie, napríklad v obvode inteligentného snímača, kde je navrhnutý pre plnenie určitých konkrétnych úloh, čím klesá na univerzálnosť v porovnaní s klasickým jedno čipovým mikropočítačom. Je možné povedať, že mikrokontrolér je užšie špecializovaný, menej univerzálny jedno čipový mikropočítač. [5]

1.3 Využitie mikropočítačov

Hlavné využitie mikropočítačov je v elektronických systémoch, kde je potrebné jeden alebo viacero veličín či už riadiť alebo vyhodnocovať. V minulosti sa využívali analógové obvody na zabezpečenie týchto potrieb. Rozvojom techniky prišli digitálne systémy, ktoré ponúkli viacej univerzálnosti a cenovo dostupnejšie riešenia. Hlavný rozdielom medzi analógovými a digitálnymi systémami je vnímanie a spravovanie signálu. Zatiaľ čo analógové sú schopné signál vnímať spojito v čase, tak digitálne systémy vnímajú signály ako diskkrétne hodnoty v jednotlivých časových jednotkách. Preto ak sa pozrieme do praxe je potrebné aby sme vedeli zmeniť analógový signál zmeniť na digitálny, resp. signál v čase

spojitý rozdeliť a diskretný. Pre lepšiu predstavu týchto dvoch rozdielnych možností technického riešenia pri konštrukcii automatizovaných systémoch nižšie uvediem hlavné výhody / nevýhody oboch možností. [2]

Analógové:

výhody:

- priame napojenie systému na spracovávanú analógovú veličinu
- vyššia rýchlosť spočívajúca v paralelnom spracovaní informácie analógovým systémom

nevýhody:

- nízka presnosť
- nestálosť parametrov analógového obvodu z dôvodu starnutia komponentov, teplotná závislosť parametrov obvodu
- pri pozmenení funkcie obvodu je nutná prestavba celého obvodu, nízka univerzálnosť
- niektoré zložité funkcie sa dajú realizovať iba s veľkou ťažkosťou

Digitálne:

výhody:

- presnosť a časová stálosť
- univerzálnosť
- zlučiteľnosť s prostriedkami výpočtovej techniky (pripojenie k PC umožňuje narábanie s veličinami, automatizácia spracovania veličín)
- hromadná výroba znížila cenu mikroprocesorov a obvodov s nimi súvisiacich

nevýhody:

- nižšia rýchlosť procesorových systémov v prípade, že spracovávaná veličina je veličinou veľmi rýchlo sa meniacou
- nutnosť pripojiť prevodníky AD a DA, ktoré sú do určitej miery zdrojom nepresností prevodu. [2]

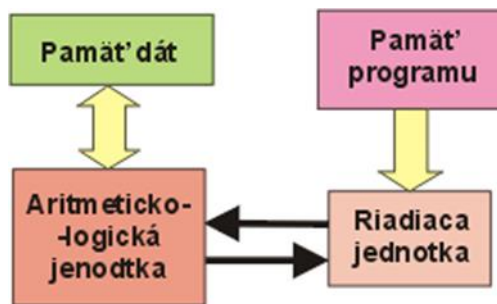
2 HISTÓRIA

Vývoj mikroprocesora môžeme rozdeliť na etapami. Na začiatku boli hlavnou stavebnou jednotkou procesorov elektronické relé, elektrónky a neskôr polovodičové tranzistory.

2.1 Nultá generácia

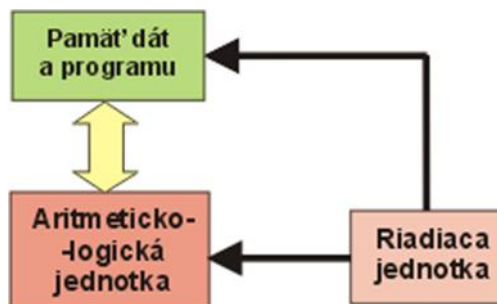
Prvými základnými stavebnými prvkami mikroprocesor boli mechanické systémy, elektromechanické relé alebo elektrónky.

V roku 1944 bola charakterizovaná prvá architektúra nazývaná harvardská. Architektúra je charakteristická svojou oddelenou pamäťou pre dáta a pre program (Obr. 3). Dátová pamäť je pripojená obojsmernou a programová pamäť jednosmernou dátovou zbernicou. Výhodou harvardskej architektúry je vyššia rýchlosť systému, keď pri výbere inštrukcie z programovej pamäte môže súčasne prebiehať aj presun dát z dátovej pamäte. [2]



Obr. 3 Harvardská architektúra [2]

Dlho na seba nenechala čakať ďalší možnosť usporiadania jednotlivých komponentov. John Von Neumann roku 1946 zadefinoval novú architektúru počítačového systému pracujúceho v dvojkovej sústave. Je používaná dodnes. Charakteristickou črtou tejto architektúry je spoločná pamäť pre dáta aj program (Obr.4). Nevýhodou je, že v spoločnej pamäti sídlia dáta a program súčasne. Obojsmerná spoločná zbernica je využitá pre dáta a inštrukcie. [2]



Obr. 4 Architektúra Von neumann [2]

2.2 Prvá generácia

Používa na konštrukciu procesora ako základný konštrukčný prvok elektrónky. [2]

2.3 Druhá generácia

Využíva základný stavebný prvok tranzistor. Vznik prvého integrovaného obvodu roku 1958. [2]

2.4 Tretia generácia

Integrovaný obvod, táto nová podoba elektronických súčiastok bol počiatkom nástupu. Prvý mikroprocesor vyrobila firma Intel roku 1971 . Intel 4004 - 4-bitový procesor s výkonom 60000 operácií za sekundu a 2500 tranzistorami na čipe s rozlohou 4 mm². Intel svoj vylepšený 8-bitový Intel 8080 schopný vykonať 29000 operácií za sekundu a adresovať 64 kB pamäte. Roku 1980 firma Intel vyrobila 8-bitový mikropočítač Intel 8051 združujúci v sebe mikroprocesor, pamäte RAM a ROM, periférne obvody ako sériový port a 4 univerzálne paralelné vstupne/výstupné porty. [2]

2.5 Štvrtá generácia

Zmenšením rozmerov narástol výkon a klesol príkon mikroprocesorov. Prvý 32 bitový procesor Intel uviedol na trh až v r. 1985 pod označením Intel 386 s taktovacou frekvenciou od 16 MHz po 40 MHz. [2]

2.6 Piata generácia

Roku 1993 vyrobil 32-bitový mikroprocesor Intel Pentium, firma AMD s procesormi AMD K5. Do oblasti jedno čipových mikropočítačov a univerzálnych mikroprocesorov okrem známej architektúry Intel 8051 vstupujú aj iné 8, 16 a aj 32-bitové architektúry (napríklad 8-bitové Atmel AVR uvedené roku 1993 a vyrábané dodnes, ARM, Motorola). [2]

3 STRUČNÝ PREHLAD MODERNÝCH MIKROPOČÍTAČOV

V tejto kapitole budú predstavené mikropočítače od firmy Atmel. Popíšem ich základné črty a rozdelenie do rodín. Ako dve základné skupiny sú vybrané mikrokontroléry AVR a mikropočítače s architektúrou ARM.

3.1 Mikrokontroléry AVR (ATMEL)

AVR sú mikrokontroléry s harvardskou architektúrou a RISC jadrom umožňujúcim výkon inštrukcií počas jedného hodinového cyklu. Konštrukčne sú plne statické, umožňujú činnosť od nulovej frekvencie po maximálnu v závislosti od daného typu. Inštrukcie sú vytvorené tak, aby minimalizovali veľkosť programu, či už bol písaný v assembleri alebo jazyku C. [8]

Tieto mikrokontroléry disponujú navyše funkciou ISP (In System Programming) – čiže možnosť programovania obvodu priamo v aplikácii. Bohatá štruktúra obsiahnutých periférií znižuje nutnosť pripojenia externých obvodov. [3]

Rodinu mikrokontrolérov tvoria AVR (klasické AVR), tinyAVR, megaAVR a XmegaAVR. Rozdiel medzi radami je vo vybavení dodatočnými perifériami a inými možnosťami. [3] TinyAVR predstavujú najjednoduchšie riešenie s minimom dodatočných integrovaných zariadení a malým počtom vývodov, hodiace sa tak pre jednoduché aplikácie. Klasické AVR predstavujú medzičlánok medzi tinyAVR a megaAVR. MegaAVR sú bohatšie vybavenou triedou s dodatočnými perifériami. Najvyššia trieda XmegaAVR je najlepšie vybavenou triedou AVR mikrokontrolérov. Jednotlivé triedy mikrokontrolérov sú smerom hore inštrukčne kompatibilné, majú rovnaký pamäťový model, takže prechod medzi typmi navzájom, je bezproblémový. Mikrokontroléry vyšších rodín sú svojou vybavenosťou a univerzálnosťou pre masovejšie využitie v praxi. Hlavným kritériom pri výbere je zváženie využitia mikrokontroléra a jeho cena.

