

*Příjemci podpory:*

*Poskytovatel:*

Bezpečné dopravní systémy nové generace  
(VB01000048)

## **VB01000048-V2**

### **Experimentální funkční vzorek kamerového systému nové generace (funkční vzorek)**

Typ výsledku dle UV č. 837/2017	Evidenční číslo (příjemce)	Rok vzniku
Gfunk - funkční vzorek	VB01000048-V2	2022
ISBN-ISSN	Webový odkaz na výsledek	Kde a kdy publikováno
N/A	N/A	N/A

#### **Anotace k výsledku:**

Výsledek je funkčním vzorkem experimentálního kamerového systému pro monitorování dopravy. Jedná se o experimentální úpravu existujícího kamerového systému s vestavěným počítačovým systémem s procesorem Intel Atom. V kamerovém systému jsou nově doplněny funkce pro detekci dopravních objektů, jak bylo i plánováno, konkrétně o detekci registračních značek prostřednictvím neuronové sítě, jakož i o detekci vozidel, konkrétně vozidel s "majákem". Tato detekce slouží pro identifikaci vozidel, která mají oprávnění jízdy ve vyhrazených jízdních pruzích. Výsledek je experimentálně ověřen a bude sloužit zejména pro další fázi projektu i jako demonstrátor.

#### **Řešitelský tým:**

Eliška Vlčková (od 1.4. 2022), Ivana Sekaninová (do 31.3.2022) - manažer projektu, Lukáš Maršík – hlavní řešitel, Vítězslav Beran – další řešitel, Miroslav Juhas, Jiří Košák, Aleš Křupka, Marek Langr, Radek Němec, Jiří Řehák, Pavel Valenta, Daniel Bambušek, David Bařina, Adam Herout, Martin Kreslík, Pavel Smrž, Jakub Špaňhel, Pavel Zemčík

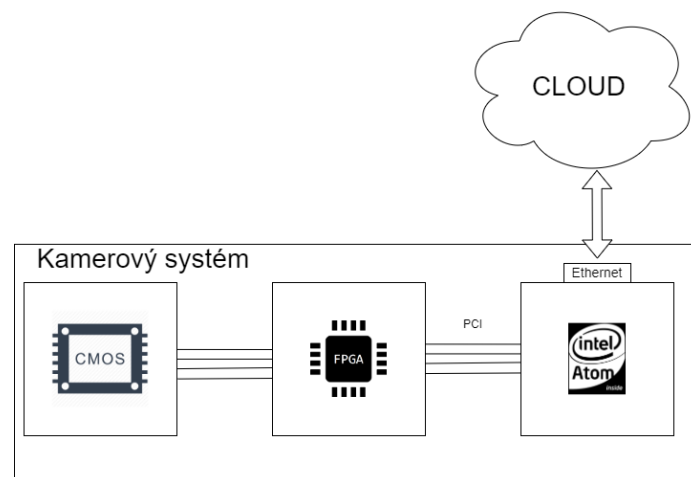
# Obsah

Obsah .....	2
1 Úvod .....	3
2 Popis funkčního vzorku .....	4
2.1 Příjem obrazu v FPGA .....	4
2.2 Zpracování a vyhodnocení snímků .....	4
3 Integrace.....	7
4 Vyhodnocení.....	8

## 1 Úvod

Tento funkční vzorek projektu MV SECTECH – Bezpečné dopravní systémy nové generace (VB01000048) demonstruje na „inteligentní kameře“ možnost detekce objektů. Technicky se jedná o kameru postavenou na již dříve zpracované kamerové platformě CAMEA s Intel ATOM, která je složena z:

- FPGA modul firmy Xilinx – realizuje interface pro příjem dat a řízení kamerového senzoru
- Intel ATOM SoC – 4-jádrový procesor vybaven integrovaným grafickým akcelerátorem
- kamerový senzor – senzor (Sony IMX – podporované jsou různé verze, např. IMX265) je připojen k FPGA pomocí LVDS



Obrázek 1: Schéma kamerového systému

Platforma je založená na procesoru řady Intel Atom x7-E3950. Výhodou platformy Intel Atom je kompatibilita s architekturou x86-64 (též AMD64), která umožňuje běh již existujícího aplikačního vybavení pro desktopové a serverové PC, a to jak SW, tak i používané inferenční frameworky Intel OpenVINO, TensorFlow nebo ONNX-Runtime. Platforma byla podrobena testování výkonu, kde se ukázala možnost využití integrovaného GPU pro obecné výpočty a akceleraci inferencie neuronových sítí. Akcelerace umožnila real-time vyhodnocování obrazu z jedné kamery na úloze detekce SPZ.



