Korisnička aplikacija uređaja za snimanje i reprodukciju video sadržaja za verifikaciju ADAS algoritama

Umiljanović, Luka

Master's thesis / Diplomski rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:324838

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-26**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni studij

KORISNIČKA APLIKACIJA UREĐAJA ZA SNIMANJE I REPRODUKCIJU VIDEO SADRŽAJA ZA VERIFIKACIJU ADAS ALGORITAMA

Diplomski rad

Luka Umiljanović

Osijek, 2018.

SADRŽAJ

1. UVOD 1

2. KORIŠTENE TEHNOLOGIJE I ALATI PRI RAZVOJU KORISNIČKE APLIKACIJE ZA *AMV GRABBER* UREĐAJ 4

2.1. *AMV Grabber* uređaj 4

2.2. Kamere korištene pri razvoju korisničke aplikacije 6

2.3. *Qt* programski okvir 7

2.3.1 *Qt Creator* 7

2.3.2. Sustav meta-objekata 9

2.3.3. Signali i utori 10

2.3.4. Qmake 11

3. REALIZACIJA KORISNIČKE APLIKACIJE ZA *AMV GRABBER* UREĐAJ 12

3.1. Upravljanje snimanjem *AMV Grabber* uređajem 13

3.1.1. Snimanje kamerama 13

3.1.2. Snimanje ispitnog uzorka 17

3.2. Pretvorbe snimljenih podataka 18

3.3. Prikaz snimljenog video sadržaja u korisničkoj aplikaciji 20

3.4. Reprodukcija snimljenih podataka 21

4. TESTIRANJE FUNKCIONALNOSTI PROGRAMSKOG RJEŠENJA 23

4.1. Moguća poboljšanja programskog rješenja 26

5. ZAKLJUČAK 27

LITERATURA 28

SAŽETAK 29

ABSTRACT 30

ŽIVOTOPIS 31

1. UVOD

Većina prometnih nesreća dogodi se zbog ljudske pogreške. Napredni sustavi za pomoć u vožnji (engl. *advanced driver-assistance systems - ADAS*) i tehnologije autonomne vožnje razvijaju se s ciljem veće sigurnosti prometa. Pružaju vozaču nužne informacije tijekom vožnje, automatiziraju teže i ponavljajuće zadatke, upozoravaju i reagiraju na kritične situacije, s ciljem smanjenja broja ljudskih pogrešaka. Potencijal im je izrazito velik jer se mogu primjenjivati na svim tipovima vozila, kao što su osobni automobili, kamioni, motori, vozila javnog prijevoza…

ADAS sustavi imaju poseban utjecaj na razvoj autonomnih vozila. Autonomna vozila su ona koja ne zahtijevaju u potpunosti ručno upravljanje vozača. Postoje u različitim oblicima, od potpuno autonomnih, do vozila kojima je potrebno upravljati dio vremena. Općenito, računalni sustavi autonomnih vozila preuzimaju dio odgovornosti i donose dio odluka tijekom vožnje. Osim sigurnosti, argumenti za korištenje autonomnih vozila su smanjenje prometnih gužvi, tj. povećanje protočnosti prometa, smanjenje potrošnje energije i emisije ispušnih plinova, povećanje efikasnosti vožnje i povećanje produktivnosti, jer vozač ne mora biti toliko angažiran u procesu vožnje.

Postoji mnogo različitih ADAS tehnologija, gdje su neke prisutne u vozilima dugo vremena i već dokazano poboljšavaju proces vožnje. Primjer toga je globalni položajni sustav (engl. *global positioning system - GPS*). Razvojem tehnologija stalno se javljaju nova rješenja koja su još u procesima testiranja i općeg prihvaćanja. Neka od tih rješenja naći će se u vozilima budućnosti, dok će neka biti zamijenjena boljim, suvremenijim rješenjima, kako to obično biva dolaskom novih tehnologija. Neke od primjena tih tehnologija su prilagodljivo održavanje stalne brzine, prepoznavanje prometnih znakova i objekata na cesti, nadzor mrtvog kuta, upozorenje i napredno kočenje u slučaju opasnosti, održavanje vozila unutar traka na cesti, praćenje svijesti i uspavanosti vozača, noćni vid, samostalno parkiranje i dr. Sve tehnologije služe za ostvarivanje razina autonomije donesenih od strane društva automobilskih inženjera (engl. *Society of Automotive Engineers - SAE*). Postoji šest razina označenih brojevima od 0 do 5. Danas se te razine primjenjuju u cijelom svijetu i postale su svojevrstan referentni faktor za kategorizaciju razine autonomije pojedinih vozila. Te razine pružaju zamišljen i očekivan redoslijed razvoja tehnologija, i zapravo predstavljaju skup mogućnosti i značajki, no nije nužno da se one razvijaju prema tim razinama. Više informacija moguće je pronaći na [1].