3.1.1 Trieda tinyAVR:

Atmel tinyAVR zariadenia sú navrhnuté špeciálne pre aplikácie, kde je dôležitá veľkosť a cena. Najmenší tinyAVR meria iba 1,5 mm x 1,4 mm. Možnosť kapacitného dotyku - Atmel QTouch knižnica uľahčí vložiť dotykové kapacitné tlačidlo alebo jazdca do aplikácie mikrokontroléra. Royalty-free QTouch knižnica podporuje rôzne množstvo dotykových kanálov, čím umožňuje flexibilitu a efektivitu pri dotykovo zameraných aplikáciách.

Silnou stránkou tinyAVR je vysoká integrácia. Každý pin má mnohostranné využitie ako vstup alebo výstup a AD prevodník. Rodina tinyAVR je vybavená Universal Serial Interface (USI), ktorý môže byť použitý ako SPI, UART alebo TWI. Devíza triedy tinyAVR je ako už spomínaná veľkosť ale hlavne spotreba energie. Dokáže pracovať už s napätím 0,7V.

3.1.2 Trieda AVR:

Táto trieda je významná ako pre svoju veľkú škálu možností pripojenia. Disponuje s USB rozhraním, Ethernet MAC, SDRAM, NAND Flash. Veľmi rýchle sériové rozhranie sú ideálne pre zložitejšie aplikácie. Aritmetický výkon pomocou integrovaného FPU zvyšuje presnosť a dynamický rozsah v digitálnom spracovaní signálov. Nízka spotreba energie - technológia picoPower prináša nižšiu spotrebu energie pri aktívnom využívaní ako aj tzv. spánku mikrokontroléra.

3.1.3 Trieda megaAVR:

Trieda megaAVR ponúka najširší výber mikrokontrolérov. Obrovská rozmanitosť na základe periférnych zariadení, umožňuje použitie rovnakého kódu vo viacerých mikrokontroléroch. O nízky odber energie sa stará picoPower technológia. Vybrané megaAVR mikrokontroléry využívajú túto technológiu v spojení s voliteľným „low - power“ režimom spánku. To ich predurčuje na vhodné použitie s batériovým napájaním. Vysoká integrita triedy megaAVR ponúka SRAM, EEPROM, SPI, TWI (I2C), a USART, USB, CAN a LIN, watchdog timer. Pokročilé analógové funkcie, ako je napríklad AD a DA prevodník, Vysoká úroveň tejto integrity napomáha k ľahšiemu využívaniu pri potrebe pracovať s analógovými hodnotami. Rýchly vývoj aplikácie pomocou výkonného In-System Programming a on-chip debug. Tieto megaAVR zariadenia patria medzi významné mikroprocesory, pokiaľ ide o spotrebu energie a voľbou pre aplikácie.

3.1.4 Trieda XmegaAVR:

Najvyššia trieda AVR mikrokontrolérov disponuje vysoko presným AD a DA 12-bitovým prevodníkom. Technológia Atmel picoPower sa stará ako aj u nižších triedach o nízku spotrebu energie. AVR Xmega zariadenia dokážu integrovať šifrovanie Advanced Encryption Standard (AES) a Data Encryption Standard (DES). MCU obsahuje PWM moduláciu UART, TWI (I2C), SPI kanály. AVR Software Library zabezpečuje knižnicu ovládačov zariadení a komunikačných modulov. Tým šetrí čas a náročnosť vývoja aplikácie.

3.2 Mikroprocesory s architektúrou ARM od firmy Atmel

Firma Atmel využíva túto architektúru na mikrokontroléry SMART. Tieto mikrokontroléry môžeme rozdeliť do dvoch základných skupín. ARM – based Flash mikrokontroléry a ARM –based embedded mikroprocesory.

3.2.1 ARM – based Flash MCU

Táto skupina mikrokontrolérov s skladá: ARM Cortex-M4, Cortex-M3, Cortex-M0+, ARM926EJ-S a ARM7TDMI. Mikrokontroléry ponúkajú od 8kB Flash pamäte a 4 kB SRAM až po 2 MB Flash pamäte a 160 kB RAM, prevádzkových frekvenciu až 120 MHz. Priamy prístup do pamäte (DMA). "Touch-ready" schopnosť komunikácie kapacitnou dotykovou technológiou cez Atmel QTouch a hardvérovou akceleráciou v rodine SAM D. Sériovo komunikačný modul (SERCOM), k dispozícii v rodine SAM D, je plne softvérovo konfigurovateľný zvládnuť I2C, USART / UART, SPI a komunikácie pre zvýšenie flexibility rozhrania. Najnižšia spotrebu energie nájdeme v mikrokontroléry ARM Cortex-M4.

3.2.2 ARM – based embedded MPU

Využívané sú mikroprocesory: ARM Cortex-A5 a ARM926EJ-S. Tieto riešenia sú cenovo prijateľné, ponúkajú 536MHz. Výpočet plávajúcej čiarky (FPU), maximálny odber energie je 150 mW pričom minimálny potrebný na prevádzku je iba 0,5mW. Obsahujú USB konektivitu, Ethernet, CAN, TFT LCD radič a ďalšie iné možnosti.

II. PRAKTICKÁ ČÁST

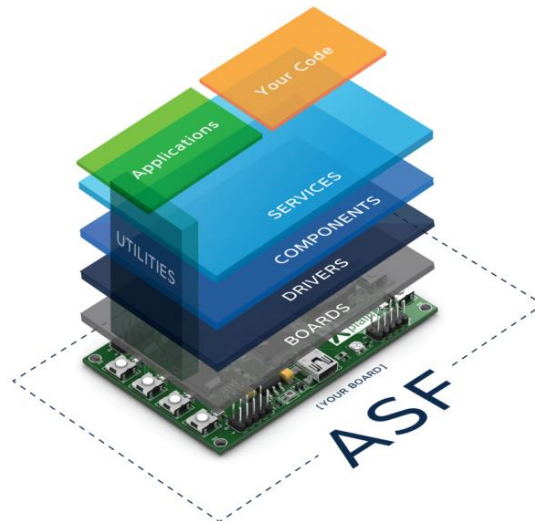
4 VÝBER VHODNÉHO MIKROPOČÍTAČA A VÝVOJOVÉHO POSTERDIA

Pre popis ASF je vybraný mikrokontrolér ATmega 328P. Je z rodiny mega AVR. Vhodný bol hlavne z možnosti použitia ASF pri jeho programovaní. Ďalšou devízou a hlavným dôvodom pre vybranie tohto konkrétneho mikropočítača bola dostupnosť a možnosť zapožičania školou. Tento mikrokontrolér je umiestnený na vývojovej doske Arduino Uno. Táto vývojová doska uľahčuje nahrávanie samotného programu do mikrokontroléra pomocou bootloaderu, ktorý je obsiahnutý vo vývojovej doske.

Vývojové prostredie Atmel Studio je vybrané z dôvodu spolupráce s knižnicou ASF. Táto knižnica je priamo v prostredí Atmel Studia implementovaná. Dodatočná inštalácia do iného vývojového prostredia by bola komplikácia. Táto možnosť existuje je však náročná. Práca s knižnicou ako aj s programom samotným je intuitívna. Taktiež výber a vloženie jednotlivých knižníc do projektu je jednoduché a rýchle.

5 ATMEL SOFTWARE FRAMEWORK (ASF)

ASF poskytuje ovládače a knižnice pre mikropočítače tinyAVR, AVR, megaAVR, XmegaAVR a SAM. Pomáha pri vývoji aplikácie a vkladání rôznych komponentov. Použitie knižníc ASF skraca čas aj náročnosť pri vývoji aplikácie pre mikropočítače. Nižšie je zobrazená štruktúra ASF (Obr. 5)



Obr. 5 Štruktúra ASF [7]

Ako je na obrázku vidieť, ASF je rozdelené do častí, ktoré má každá svoju úlohu.

Boards – poskytuje mapovanie všetkých digitálnych a analógových vstupov a výstupov. ASF obsahuje túto možnosť len pre vývojové dosky od firmy Atmel. [7]

Drivers – skladá sa zo súborov driver.c a driver.h, ktoré nám umožňujú ľahší prístup k „low level register“ pre periférne zariadenia. [7]

Components – je časť ASF, ktorá poskytuje softwarové ovládače pre prístup k externým hardwarovým komponentom. To môžu byť napríklad pamäte, displeje, senzory. [7]

Services – časť, ktorá sa stará o aplikačne orientovaný software ako súborové systémy, USB periférie, grafické knižnice. [7]

Možnosť využitia ASF sa líši, závisí od použitého konkrétneho mikrokontroléra. Pre túto bakalársku prácu bol vybraný mikrokontrolér ATmega 328P. Nižšie je uvedená tabuľka s dostupnými API pre tento konkrétny mikrokontrolér (Tab. 1).

Kategória	Názov
Analog Conversion	ADC - Analog to Digital Converter
	Calendar functionality
CPU	CPU specific features
	Delay routines
	Generic board support
I/O	GPIO - General purpose Input/Output
Interrupt	Interrupt management - megaAVR and tinyAVR implementation
I/O	IOPORT - Input/Output Port Controller
Clocks	System Clock Control - MEGA RF A1 implementation
	Unit test framework - MEGA implementation

Tab. 1 API obsahované v ASF pre ATmega328P

5.1 ADC – Analog to Digital Converter

ADC ovládač obsahuje funkcie na prevod analógovej hodnoty na digitálnu. Mikro-počítač nevie pracovať s analógovou hodnotou, ktorá je použitá v systéme. AD prevodník tak umožňuje pracovať mikropočítaču s digitálnou hodnotou získanou so systému. Nižšie popisujem základné funkcie ovládača.