Obrázek 2: Kamerový systém (vlevo bez krytu, vpravo s krytem)

## 2 Popis funkčního vzorku

Z funkčního pohledu se jedná o kamerovou platformu, do které jsou v rámci projektu integrovány nové funkce a která je schopna realizovat pořízení a následně zpracování obrazu. Data jsou přenesena ze senzoru do FPGA logiky, která realizuje příjem dat a rychlý výpočet statistických informací z obrazu. Data jsou následně přenesena z FPGA do RAM Intel Atom prostřednictvím PCI sběrnice. Data jsou vyčítána z PCI samostatným procesem, po kterém následuje zpracování obrazových dat následným způsobem:

1. Předzpracování obrazu:  
Převod barevného modelu (výstup ze senzoru je, v případě barevného senzoru, ve formátu Bayerové masky), tj. Bayer → BGR. V případě černo-bílého senzoru je potřebné vykonat rozšíření na tři kanály.
2. Nalezení SPZ (státní poznávací značky) pomocí čtecího modulů
3. V případě nalezení SPZ, spustí se modul pro detekci majáků
4. Informace o SPZ a majáku je předána do cloudu prostřednictvím ethernetu



Obrázek 3: Diagram toku dat

### 2.1 Příjem obrazu v FPGA

V prvním kroku probíhá dekodování obrazu z CMOS obrazu (funkční blok Serdes2Axis). Je napojen na hardwarové SERDES pro demultiplex vysokorychlostních serio-parallelních linek, realizuje hledání synchronizačních patternů v signálu, dekodování jednotlivých řádků obrazu a také reportuje počty chyb vzniklých na linkách. Následuje blok pro základní předzpracování obrazu – FrameProcessing. Zde probíhá vytvoření zájmové části obrazu (region of interest), aplikace white-balance, výpočet statistických parametrů obrazu (suma obrazu pro jednotlivé barevné kanály). Následně je možné aplikovat “frame binnig” (sčítání řádků a/nebo sloupců). Nakonec je obraz přes rozhraní PCI přenesen do RAM procesoru Intel Atom.

### 2.2 Zpracování a vyhodnocení snímků

Pro účely vzniku funkčního vzorku byl vytvořen SW systém, který umožňuje zpracování a vyhodnocení obrazu přímo v kameře. Tento řídicí software se skládá z dvou modulů:

- Předzpracování obrazu
- Modul detektoru SPZ a majáků

Moduly jsou napsány v jazyce C++ s využitím následujících knihoven/frameworků:

- OpenCV<sup>1</sup> (Open Source Computer Vision Library) - je knihovna pro práci s počítačovým viděním a strojovým učení. Poskytuje širokou škálu nástrojů pro analýzu obrazu, jako jsou detekce tváří, sledování objektů, rozpoznávání textu a další.
- OpenCL<sup>2</sup> (Open Computing Language) - je otevřený standard pro paralelní programování, který umožňuje využít výpočetní výkon různých typů hardwaru, včetně procesorů, grafických karet

<sup>1</sup> <https://opencv.org/>

<sup>2</sup> <https://www.khronos.org/opencl/>

a dalších zařízení. Pro Intel iGPU (integrovaná grafická jednotka) OpenCL poskytuje rozhraní pro využití výkonu iGPU pro paralelní výpočty, což může výrazně zvýšit výkon aplikací, které vyžadují velké množství paralelních výpočtů, jako jsou například aplikace pro počítačové vidění nebo strojové učení.