ADAS sustavi prikupljaju i obrađuju podatke u realnom vremenu pa zahtijevaju računala visokih performansi i visoke preciznosti. Oslanjaju se na ulazne podatke dobivene s više perifernih izvora

kao što su kamere, uređaji za svjetlosno zamjećivanje i klasifikaciju (engl. *light detection and ranging*), te radarskih, ultrazvučnih i infracrvenih senzora. Ti uređaji moraju biti pouzdani i otporni na vanjske uvjete kao što su vlaga, ekstremna hladnoća ili vrućina, prljavština, vibracije i ne smiju mijenjati specifikacije sustava, odnosno ne smiju dovesti do pada kvalitete cijelog sustava.

Razvoj tehnologija i smanjenje cijena omogućuju značajan pomak prema ugradnji sustava elektroničke sigurnosti i tehnologija autonomne vožnje u sva vozila. To utječe na povećanje broja zakonskih regulativa koje propisuju ADAS sustave kao obavezne dijelove svih vozila i čini razvoj ADAS sustava jednim od najbrže rastućih dijelova automobilske industrije.

Razvoj ADAS algoritama zahtijeva stalno i temeljito testiranje i davanje potvrda o validnim i nedvosmislenim odlukama algoritama. Ovo je vrlo složen proces, s obzirom da svaki put kada se napravi određena promjena algoritma, tu promjenu treba ispitati izravno na poligonu ili u realnim uvjetima u prometu. To testiranje predstavlja problem jer je s jedne strane potrebno kreiranje i prilagodba poligona, kako bi testovi mogli simulirati različite situacije, što predstavlja značajan trošak. S druge strane, nije moguće testirati algoritme u realnim uvjetima jer takvo testiranje predstavlja rizik za sve sudionike u prometu.

Radi bržeg testiranja i verifikacije ispravnog rada algoritama za pomoć vozaču u vožnji, razvijaju se sustavi koji simuliraju realno okruženje vozila. Ti sustavi omogućuju testiranje rješenja za računalnu percepciju okoline vozila u zatvorenim prostorima, prije testiranja u stvarnim vozilima. Jedan od takvih sustava je *AMV Grabber*, više-kanalni video uređaj za istovremeno snimanje i reprodukciju video signala s devet kamera koje su prostorno raspoređene na vozilu. Podržava prijenos podataka (video signala) velikim brzinama putem *PCIe* (engl. *peripheral component interconnect express*) sučelja kao i upravljanje samim uređajem. Video sadržaji služe za verifikaciju ADAS algoritama u kontroliranim uvjetima.

U okviru ovog diplomskog rada razvijena je korisnička aplikacija za *AMV Grabber* uređaj. Aplikacija kroz korisničko sučelje omogućuje konfiguriranje spojenih kamera (prednja, zadnja, bočna, za praćenje vozača i slično), odabir kamera za snimanje/reprodukciju, izbor putanje odredišne/izvorišne datoteke, pokretanje i zaustavljanje snimanja odnosno reprodukcije, kao i pretvorbu i prikaz snimljenog sadržaja. Za realizaciju funkcionalnosti korištene su komponente koje omogućavaju komunikaciju s *AMV Grabber* uređajem. Aplikacija se izvršava na osobnom računalu unutar Microsoft Windows radnog okruženja.

U drugom poglavlju opisane su korištene tehnologije i alati. Opisan je tip uređaja koji se koriste za hvatanje analognog ili digitalnog video signala s kamera i njihov prijenos na računalo, te za slanje podataka s računala na izlazne priključke na koje je spojena neka druga vrsta uređaja za obradu podataka. Nakon toga su opisani *AMV Grabber* uređaj i modeli kamera korišteni pri razvoju i testiranju cijelog sustava, te je opisan programski okvir *Qt* i njegovi najznačajniji dijelovi pomoću kojih je zadatak realiziran. U trećem poglavlju je opisan način realizacije zadatka. Opisana su dva načina snimanja *AMV Grabber* uređajem, tj. način snimanja kamerama i način snimanja ispitnog uzorka. Nakon toga je opisan način pretvorbe snimljenih podataka u zapise koje je moguće prikazati na računalu, te je opisan način prikaza konačnog video sadržaja u korisničkoj aplikaciji. Također je opisan način reprodukcije snimljenih podataka, tj. slanje podataka s računala na *AMV Grabber* uređaj. U četvrtom poglavlju je opisano testiranje funkcionalnosti aplikacije s *AMV Grabber* uređajem te su navedene mogućnosti poboljšanja cijelog sustava. Peto poglavlje donosi zaključke rada.