Inicializácia ovládača s nastavením preslacera:

```
adc_init(ADC_PRESCALER_DIV32);
```

Načítanie z 8 bitového prevodníku z channel 0 použitím 1.1V napätia:

```
analog_val = adc_read_8bit(ADC_MUX_ADC0, ADC_VREF_1V1);
```

Načítanie z 10 bitového prevodníku z channel 0 použitím 1.1V napätia:

```
analog_val = adc_read_10bit(ADC_MUX_ADC0, ADC_VREF_1V1);
```

5.2 Calendar service

Poskytuje služby umožňujúce prevod Unix Time Stamp na dátum, Unix Time Stamp takzvaný Unixový čas používajú hlavne operačné systémy založené na Unixe ale aj iné operačné systémy a súborové formáty. Tento časový formát je vyjadrením počtu sekúnd od konkrétneho dátumu a času. Výpočet sa začal od 1. Januára 1970 v čase 00:00:00, Premenná vyjadrujúca čas a dátum sa každú sekundu zväčšuje o jednu. Ďalšou službou je

5.3 Delay routines

Ovládač vedomého a želaného zdržania programu. Pomocou tohto ovládača vieme zabezpečiť, že na dobu nami učenú nebude mikroprocesor vykonávať žiadnu ďalšiu inštrukciu kódu. Jednotlivé funkcie s ich stručným popisom som uviedol nižšie.

Inicializácia funkcie:

```
delay_init();
```

Definovanie frekvencie CPU:

```
#define F_CPU sysclk_get_cpu_hz();
```

Funkcie:

zdržanie, kde čas zadávame v mikrosekundách:

```
delay_us(delay);
```

zdržanie, kde čas zadávame v milisekundách:

```
delay_ms(delay);
```

zdržanie, kde čas zadávame v sekundách:

```
delay_s(delay);
```

5.4 Generic board support

Firma Atmel pri tvorbe ASF myslela aj na svoje vývojové dosky, preto do ASF vložila aj ovládač pre lepšiu prácu s nimi. V tabuľke (Tab. 4), ktorá sa nachádza v prílohe, som uviedol podporované vývojové dosky aktuálnym ASF. Ovládač obsahuje napríklad zadenovanie portov vstupov a výstupov jednotlivých vývojových dosiek.

Inicializácia:

```
board_init();
```

5.5 GPIO – General purpose Input / Output

V tomto ovládači nájdeme kompletne nastavenie vstupov a výstupov mikrokontroléra. Tieto vstupy a výstupy sú používané pre periférie mikrokontroléra. Napríklad LED, tlačidlá ale aj AD / DA prevodníky. Táto knižnica je hlavne využívaná ďalšími službami obsiahnutými v ASF.

5.6 IOPORT – Input / Output Pot Controller

Služba pre štandardnú konfiguráciu a riadenie I/O portov. Umožňuje nám svojimi funkciami podrobnejšie nastavenie jednotlivých pinov a portov. Tieto jednotlivé funkcie som bližšie popísal nižšie v texte.

Inicializácia:

```
ioport_init();
```

Zadefinovanie pinu:

```
#define MY_LED      IOPORT_CREATE_PIN(PORTA, 5)
#define MY_BUTTON  IOPORT_CREATE_PIN(PORTA, 6)
```

Funkcie pre PIN:

Nastavenie pinu MY_LED na výstup:

```
ioport_set_pin_dir(MY_LED, IOPORT_DIR_OUTPUT);
```

Nastavenie pinu MY_BUTTON na vstup:

```
ioport_set_pin_dir(MY_BUTTON, IOPORT_DIR_INPUT);
```

Nastavenie pull up rezistora na pine:

```
ioport_set_pin_mode(MY_BUTTON, IOPORT_MODE_PULLDOWN);
```

Nastavenie pull down rezistora na pine:

```
ioport_set_pin_mode(MY_BUTTON, IOPORT_MODE_PULLUP);
```

Práca s hodnotou z pinu:

```
bool hodnota;
```

Získanie hodnoty:

```
hodnota = ioport_get_pin_level(MY_BUTTON);
```

Priradenie hodnoty:

```
ioport_set_pin_level(MY_LED, hodnota);
```

Reštart módu pinu:

```
ioport_reset_pin_mode(ioport_pin_t pin);
```

Funkcie pre PORT:

Nastavenie portu na výstup:

```
ioport_set_port_dir(ioport_port_t port, ioport_mask_t mask,  
                    IOPORT_DIR_OUTPUT);
```

Nastavenie portu na vstup:

```
ioport_set_port_dir(ioport_port_t port, ioport_mask_t mask,  
                    IOPORT_DIR_INPUT);
```

Nastavenie pull up rezistora na porte:

```
ioport_set_port_mode(ioport_port_t port, ioport_mask_t mask,  
                    IOPORT_MODE_PULLUP);
```

Nastavenie pull down rezistora na porte:

```
ioport_set_port_mode(ioport_port_t port, ioport_mask_t mask,  
                    IOPORT_MODE_PULLDOWN);
```

Práca s hodnotou z portu:

```
ioport_port_mask_t hodnota;
```

Získanie hodnoty:

```
hodnota = ioport_get_port_level(ioport_port_t port,  
                                ioport_mask_t mask);
```

Priradenie hodnoty:

```
ioport_set_port_level(ioport_port_t port, ioport_mask_t mask,  
                      hodnota);
```

Reštart módu portu:

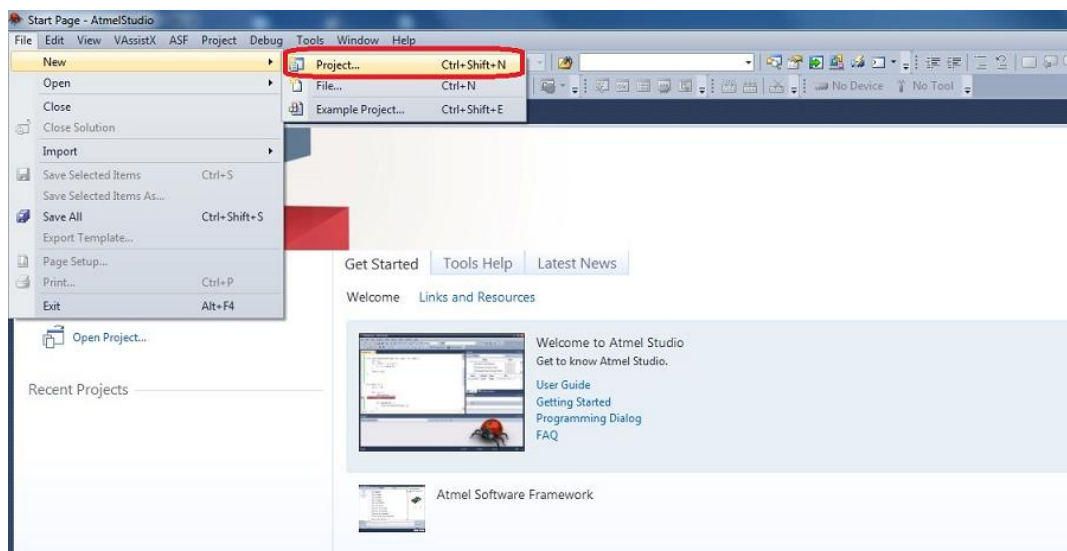
```
ioport_reset_port_mode(ioport_port_t port,  
                       ioport_mask_t mask);
```

6 POSTUP POUŽÍVANIA ASF

K efektívnemu a jednoduchému využitiu ASF potrebujeme mať nainštalovaný Atmel Studio 6 a jeden z podporovaných mikrokontrolérov z rodín tinyAVR, megaAVR, AVR XMEGA, AVR UC3 alebo SAM.

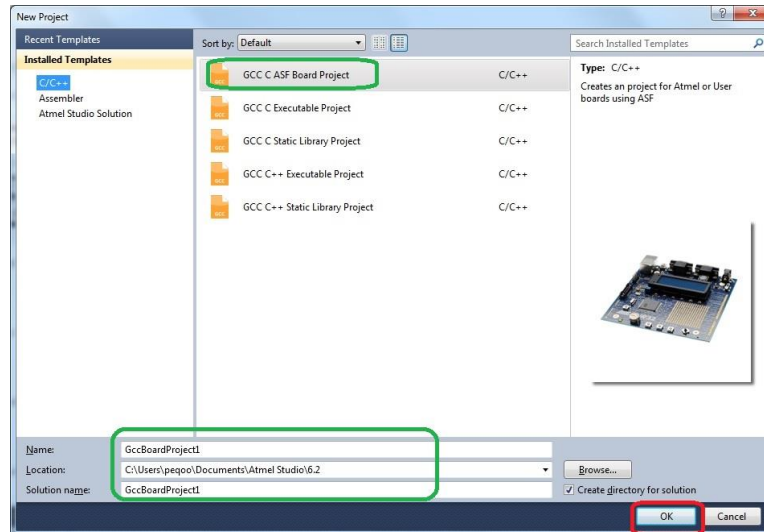
6.1 Vytvorenie nového projektu

Začneme stlačením ponuky File → „New“ → „Projekt“ (Obr. 5). Podobne môžeme použiť skratku „Ctrl+Shift+N“.