- OpenVINO<sup>3</sup> (Open Visual Inference and Neural network Optimization) je sada nástrojů od společnosti Intel pro optimalizaci a rychlé provozování aplikací strojového učení na různých platformách, včetně procesorů, grafických karet a výkonných zařízení pro strojové učení. OpenVINO poskytuje sadu nástrojů pro optimalizaci modelů strojového učení, které umožňují snížit nároky na výpočetní výkon a snížit velikost modelu, což umožňuje provozovat modely na zařízeních s omezenými zdroji. OpenVINO také poskytuje sadu knihoven a API pro využití výkonu různých typů hardwaru pro provoz aplikací strojového učení, včetně podpory pro OpenCL a OpenVX.

## Předzpracování obrazu

Algoritmy zpracování obrazu jsou implementovány, jak pro CPU (typicky implementace přes knihovnu OpenCV, popřípadě jako rozšíření OpenCV), tak i pro integrovanou GPU. Společně algoritmy a vyhodnocení CNN vytváří řetězec metod. Pro implementaci algoritmů pro integrovanou Intel GPU je použit framework OpenCL.

Mezi implementované algoritmy zpracování obrazu v řetězci metod patří:

- Transformace z Bayerové reprezentace na RGB - je metoda, která se používá pro kódování barevných obrazů v digitálních obrazových snímačích. Jde o filtr, který se nachází přímo na snímači (CMOS nebo CCD), a který rozděluje světlo na tři základní barvy: zelenou, červenou, modrou. Tyto barvy se poté zpracovávají samostatně a sloučí se do jednoho barevného obrazu. Bayerova maska je velmi populární, protože umožňuje vytvořit vysoké rozlišení barevných obrazů s relativně malým počtem pixelů. Aktuálně je implementována metoda „4-way linear interpolation“
- Transformace z RGB na šedo-tónový
- Změna velikosti snímku

Mezi čistě CPU algoritmy patří (pouze na CPU vzhledem k jejich náročnosti a doby vykonávání – neušetřilo by významnější čas):

- Afinní transformace obrazu/výřezu
- Geometrické transformace
- Výpočet non-maxima suppression – je metoda pro snížení počtu falešných pozitivních výsledků při detekci objektů. Tato metoda se často používá v kombinaci s konvolučními neurony (CNN) nebo jinými algoritmy pro detekci objektů. Princip NMS je v tom, že se prochází detekce objektu a pro každý objekt se zjistí jestli se nachází v blízkosti jiného objektu s vyšší hodnotou detekce. Pokud ano tak se nižší detekce zahazuje.

Vzhledem k omezeným výpočetním zdrojům na platformě Intel ATOM byl velký důraz kladen na optimální integraci a ověření jednotlivých algoritmů na cílové platformě.

---

<sup>3</sup> <https://docs.openvino.ai/latest/home.html>

## Modul detektoru SPZ a detektoru majáků

Detektor (SPZ i majáky) je novou funkcí platformy a je založen na konvolučních neuronových sítích, které byly natrénovány ve frameworku TensorFlow a optimalizovány pomocí toolkitu Intel OpenVINO. Aplikovaný postup umožňuje pro inferenci využít i integrovanou GPU.

Modul detekuje běžně používané majáky v České republice. Příklady jsou zobrazeny na následujících snímcích:



Obrázek 4: Typické typy majáků

Vstupem do modulu je aktuální snímek spolu s metadaty (časové razítko, expozice, statistické údaje o obrazu). V prvním kroku je vyhodnocen detektor SPZ. Obraz je nakopírován do paměti GPU a předán na další zpracování. Nejdříve je obraz naškálován na vhodnou vstupní velikost sítě (zahrnuje i zarovnání obrazu na násobek 32 pixelů). Následně je ukazatel na snímek předán do inferenčního modulu OpenVINO, bez kopírování je předán přímo OpenCL buffer. Po vyhodnocení sítě jsou výstupní tensor<sup>4</sup> předány do vyhodnocovacího modulu. Zde dojde k dekodování tensorů a aplikaci non-maxima suppression.

Intel OpenVINO umožňuje snadno optimalizovat a provozovat aplikace pro inferenci na různých platformách. Tuto platformu nabízí Intel a umožňuje využít různé technologie pro zrychlení inferenčních úloh na různých zařízeních. OpenVINO také poskytuje sadu nástrojů pro modelování a analýzu dat, což umožňuje snadněji pracovat s různými typy dat a modely.