2. KORIŠTENE TEHNOLOGIJE I ALATI PRI RAZVOJU KORISNIČKE APLIKACIJE ZA *AMV GRABBER* UREĐAJ

U ovom poglavlju je opisan uređaj za snimanje i reprodukciju video sadržaja za verifikaciju ADAS algoritma, *AMV Grabber*. Opisani su modeli kamera korišteni za snimanje video sadržaja, te je opisan *Qt* programski okvir korišten za razvoj korisničke aplikacije za *AMV Grabber* uređaj.

2.1. *AMV Grabber* uređaj

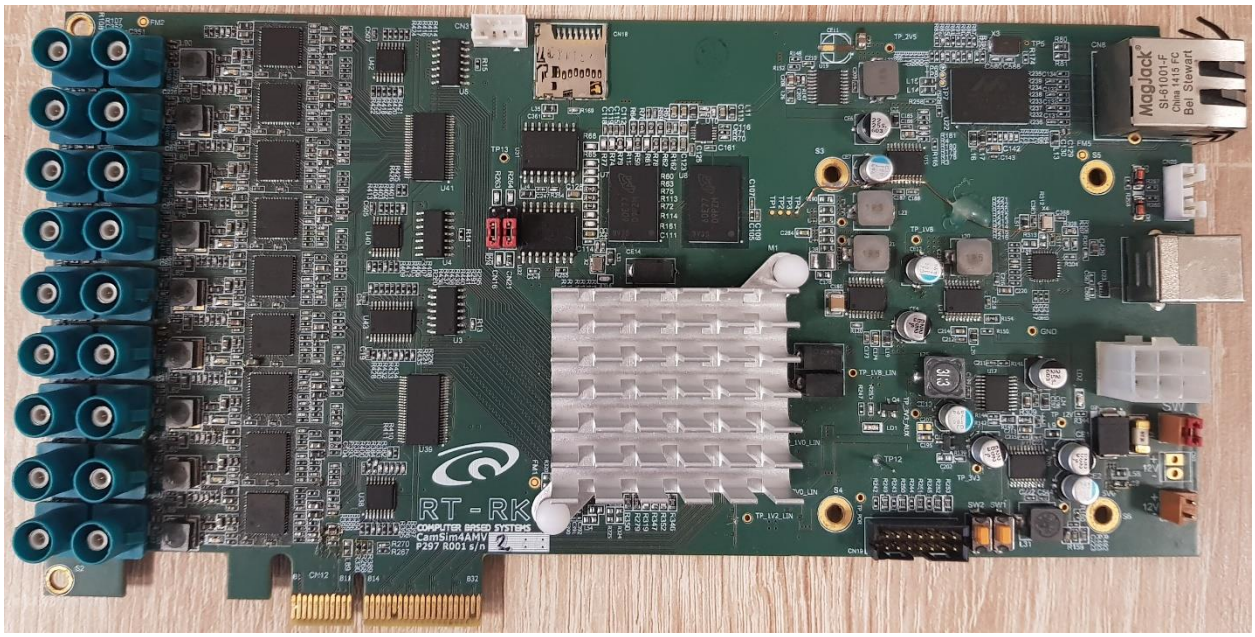
Postoje mnogi elektronički uređaji koji obavljaju funkcije hvatanja analognog ili digitalnog video signala s kamera i njihovog prijenosa na računalo putem različitih ulaznih sučelja kao što su *USB*, *FireWire*, *Gigabit Ethernet* i *PCIe*. Takva vrsta uređaja se naziva hvatač okvira (engl. *frame grabber*) [2]. Spajanjem s matičnom pločom računala, hvatači okvira pružaju fizički priključak za povezivanje kamera korištenih u području računalnog vida. Podržavaju trenutni prikaz, pohranu, obradu i prijenos podataka s više ulaznih priključaka s kojih hvataju video signale. Također podržavaju slanje podataka u drugom smjeru, s računala na izlazne priključke ili s ulaznih priključaka izravno na izlazne na koje je spojena neka druga vrsta uređaja za obradu podataka.