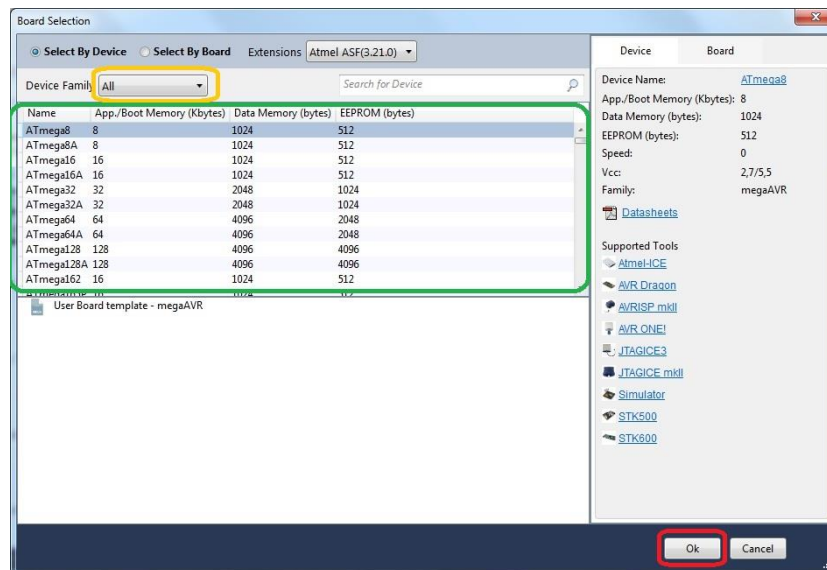
Obr. 6 Vytvorenie nového projektu

Pre ilustráciu som použil len výrez daného okna aby bolo zreteľne a viditeľne vidieť postup výberu. Pokračujeme výberom typu nového projektu. pre naše použitie vyberieme prvú voľbu označenú zeleným políčkom „GCC C ASF Board Project“. Druhé zelené políčko označuje možnosti zvolenia názvu, umiestnenia projektu a názov riešenia (názov projektu a riešenia sa obvykle zhodujú). Po výbere všetkých parametrov pokračujeme tlačidlo „OK“ (Obr. 6).



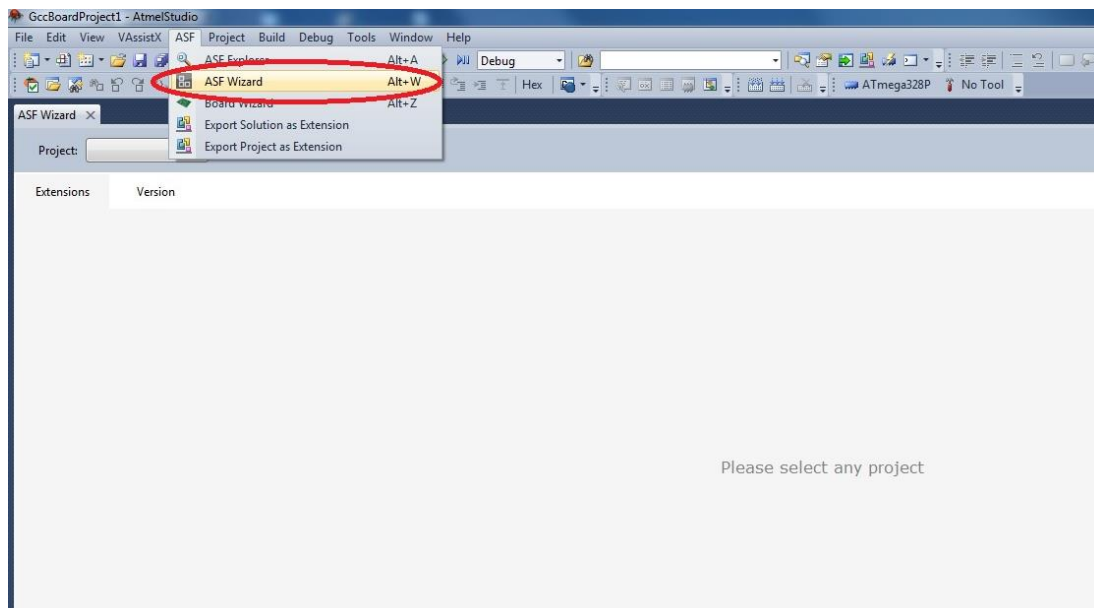
Obr. 7 Nastavenie nového projektu

V nasledujúcom okne vyberieme mikropočítač na ktorú budeme aplikovať vytvorený program. výber potvrdíme tlačidlo „OK“ (Obr. 7).



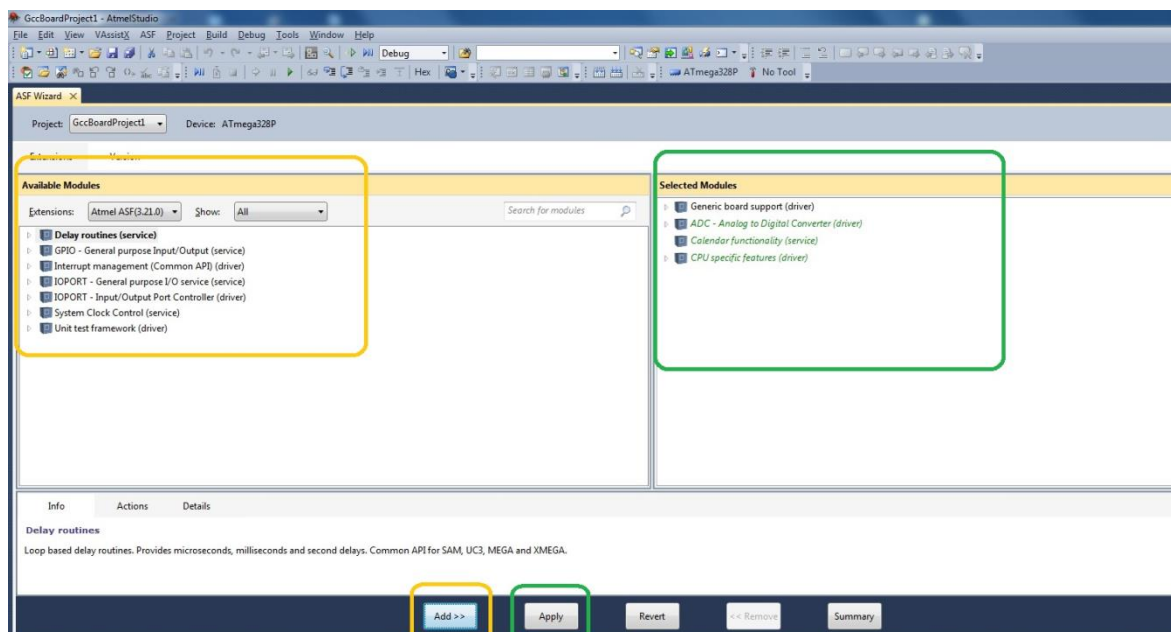
Obr. 8 Výber dosky

Ďalej musíme vybrať súčasti Atmel Software Framework, ktoré budú implementované do nášho projektu. Preto zvolíme ponuku „ASF“ → „ASF Wizard“ (Obr. 8).



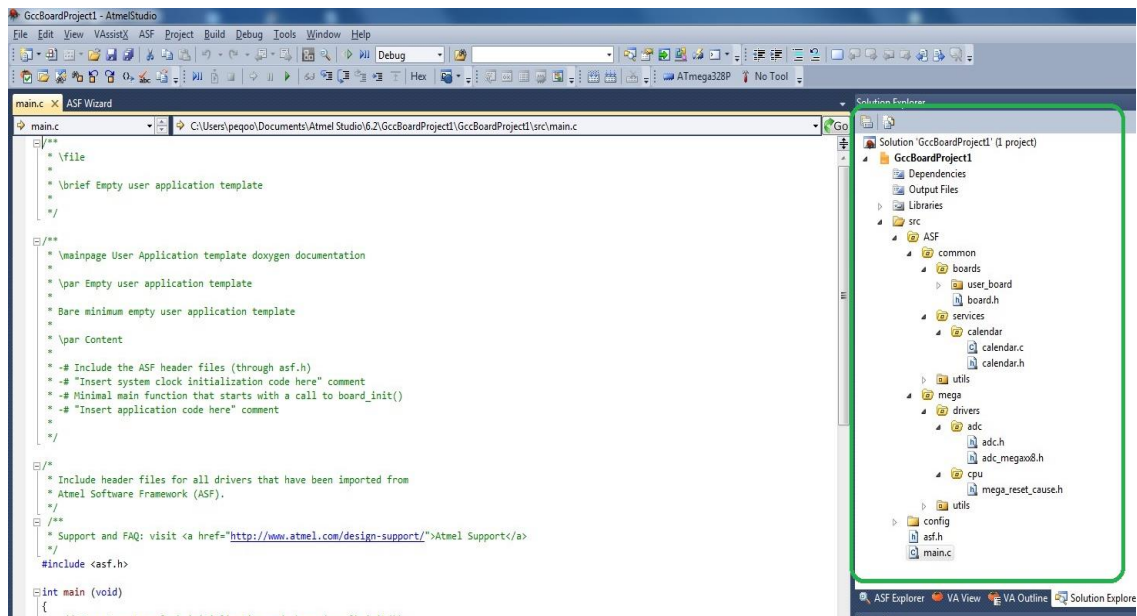
Obr. 9 Vloženie knižovien ASF

Otvorí sa nám okno, v ktorom vidíme v žltom ráme možnosti knižníc ASF pre našu konkrétnu vybranú dosku. Tlačidlom „Add“ pridáme zvolenú knižnicu do zoznamu v zelenom ráme. V tomto zozname môžeme vidieť už implementované knižnice čiernou farbou písma a knižnice, ktoré budú implementované zelenou farbou písma. Implementáciu týchto knižníc potvrdíme tlačidlom „Apply“ v zelenom políčku (Obr. 10).