Celý řetězec metod je vyhodnocen v reálném čase – tj. každý snímek z kamery je zpracován a vyhodnocen s nízkou latencí (čas mezi získáním snímku a jeho vyhodnocením je v řádech desítek milisekund – při 20 snímcích za sekundu). Důsledkem je, že nedochází k odesílání všech obrazových dat do cloudu, a tedy inteligentní kamera funguje jako „filtr“ a odesílá jenom zájmová data. Ovšem je důležité zmínit možnost spuštění vyhodnocení i na cloudu (jak bylo zmíněno, je zde kompatibilita s architekturou x86-64).

Zpracování obrazu a, inference a vyhodnocení je napsáno v jazyce C++ kompilovatelné pomocí G++ 7.5. Vyšší logika (příprava dat pro odeslání do cloud, odeslání do cloud, řízení běhu aplikace) je napsáno v jazyce C#. Algoritmy zpracování obrazu jsou akcelerovány na GPU pomocí OpenCL, pro dílčí úlohy je využita knihovna OpenCV a pro inferenci, již výše zmíněné, OpenVINO. Důraz byl kladen na nízkou latenci celého systému.

---

<sup>4</sup> Tensor je matematický objekt, který se používá pro popis dat v několika dimenzích. Jde o n-tice čísel nebo symbolů, které jsou uspořádány do tabulky (matice) nebo do prostoru vyššího počtu dimenzí. Tensory se často používají v matematice, fyzice a strojovém učení. V posledním případě se používají k reprezentaci a zpracování dat v neuronových sítích.

Ukázky detekce:



Obrázek 4: Ukázky detekce majáků

### 3 Integrace

Tyto moduly byly následně zintegrovány do existujícího řídicího systému, což umožnilo integraci do stávajících systémů CAMEA (využití existující infrastruktury, SW prostředků, ovládání kamery a podobně). Vzhledem k faktu, že moduly jsou implementovány v C/C++ bylo potřebné vykonat integraci do existujících C# modulů. Integrace kódu C do C# lze provést pomocí třídy "DllImport" v C#. Tato třída umožňuje načtení a volání funkcí z externí knihovny (dynamická knihovna .so na Unix systémech), která je naprogramovaná v jazyce C. Před použitím třídy "DllImport" je nutné specifikovat cestu k externí knihovně a deklarovat volané funkce. Poté lze tyto funkce volat stejně jako volání jakékoli jiné funkce v C#, například:

**C:**

```
API Detections *Detect(Handle handle, unsigned moduleID, Image *img, Rectangle *areaOfInterest,
    ErrorCode *status);
```

**C#:**

```
[DllImport("beacon_detector.so", CallingConvention = CallingConvention.Cdecl)]
private static extern Detections* Detect(Handle handle, uint moduleID, Image* img, Rectangle*
    areaOfInterest, ErrorCode* status);
```

## 4 Vyhodnocení

Pro referenci byl test proveden i na platformě NVIDIA Jetson NANO, který disponuje výkonnější GPU, Vstupní dataset obsahuje SPZ aut z běžného provozu pořízených portálovou kamerou. Pořízené snímky mají rozlišení 1280x1024 a v tabulce níže uvádíme jak výkon pro jedno-vláknové zpracování tak i maximální výkon dosažený paralelním zpracováním (rozpracováno více snímků současně), ale má velmi omezený výkon CPU. Následující výsledky jsou pro úlohu detekce SPZ:

Platforma	Framework	Výpočetní jednotka	Paralelismus	FPS
<b>Intel Atom e3950</b>	OpenVino	CPU	1	4
<b>Intel Atom e3950</b>	OpenVino	CPU	4	9
<b>Intel Atom e3950</b>	OpenVino	Integrovaná GPU	1	17
<b>Intel Atom e3950</b>	OpenVino	Integrovaná GPU	4	31
<b>Jetson NANO</b>	TensorRT	GPU	1	18
<b>Jetson NANO</b>	TensorRT	GPU	2	31