Osim prijenosa video podataka, hvatači okvira obavljaju razne dodatne zadatke važne za sustave računalnog vida. Obavljaju ponovno sastavljanje serije video okvira iz paketa informacija kreiranih za prijenos preko serijskih sučelja, pohranu video okvira u međuspremnike sve dok prijemna procesorska jedinica nije spremna obraditi te okvire, te pružanje kontrole kamere u stvarnom vremenu [3]. Postoje kamere koje proizvode više od 1 GB podataka u sekundi. Bez namjenskih uređaja kao što su hvatači okvira, hvatanje i formatiranje podataka te odgovaranje na vanjske signale bili bi zadaci računala, što bi ostavilo vrlo malo kapaciteta procesora računala za obradu slike i ostale zadatke. Najvažnija mogućnost hvatača okvira je prijenos velike količine podataka visokom brzinom i omogućavanje rada u stvarnom vremenu. Slikovni senzori (engl. *image sensor*s), pružaju okvire visokih rezolucija koje je potrebno obraditi i prenijeti uz visoku brzinu izmjene okvira (eng. *frames per second* - *fps*). Povećanje kvalitete kamera utječe na povećanje performansi hvatača okvira, tj. zahtijeva povećanje brzine i smanjenje vremena potrebnog za zapis okvira u međuspremnike, te čitanje i slanje okvira iz međuspremnika. Za postizanje najvećih brzina koristi se *PCIe* sučelje, što pak ograničava fleksibilnost korištenja jer hvatač okvira mora biti izravno priključen na odgovarajuće sučelje na matičnoj ploči. Korištenje *USB*, *FireWire* ili *Gigabit Ethernet* sučelja ne garantira uvijek potrebnu propusnost (engl. *bandwidth*) i pouzdanost kod prijenosa velikih količina podataka. Korištenje *PCIe* sučelja pruža

pouzdaniji prijenos kod kojeg je moguće uvijek odrediti kada i gdje se signal prenosi, koliko prijenos traje i kada će signal biti primljen.

Hvatači okvira obično ne dolaze s upravljačkim programima (engl. *drivers*), već je razvijanje programske podrške zadaća korisnika. To omogućuje korisnicima programiranje uređaja prema vlastitim potrebama za najoptimalnije iskorištavanje mogućnosti uređaja, no zahtijeva veću stručnost korisnika.

AMV Grabber uređaj je hvatač okvira koji podržava istovremeno snimanje i reprodukciju video signala. Ima devet ulaznih i devet izlaznih priključaka za kamere, što omogućava simuliranje prostorno raspoređenih kamera po cijelom vozilu. Povezuje se s računalom putem *PCIe* sučelja, što omogućava prijenos podataka sa svih priključaka i upravljanje u realnom vremenu. *AMV Grabber* je prikazan na slici 2.1.



Sl. 2.1. *AMV Grabber* uređaj korišten u ovom radu

AMV Grabber se povezuje na ispitivanu jedinicu (uređaj za izvođenje ADAS algoritama) preko koaksijalnih FPD Link III kablova za prijem i reprodukciju video signala. Sadrži devet komponenti DS90UB91 koje omogućuju serijalizaciju i deserijalizaciju podataka korištenjem dvosmjernog kontrolnog kanala. Parovi komponenata za serijalizaciju i deserijalizaciju koriste se za povezivanja između slikovnih senzora i video procesora *AMV Grabber* uređaja. Serijalizacija je proces pretvorbe podatkovnih struktura (u ovom slučaju video signala) u format kojeg je moguće spremiti u memoriju i poslati na računalo. Deserijalizacija je proces pretvorbe podataka iz memorije uređaja u video signale koji se šalju na izlazne priključke.

Centralna procesorska jedinica realizirana je u hibridnoj programabilnoj mreži Zynq XC7Z030, koja osim FPGA (engl. *field-programmable gate array*) dijela sadrži i dvojezgreni procesor ARM Cortex-A9.