Obr. 10 Implementácia knižníc do projektu

Po dokončení implementovania knižníc do nášho projektu, môžeme začať písať kód. Knižnice už budú funkčné. Na pravej strane obrázku (Obr. 11) v zelenom ráme môžeme vidieť všetky súbory projektu, tak aj hlavičkové súbory jednotlivých knižníc, ktoré sme vložili do projektu.

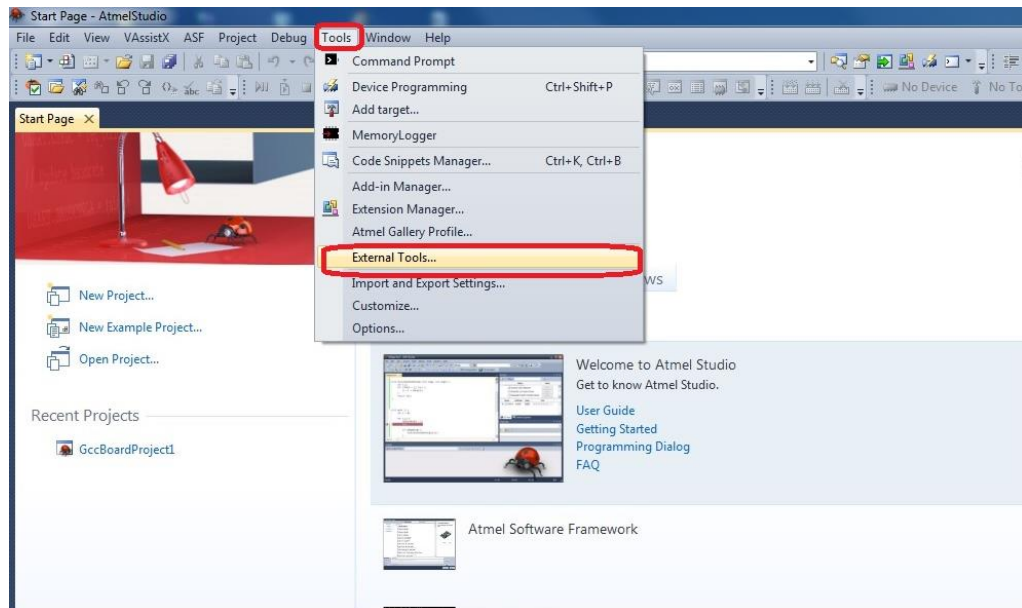


Obr. 11 Vložené hlavičkové súbory v projekte

6.2 Komunikácia s vývojovou doskou – Arduino Uno

Pre nahrávanie vytvoreného kódu do konkrétneho mikrokontroléra ATmega 328P, ktorý je umiestnený na vývojovej doske Arduino Uno je dôležité nastaviť Atmel Studio. Potrebný je taktiež program avrdude. Ten je súčasťou Arduino Software, preto ho použijeme s ním. Je bezplatne dostupný na stránke <http://www.arduino.cc/en/Main/Software>. Doporučené je stiahnuť ZIP verziu a komprimovaný súbor rozbaľiť do zvlášť zložky. Najlepšie priamo na disk kde je umiestnený operačný systém.

Pripojením mikropočítača s vývojovou doskou k počítaču, sa automatika nainštaluje ovládač. Po dokončení otvoríme Atmel Studio. Ponuka Tools → External Tools (Obr. 12).



Obr. 12 Inštalácia externej vývojovej dosky

Zobrazí sa okno s nastavením externej vývojovej dosky. Postupu podľa obrázku (Obr. 18). V žltom obdĺžniku je názov tohto zariadenia, môžeme ho ľubovoľne podľa svojho uváženia zmeniť. V zeleno označeného obdĺžnika je vyplnené

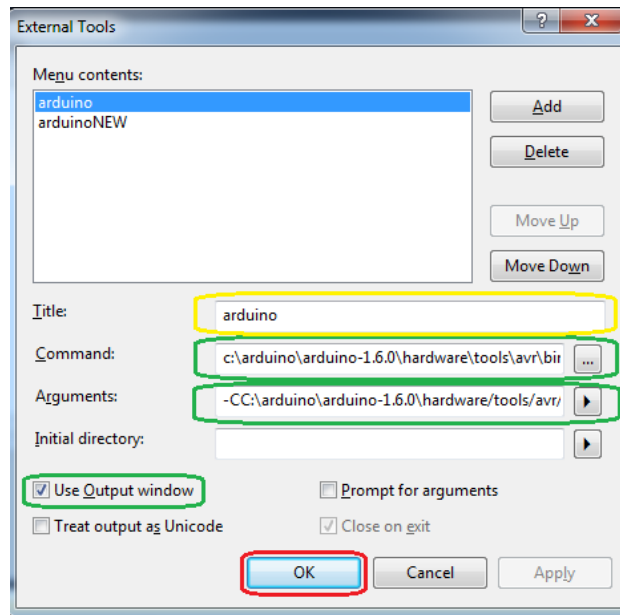
Command: c:\arduino\arduino-1.6.0\hardware\tools\avr\bin\avrdude.exe
cesta sa môže líšiť podľa umiestnenia stiahnutého programu Arduino.

Arguments: -CC:\arduino\arduino-1.6.0\hardware/tools/avr/etc/avrdude.conf -v -v

-patmega328p -carduino -P\\.\COM3 -b115200 -D

-Uflash:w:"\$(ProjectDir)Debug\\$(ItemFileName).hex":i

Tak ako aj Command aj Arguments musí byť upravené na konkrétne umiestnenie avrdude. Dôležité je aj zistiť aký port priradil počítač vývojovej doske po jej pripojení. Pre zobrazenie priebehu nahrania kódu do mikrokontroléra je zaškrtnutý aj posledný zelený obdĺžnik „Use Output window“ (Obr. 13).



Obr. 13 Nastavenie externého zariadenia

Po dokončení kliknutím na červený obdĺžnik „OK“ je nastavené nahrávanie do vývojovej dosky.

7 UKÁŽKOVÉ APLIKÁCIE VYTVORENÉ POMOCOU ASF

Pre lepšie praktické načrtnutie funkcií ASF sú vytvorené dve aplikácie. Prvou je rozblikanie LED diódy na vývojovej doske, druhou je AD prevodník.

7.1 Aplikácia – blikanie LED

Prvá ukážka je rozblikanie diódy. K úspešnému napísaniu kódu je potrebné pár technických údajov o vývojovej doske a umiestnení jej periférií. Pre získanie týchto údajov je využitá priamo stránka výrobcu mikrokontroléra. Potrebné je zistiť a použiť v kóde rozmiestnenie portov a pinov na vývojovej doske a prístup k nim. Pre nastavenie pinov využívam knižnicu IOPORT - Input/Output Port Controller. Ak chceme aby dióda blikala s nastaveným konkrétnym časom potrebujeme na to príslušnú funkciu Delay routines.

Nižšie vidíme obrázok (Obr. 14) s kódom mojej aplikácie. Pod obrázkom uvediem aj presný popis tohto kódu s využitými knižnicami a funkciami.

```
1  #include <asf.h> // vloženie knižnice ASF
2  #define MY_LED   IOPORT_CREATE_PIN(PORTB, 5) // definovanie LED
3
4
5  int main (void)
6  {
7      sysclk_init(); // inicializacia system clock
8      ioport_init(); // inicializacia funkcii z knižnice ioport
9
10
11
12     ioport_set_pin_dir(MY_LED, IOPORT_DIR_OUTPUT); // nastavenie pinu na vystup
13
14     ioport_set_pin_level(MY_LED, false); // nastavenie počiatočnej hodnoty pinu
15
16     while(1)
17     {
18
19         ioport_set_pin_level(MY_LED, true); //nastavenie diódy - LED svieti
20         delay_s(1); //pozastavenie cinnosti procesora
21         ioport_set_pin_level(MY_LED, false); //nastavenie diódy - LED nesvieti
22         delay_s(1); //pozastavenie cinnosti procesora
23
24     }
25 }
26
```

Obr. 14 Kód aplikácie - blikanie LED

Na začiatku kódu je vloženie knižnice ASF. Táto knižnica obsahuje vloženie všetkých knižníc ASF, ktoré sú implementované do nášho projektu. Následne je definovaná MY_LED pomocou vytvorenia nového pinu, resp. priradenie názvu ku konkrétnemu pinu,

ktorý predstavuje LED. To, na ktorom pine sa nachádza dióda je individuálne od využívania konkrétnej vývojovej dosky, to je potrebné dohľadať v príslušnej dokumentácii.