2.2. Kamere korištene pri razvoju korisničke aplikacije

Pri realizaciji zadatka korištena su tri modela kamera: RDACM23-01, RDACM24B-01 i RDACM24B-06. Sva tri modela su kreirana prvenstveno za korištenje u vozilima s ADAS sustavima, te su dizajnirana za odolijevanje teškim uvjetima okoline. Pružaju digitalni video tok elektroničkoj kontrolnoj jedinici (engl. *electronic control unit - ECU*) koja obrađuje podatke i pretače ih u informacije koje pomažu pri vožnji. U ovom slučaju elektronička kontrolna jedinica je *AMV Grabber* uređaj, s kojim se kamere povezuju i pokreću korištenjem koaksijalnih kabela s FAKRA priključkom. Kamere mogu biti konfigurirane korištenjem dvosmjernog kontrolnog kanala.

RDACM23-01 kamera ima OV10635 slikovni senzor i podržava RAW i YUV formate s maksimalnom rezolucijom 1280 x 800 elemenata slike (engl. *pixels*). Pri maksimalnoj rezoluciji moguće je ostvariti maksimalno 30 fps.

RDACM24B-01 i RDACM24B-06 imaju OV10640 slikovni senzor i podržavaju samo RAW format s maksimalnom rezolucijom 1280 x 1080 elemenata slike. Pri maksimalnoj rezoluciji moguće je ostvariti maksimalno 30 fps. Jedina razlika je vrsta leće. RDACM24B-06 ima ravnu leću, dok RDACM24B-01 ima zakrivljenu, tzv. „riblje oko“ (engl. *fisheye*) leću.

Sva tri modela kamera prikazana su na slici 2.2.

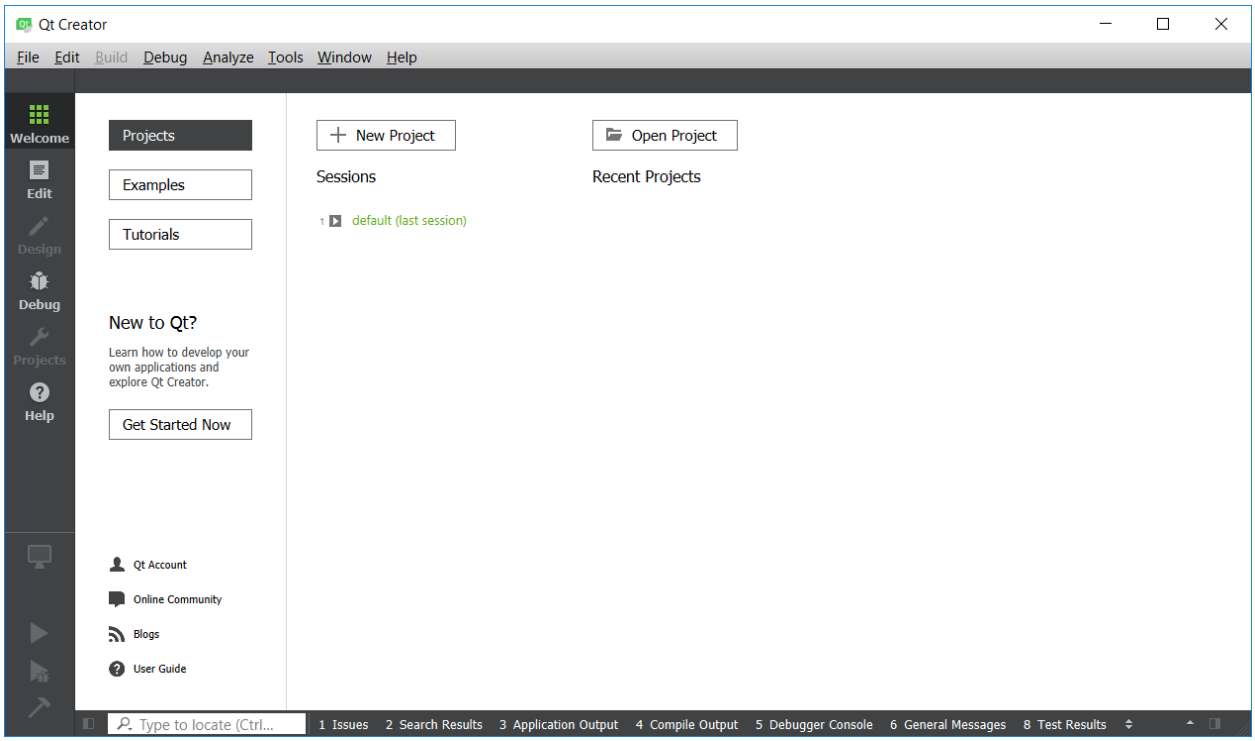
Sl. 2.2. Kamere korištene pri razvoju korisničke aplikacije

2.3. *Qt* programski okvir

Qt je višeplatformski programski okvir (engl. *framework*) za razvoj programske podrške (engl. *software*) za stolna (engl. *desktop*) računala, ugradbene (engl. *embedded*) i prijenosne sustave. Podržane platforme uključuju Linux, Microsoft Windows, OS X, Android, iOS i dr. Zasnovan je na C++ programskom jeziku, što omogućava prevođenje različitim C++ jezičnim prevoditeljima (engl. *compiler*) kao što su Clang, GCC, ICC, MinGW i MSVC. *Qt* je dostupan za razvoj korištenjem dozvole otvorenog koda (engl. *open source*) ili komercijalne dozvole.

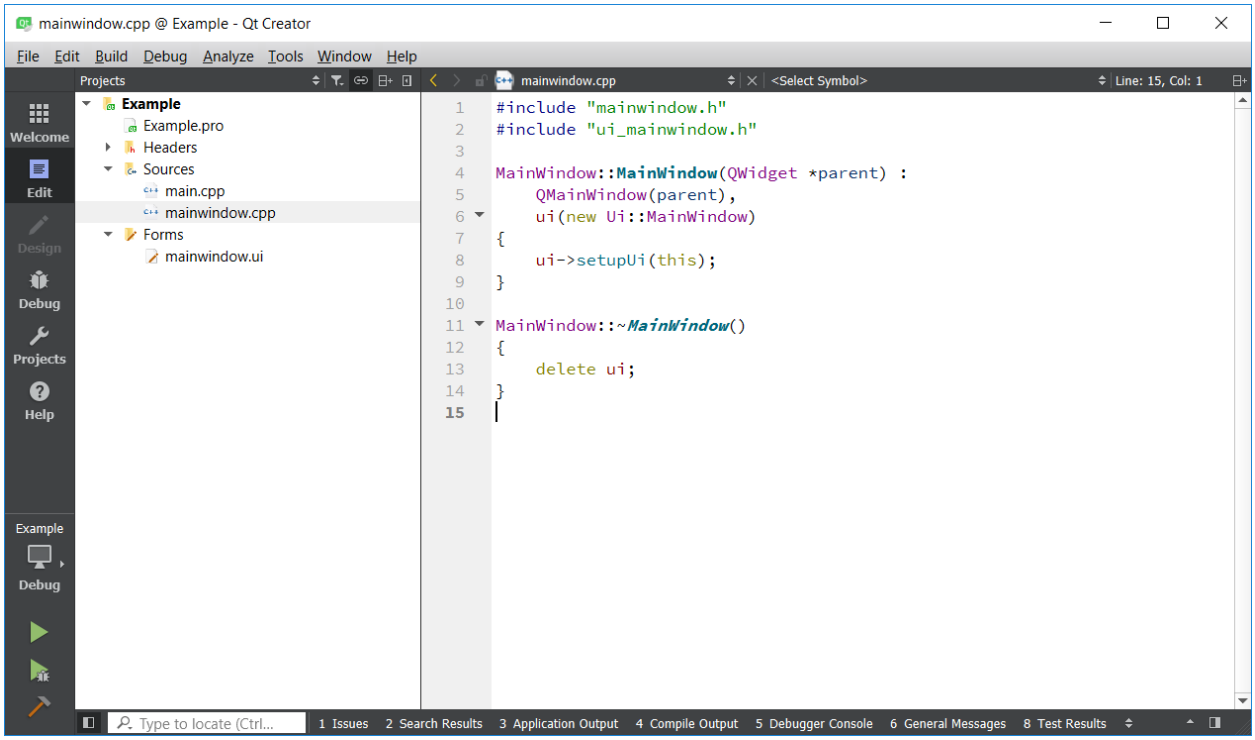
2.3.1 *Qt Creator*

Qt Creator [4] je integrirano razvojno okruženje (engl. *integrated development environment*) koje pruža alate za dizajn i razvoj aplikacija u *Qt* programskom okviru. Omogućava podršku za cijeli ciklus razvoja aplikacije, od kreiranja projekta do objave konačnog rješenja. Uključuje upravitelj projekta koji može koristiti raznolike tipove projekata kao što su *.pro, CMake, Autotools* i dr. Projektna datoteka može sadržavati informacije kao što su uključene izvorne datoteke, biblioteke, vlastite postavke i preferencije za prevođenje i izgradnju (engl. *build*) programskog rješenja. Korisniku je omogućeno jednostavno prebacivanje između različitih postavki, *Qt* verzija i ciljanih platformi uz minimalne izmjene koda. Početni zaslon *Qt Creator* integriranog razvojnog okruženja prikazan je na slici 2.3.