V hlavnom programe main sú najprv inicializované použité knižnice. Sysclk_init() inicializuje knižnicu System Clock Control. Tá je použitá pri určení frekvencie CPU, potrebnej pri časovom pozastavení práce procesora, následne ioport_init() pre fungovanie funkcií pri práci s pinami. Nastavenie pinu MY_LED na výstup je dôležité k informovaniu mikrokontroléra o jeho použití. Následne je nastavená počiatočná hodnota LED na „false“, čiže nesvietiť. V nekonečnom cykle striedavo nastavujeme hodnotu MY_LED na „true“ (LED sa rozsvieti) a po uplynutí časového zdržania procesora opäť na hodnotu „false“ (LED zhasne). Tento kód sa v cykle bude opakovať. Tým je zabezpečené blikanie LED v aplikácii, ktorá diódu rozsvieti na dobu jednej sekundy a zhasne na dobu jednej sekundy.

7.2 AD převodník

Pre druhú ukážku ASF je znázornený vzorový príklad s využitím Analog to Digital Converter. Po vložení príslušnej ADC knižnice do projektu je nižšie popísaný kód aplikácie (Obr. 15).

```
1  #include "usart.h"
2  #include <asf.h>
3  #include <stdlib.h>
4
5  int main(void) {
6
7      USART_init(); // inicializacia funkcii kniznice usart.h
8      adc_init(ADC_PRESCALER_DIV32); // inicializacia ADC prevodnika s nastavenim prescalera
9
10     char slovo [5];
11     char slovoV[2];
12     char slovoVzv[2];
13     int cislo;
14     int cisloV;
15     int cisloVzv;
16
17     while(1){
18
19         cislo = adc_read_10bit(ADC_MUX_ADC0,ADC_VREF_AVCC); //pridanie hodnoty z prevodnika do premennej
20
21         cisloV = cislo / 204; // vypocet celej casti hodnoty napatia
22         cisloVzv = (cislo % 204) / 21; // vypocet desatinnej casti hodnoty napatia
23
24         // prevod datoveho typu integer na retazec char (string)
25         itoa(cislo,slovo,10);
26         itoa(cisloV,slovoV,10);
27         itoa(cisloVzv,slovoVzv,10);
28
29         // vypis na seriovu linku
30         USART_putstring("10bit hodnota = ");
31         USART_putstring(slovo);
32         USART_putstring("\t napatie = ");
33         USART_putstring(slovoV);
34         USART_putstring(".");
35         USART_putstring(slovoVzv);
36         USART_putstring(" V\n");
37
38         delay_ms(1000); //pozastavenie procesora
39     }
40     return 0;
41 }
```

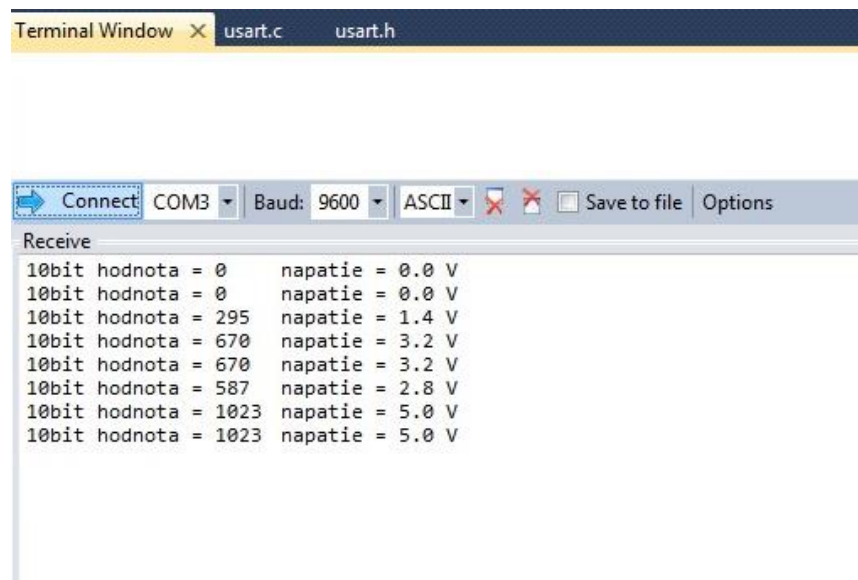
Obr. 15 Kód aplikácie - AD převodník

V prvých riadkoch kódu je vkladanie príslušných hlavičkových súborov knižníc, ktoré sú využívané v aplikáciách. Nasleduje hlavný program. Po inicializácii funkcií, sú definované potrebné premenne.

Pre zobrazenie výsledku aplikácie je potrebné vytvoriť komunikáciu mikrokontroléra s počítačom ako zobrazovacím zariadením cez sériovú linku. Knižnica ASF neponúka pre tento konkrétny mikrokontrolér takúto možnosť, preto bolo nutné ju vytvoriť. Knižnica `usart.c` obsahuje základné funkcie na zobrazenie textového reťazca string na terminál. Jej kód je uvedený v prílohe.

V nekonečnom cykle je najprv získaná hodnota z AD prevodníka, následne prebieha výpočet celej a desatinnej časti digitálneho čísla. Pri 10 bit prevodníku dostaneme maximálnu hodnotu 1023 čo predstavuje maximálne napätie 5V v analógovej hodnote. Výpočet celej časti preto pozostáva z vydelenia aktuálnej digitálnej hodnoty jednou pätinou maximálnej hodnoty prevodníka, zaokrúhlenej smerom nadol. Výpočet desatinnej časti je podobný s ohľadom na pohyb desatinnej čiarky. Výpočet pri takomto zaokrúhľovaní nie je celkom presný a však pre ilustráciu AD prevodníka postačujúci. Nasleduje prevod premennej typu integer na string pre potrebu zobrazenia na sériovú linku a samotný výpis na terminál. Táto časť kódu sa opakuje v sekundovom intervale.

Výstup z programu je terminálové okno (Obr. 16) kde sa nám zobrazujú získané a vypočítané údaje pomocou sériovej linky a knižnice `usart.c`.



```
Terminal Window x usart.c usart.h

Connect COM3 Baud: 9600 ASCII Save to file Options

Receive
10bit hodnota = 0      napatie = 0.0 V
10bit hodnota = 0      napatie = 0.0 V
10bit hodnota = 295    napatie = 1.4 V
10bit hodnota = 670    napatie = 3.2 V
10bit hodnota = 670    napatie = 3.2 V
10bit hodnota = 587    napatie = 2.8 V
10bit hodnota = 1023   napatie = 5.0 V
10bit hodnota = 1023   napatie = 5.0 V
```

Obr. 16 Výstup zo sériovej linky AD prevodníka

8 VÝHODY A NEVÝHODY ATMEL SOFTWARE FRAMEWORK

Programovanie mikropočítačov pomocou Atmel Software Framework v jazyku C je veľmi komfortné a časovo menej náročné. Podpora mikropočítačov firmy Atmel je na vysokej úrovni. ASF ponúka pre jednotlivé mikropočítače knižnice, ktoré pomáhajú pri ich programovaní. Je výnimočný svojou univerzálnosťou a použiteľnosťou pre množstvo mikropočítačov od firmy Atmel.

Snáď jedinou nevýhodou pri používaní ASF mi prišla dokumentácia k jednotlivým knižniciam. Popis niektorých funkcií nie je dostatočný. Za spomenutie snáď stojí aj fakt, že vývojové prostredie Atmel Studio, s ktorým ASF najlepšie spolupracuje a je jeho súčasťou, je náročnejšie na výkonnosť počítača, na ktorom pracujete. Počas nastavovania programu som sa dostal do situácií, kedy bol program zaseknutý a bolo ho nutné reštartovať alebo načítavanie dlho trvalo.

Ak ale porovnať výhody programovanie mikroprocesor pomocou Atmel Software Framework v jazyku C alebo C++ z jeho nevýhodami. Iste by sme si radšej vybrali použiť knižnicu ASF.