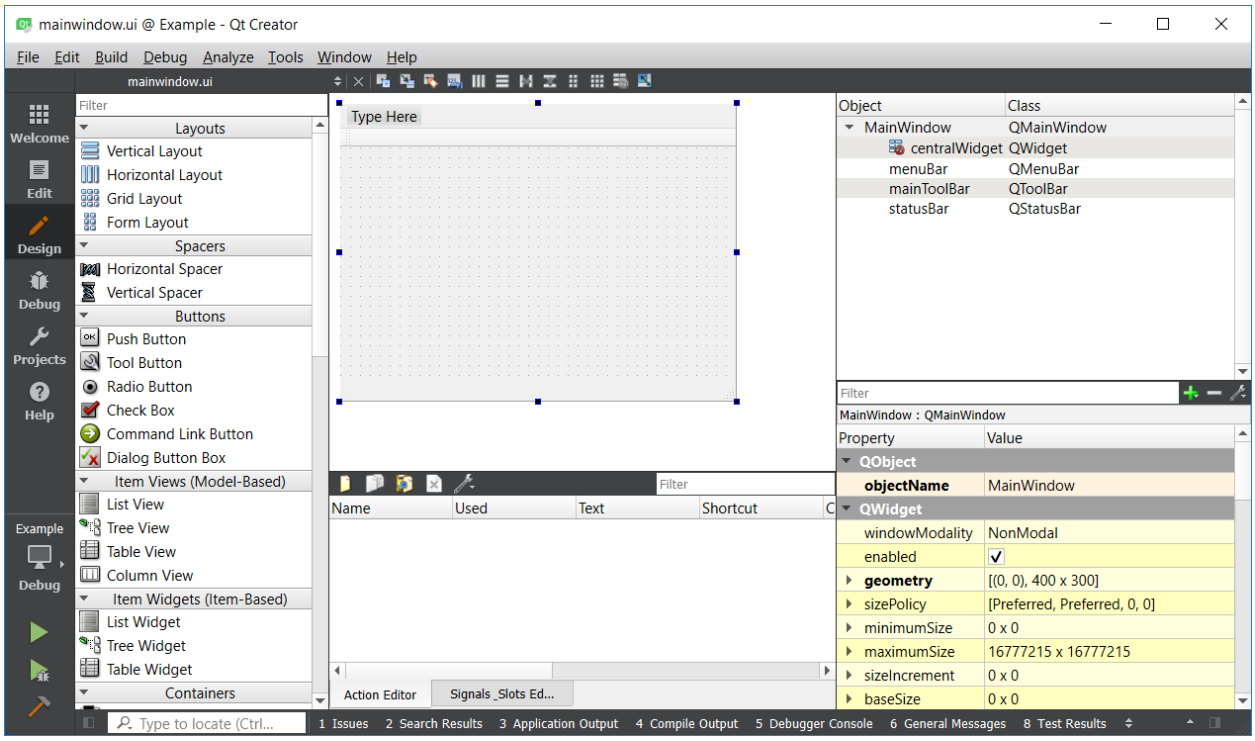
Sl. 2.3. Početni zaslon *Qt Creator* integriranog razvojnog okruženja

Qt uređivač koda ubrzava proces razvoja tako što omogućava provjeru i isticanje sintakse, automatsko završavanje i refaktoriranje koda, povezivanje sa sustavima za verzioniranje, pronalazak pogrešaka (engl. *debugging*), prikaz poruka upozorenja tijekom pisanja koda i slično. Uređivaču koda se pristupa klikom na *Edit* gumb u lijevom izborniku. Primjer izgleda *Qt* uređivača koda tijekom razvoja prikazan je na slici 2.4.