ZÁVĚR

Táto bakalárska práca bola zameraná na knižnicu firmy Atmel Software Framework ako podpora mikropočítačov a uľahčenie ich programovania. Začiatkom bolo predstavenie knižnice ASF, pre ktoré bolo niečo krátko popísané o mikropočítačoch samotných. Opísané boli základné súčasti knižnice s príslušnými funkciami. Ich využitie, už záleží na individuálnosti prípadu. V tejto práci boli uvedené dva konkrétne príklady

K praktickej časti bol vybraný konkrétny mikrokontrolér ATmega 328P. Práca s týmto mikrokontrolérom bola zaujímavá, svojimi periférnymi možnosťami aj rôznorodá. LED dióda na vývojovej doske Arduino Uno bola využitá na jednu z praktických ukážok ASF. Demonštrovaná bola práca s pinami a časovým zdržaním, riadením, mikrokontroléra. Druhou z ukážkových programov bola založená na možnosti mikrokontroléra a to AD prevodník. Výsledkom bolo prevod analógovej hodnoty dostupnej z vývojovej dosky na digitálnu hodnotu s použitým rozsahom prevodníku. Opis oboch programov je poňatý podrobne s detailným rozborom kódu.

Po oboznámení sa s knižnicou ASF, predstavením aktuálne využívaných mikropočítačov od firmy Atmel, výbere vhodného mikrokontroléra ako aj vývojového prostredia použitého k práci s knižnicou bolo ku koncu práce zhodnotené iné možnosti práce s knižnicou, využitie iných vývojových prostredí a postupu programovania. Po naštudovaní tejto bakalárskej práce by malo byť programovanie pomocou ASF jednoduchšie. Výber práve tejto knižnice, prácu s programovaním mikropočítača urýchli a uľahčí.

SEZNAM POUŽITÉ LITERATURY

- [1] PINKER, Jiří. 2008. Mikroprocesory a mikropočítače. 1. vyd., 1. dot. Praha: BEN - technická literatura, 159 s. ISBN 978-80-7300-110-0
- [2] Úvod do mikroprocesorové techniky. KEMT FEI TUKE [online]. 2004 [cit. 2015-05-17]. Dostupné z: http://kemt-old.fei.tuke.sk/Predmety/KEMT411_ESM/_web/wwwfiles/str%2001.htm
- [3] VÁŇA, Vladimír. Mikrokontroléry ATMEL AVR: programování v jazyce C : popis a práce ve vývojovém prostředí CodeVisionAVR C. 1. vyd. Praha: BEN - technická literatura, 2003, 215 s. ISBN 80-730-0102-0.
- [4] Mikroprocesory, mikropočítače a ich základné bloky. KEMT FEI TUKE [online]. 2004 [cit. 2015-05-17]. Dostupné z: http://kemt-old.fei.tuke.sk/Predmety/KEMT411_ESM/_web/wwwfiles/str%2002.htm
- [5] Rozdelenie mikroprocesorov. a mikropočítačov. KEMT FEI TUKE [online]. 2004 [cit. 2015-05-17]. Dostupné z: http://kemt-old.fei.tuke.sk/Predmety/KEMT411_ESM/_web/wwwfiles/str%2003.htm
- [6] MANN, Burkhard. C pro mikrokontroléry: ANSI-C, kompilátory C, spojovací programy - linkery, práce s ATMEL AVR a MSC-51, příklady programování v jazyce C, nástroje pro programování, tipy a triky. Vyd. 1. Praha: BEN, 2003, 279 s. ISBN 80-730-0077-6.
- [7] ASF Architecture. Atmel [online]. 2012 [cit. 2015-05-17]. Dostupné z: <http://asf.atmel.com/docs/latest/architecture.html>
- [8] BARR, Michael a Anthony J MASSA. Programming embedded systems. 2nd ed. Sebastopol: O'Reilly, 2006, xxi, 301 s. ISBN 978-0-596-00983-0.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ASF	Atmel Software Framework
MPU	Mikro Processor Unit
MCU	Mikro Controller Unit
CPU	Central Processing Unit
ALU	Arithmetic Logic Unit
FPU	Floating Point Unit
RAM	Random Access Memory
SRAM	Static Random Access Memory
EEPROM	Electrically Erasable Programmable Read-Only Memory
DMA	Direct Memory Access
AD	Analógovo digitálny prevodník
DA	Digitálne analógový prevodník
USI	Universal Serial Interface
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver / Transmitter
USART	Universal Synchronous / Asynchronous Receiver and Transmitter
TWI / I2C	Two Wire Interface
CAN	Controller Area Network
LIN	Local Interconnect Network
AES	Advanced Encryption Standard
DES	Data Encryption Standard

SEZNAM OBRÁZKŮ

Obr. 1 Schéma mikroprocesora [4].....	13
Obr. 2 Schéma mikropočítača [4].....	14
Obr. 3 Harvardská architektúra [2].....	16
Obr. 4 Architektúra Von neumann [2].....	16
Obr. 5 Štruktúra ASF [7]	23
Obr. 6 Vytvorenie nového projektu	30
Obr. 7 Nastavenie nového projektu	31
Obr. 8 Výber dosky.....	31
Obr. 9 Vloženie knihovien ASF	32
Obr. 10 Implementácia knižníc do projektu	32
Obr. 11 Vložené hlavičkové súbory v projekte	33
Obr. 12 Inštalácia externej vývojovej dosky	34
Obr. 13 Nastavenie externého zariadenia	35
Obr. 14 Kód aplikácie - blikanie LED.....	36
Obr. 15 Kód aplikácie - AD prevodník.....	38
Obr. 16 Výstup zo sériovej linky AD prevodníka	39
Obr. 17 Úvodné okno inštalácie	47
Obr. 18 Inštalácia - súhlas s licenčnými podmienkami	48
Obr. 19 Úvodné okno inštalácie Atmel Studia	48
Obr. 20 Súčasti inštalácie Atmel Studia	49
Obr. 21 Dokončenie inštalácie Atmel Studia.....	49
Obr. 22 Dostupná aktualizácia Atmel Studia.....	50
Obr. 23 Zdrojový kód knižnice usart.h.....	52
Obr. 24 Zdrojový kód usart.c.....	52

SEZNAM TABULEK

Tab. 1 API obsahované v ASF pro ATmega328P	24
Tab. 2 Vývojové desky dostupné Generic board support.....	51

SEZNAM PŘÍLOH

PŘÍLOHA P I: DOKUMENTÁCIA K INŠTALÁCIÍ ATMEL STUDIO

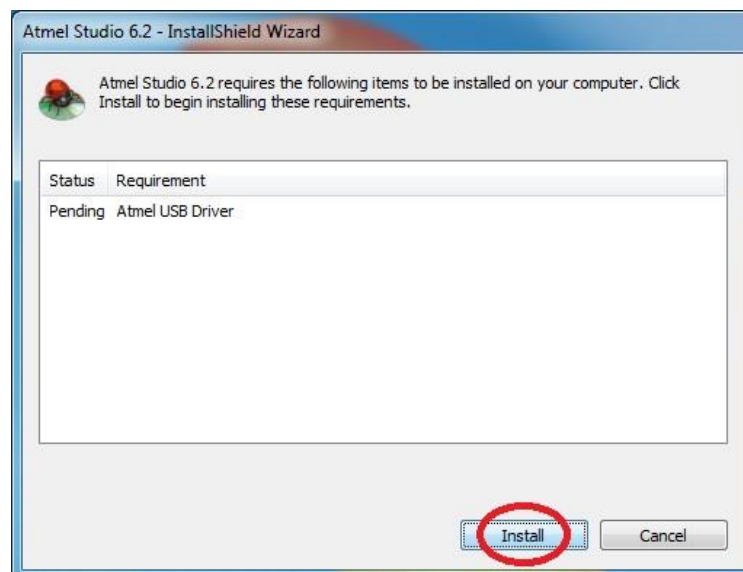
PŘÍLOHA P II: VÝPIS DOSTUPNÝCH VÝVOJOVÝH DOSIEK V GENERIC BOARD SUPPORT

PŘÍLOHA P III: ZDROJOVÉ KÓDY KNIŽNICE USART

PŘÍLOHA P I: DOKUMENTÁCIA K INŠTALÁCIÍ ATMEL STUDIO

Inštalácia Atmel Studio 6

Základný popis postupu inštalácie vývojového prostredia Atmel Studio). Stiahnutie tohto vývojového prostredia je bezplatné. Môžeme tak spraviť na webovej stránke <http://www.atmel.com/tools/atmelstudio.aspx>. Po dokončení sťahovanie otvoríme aplikáciu. Privíta nás úvodné okno (Obr. 16).



Obr. 17 Úvodné okno inštalácie

Inštalačná aplikácia sa nám oznamuje čo je potrebné na inštaláciu. Do počítača sa nám nainštaluje samotný program s potrebnými aplikáciami a špeciálny ovládačom pre USB konektivitu. Inštaláciu potvrdíme označením tlačidlom „Install“. Ďalej sa nám zobrazí okno s licenčnými podmienkami inštalácie a používania ovládača USB driver package. Pre súhlas s nimi vyznačíme príslušné políčko. Pokračujme stlačením tlačidla „Install“ (Obr. 17). Po odsúhlasení podmienok a príkaze inštalácie USB driver package nasleduje jeho samotná inštalácia.



Obr. 18 Inštalácia - súhlas s licenčnými podmienkami

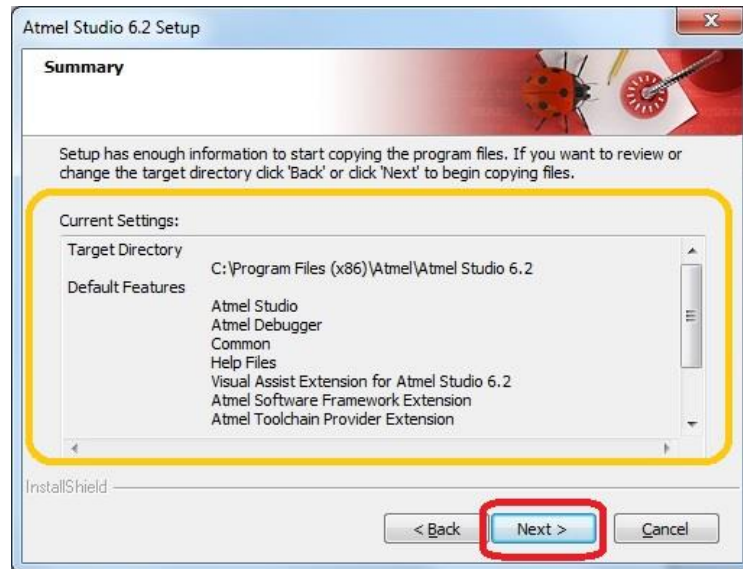
Po nainštalovaní sa nám spustí inštalacia aplikácia priamo Atmel Studia. Vidíme úvodné okno. Potvrdíme stlačením „Next“ (Obr. 18).



Obr. 19 Úvodné okno inštalácie Atmel Studia

Pokračujeme oknom, v ktorom si môžeme vybrať umiestnenie programu. Odporúčaný je priamo adresár vybraný inštalacným programom, pri prípadnej zmene isto pre lepšiu funkcionality odporúčam voliť disk, na ktorom je umiestnený operačný systém. Po výbere cieľového adresára stlačením tlačidla „Next“ postúpime ďalej. Nasledujúcim okne si môžeme

všimnúť vybranú cestu k adresáru inštalácie a súčasti inštalácie Atmel Studia. Pokračujeme odsúhlasením, tlačidlom „Next“ (Obr. 19).



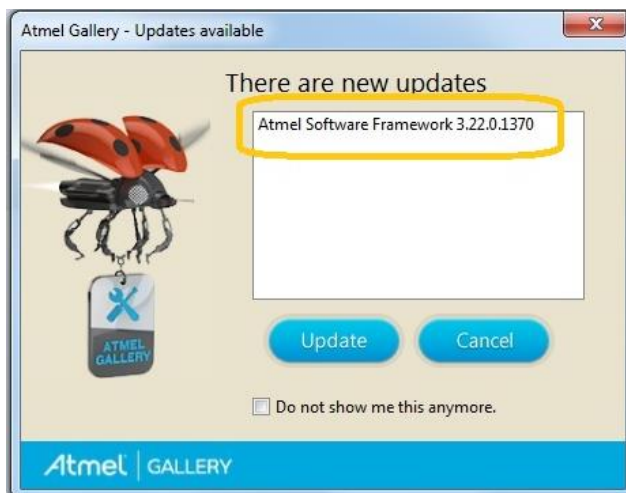
Obr. 20 Súčasti inštalácie Atmel Studia

Začne samotná inštalácia programu. Po jej dokončení by sa nám malo zobrazit' okno s jej úspešným dokončením a dopytom na prepojenie súborových koncoviek *.c, *.s *.asm, *.h s programom Atmel Studio. Túto voľbu odporúčam iba v prípade, že tieto typy súborov bude využívať tento jediný program. Inštaláciu dokončíme tlačidlo „Finish“ (Obr. 20). Úspešne sme nainštalovali vývojové program Atmel Studio. Ostatná práca sa bude odohrávať konkrétne v ňom.



Obr. 21 Dokončenie inštalácie Atmel Studia

Otvoríme si Atmel Studio, privíta nás uvítacie okno. Ako nižšie na obrázku (Obr. 21) vidíme mám ponuku aktualizácii Atmel software Frameworku. Aktualizáciu vrelo odporúča, pri našej práci nám zabezpečí bezproblémovjšie fungovanie Frameworku.



Obr. 22 Dostupná aktualizácia Atmel Studia

Tieto aktualizácie sú kontrolované samotným programom automaticky pri jeho zapnutí. Môžeme si byť tak istý, že ako aj program Atmel Studio tak aj pomocné aplikácie sú udržiavané aktualizované. Po úspešnom aktualizovaní je Atmel Studio pripravené.

PŘÍLOHA P II: VÝPIS DOSTUPNÝCH VÝVOJOVÝH DOSIEK V GENERIC BOARD SUPPORT

Názov vývojovej dosky	Názov vývojovej dosky
Arduino Due/X board.	SAM D21 Xplained Pro board.
AT32UC3 MC300 board.	SAM R21 Xplained Pro board.
AT32UC3A EVK1100 board.	SAM3N-EK board.
AT32UC3A EVK1105 board.	SAM3S-EK board.
AT32UC3A RZ600 MCU board.	SAM3S-EK2 board.
AT32UC3A3 EVK1104 board.	SAM3U-EK board.
AT32UC3B EVK1101 board.	SAM3X-EK board.
AT32UC3B EXT1102 board.	SAM4C-EK board.
AT32UC3C UC3C-EK board.	SAM4CMP demo board.
AT32UC3L-EK board.	SAM4CMS demo board.
AT86RF212 RF board in RZ600.	SAM4CP16BMB board.
AT86RF230B RF board in RZ600.	SAM4E Xplained Pro board.
AT86RF231 RF board in RZ600.	SAM4E-EK board.
ATmega1284P Xplained board.	SAM4L Xplained Pro board.
ATmega256RFR2 Xplained Pro board.	SAM4L8 Xplained Pro board.
ATmega256RFR2 zigbit.	SAM4L-EK board.
ATPL230AMB board.	SAM4N Xplained Pro board.
ATUC3A3 UC3-A3 Xplained board.	SAM4S Xplained board.
ATUC3L0 UC3-L0 Xplained board.	SAM4S Xplained Pro board.
ATUCL3 STK600 board.	SAM4S-EK board.
ATUCL4 STK600 board.	SAM4S-EK2 board.
ATxmega128A1 STK600 board.	SAM4S-WPIR-RD board.
ATxmega128A1 Xplain board.	SAMG53 Xplained Pro board.
ATxmega128A1 Xplain-A1 board.	STK600 MEGA board.
ATxmega128B1 Xplained board.	STK600 MEGA RF EVK board.
ATxmega256A3 STK600 board.	STK600 RCUC3A0 board.
ATxmega256A3BU Xplained board.	STK600 RCUC3B0 board.
ATxmega256A3U with AT86RF212B Zigbit.	STK600 RCUC3C board.
ATxmega256A3U with AT86RF233 Zigbit.	STK600 RCUC3D board.
ATxmega32E5 Xplained board.	STK600 RCUC3L0 board.
ATxmega384C3 Xplained board.	STK600 with RC032X routing card board.
QT600 ATXMEGA128A1 MCU board.	STK600 with RC044X routing card board.
QT600 UC3L0 MCU board.	XMEGA REB Controller Base board.
RFR2 & RFA1 RCB.	XMEGAB1 STK600 RC100X LCDX board.
RFR2 RCB Xplained Pro board.	XMEGAB3 STK600 RC064X LCDX board.
SAM D20 Xplained Pro board.	

Tab. 2 Vývojové dosky dostupné Generic board support

PŘÍLOHA P III: ZDROJOVÉ KÓDY KNIŽNICE USART

```
9 #ifndef USART_H_
10 #define USART_H_
11
12 #include<ioport.h>
13
14 #define F_CPU 16000000UL
15 #define BAUDRATE 9600
16 #define BAUD_PRESCALLER (((F_CPU / (BAUDRATE * 16UL))) - 1)
17
18 void USART_init(void);
19 unsigned char USART_receive(void);
20 void USART_send(unsigned char data);
21 void USART_sendI(unsigned int data);
22 void USART_putstring(char* StringPtr);
23
24
25 #endif /* USART_H_ */
```

Obr. 23 Zdrojový kód knihovny usart.h

```
7 #include "usart.h"
8 void USART_init(void){
9
10     UCSRA &= ~_BV(U2X0);
11     UBRROH = (uint8_t) (BAUD_PRESCALLER>>8);
12     UBRROL = (uint8_t) (BAUD_PRESCALLER);
13     UCSRB = (1<<RXEN0) | (1<<TXEN0);
14     UCSRC = (3<<UCSZ00);
15 }
16
17 unsigned char USART_receive(void){
18
19     while(!(UCSRA & (1<<RXC0)));
20     return UDR0;
21 }
22
23 void USART_send(unsigned char data){
24
25     while(!(UCSRA & (1<<UDRE0)));
26     UDR0 = data;
27 }
28
29 void USART_sendI(unsigned int data){
30
31     while(!(UCSRA & (1<<UDRE0)));
32     UDR0 = data;
33 }
34 }
35
36 void USART_putstring(char* StringPtr){
37
38     while(*StringPtr != 0x00){
39         USART_send(*StringPtr);
40         StringPtr++;}
41 }
```

Obr. 24 Zdrojový kód usart.c