Sl. 2.4. *Qt Creator* uređivač koda

Uređivač grafičkog sučelja *Qt Designer* služi za kreiranje i uređivanje grafičkog sučelja programskih rješenja. Moguće je sastavljati i prilagođavati prozore i dijaloge dodavanjem dostupnih grafičkih elemenata korištenjem „povuci i ispusti“ (engl. *drag and drop*) tehnike. Moguće je i kreiranje vlastitih grafičkih elemenata (engl. *widgets*) korištenjem standardnih elemenata. Sva svojstva i elemente postavljene u uređivaču grafičkog sučelja moguće je dinamički promijeniti u uređivaču koda. Grafički elementi se integriraju s napisanim kodom korištenjem Qt mehanizma signala i utora (engl. *signals and slots*). *Qt Designer* također omogućava učitavanje, pohranjivanje i korištenje vlastitih resursa koji se spremaju kao binarne datoteke u konačnoj izvršnoj datoteci. Svaki kreirani grafički obrazac može imati vlastitu datoteku s resursima. Ova mogućnost je korisna kada aplikacija zahtijeva vanjske resurse, a ne želi se riskirati neočekivanim gubljenjem istih. *Qt Designer* uređivač grafičkog sučelja prikazan je na slici 2.5.



Sl. 2.5. *Qt Designer* uređivač grafičkog sučelja

2.3.2. Sustav meta-objekata

Sustav meta-objekata (engl. *meta-object system*) [5] dio je jezgre *Qt* programskog okvira. Pruža mehanizam signala i utora za unutarnju komunikaciju među objektima, informacije o tipovima podataka za vrijeme izvršavanja (engl. *run-time type information*) te sustav dinamičkih svojstava.

Osnovne značajke sustava meta-objekata:

- *QObject* klasa koja pruža osnovnu klasu za objekte koji mogu iskorištavati prednosti sustava meta-objekata.
- *Q_OBJECT* makro unutar privatne sekcije deklaracije klase koji se koristi za omogućavanje svojstava meta-objekata.
- Prevoditelj meta-objekata (engl. m*eta-object compiler* - *moc*) koji opskrbljuje svaku *QObject* podklasu s potrebnim kodom za implementaciju svojstava meta-objekata.

Kada prevoditelj meta-objekata naiđe na *Q_OBJECT* makro unutar jedne ili više deklaracija klasa, stvara posebnu C++ datoteku koja sadrži kod meta-objekata za svaku klasu. Ta generirana datoteka je obično prevedena i povezana s implementacijom klase.

Korištenje sustava meta-objekata pri kreiranju programskih rješenja omogućava korištenje programskih mogućnosti koje nisu dostupne u nativnom C++ programskom jeziku.

2.3.3. Signali i utori

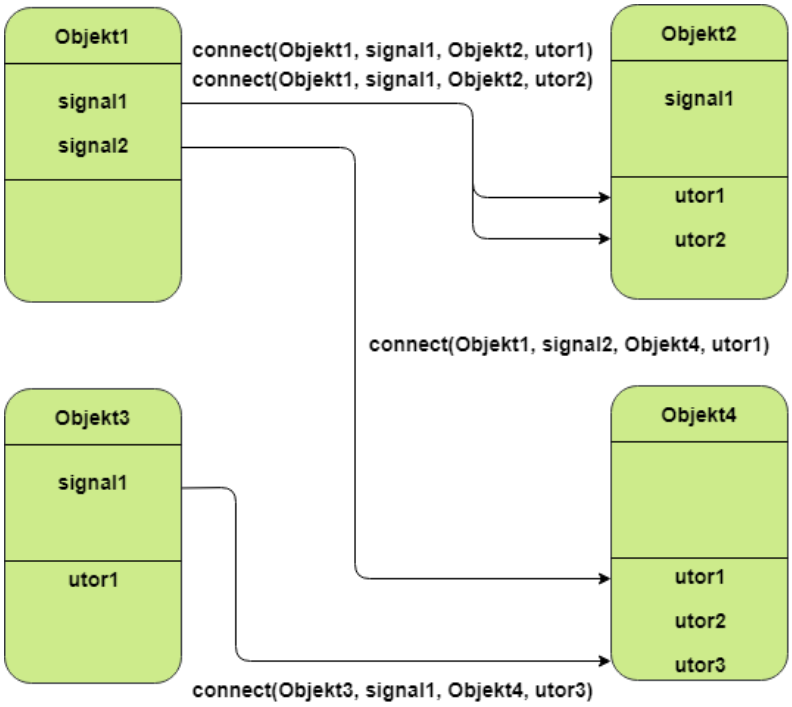
Signali i utori [6] su jedna od ključnih značajki *Qt* programskog okvira. Koriste se za komunikaciju među objektima i potiču korištenje višestruko upotrebljivih komponenata u razvoju programskih rješenja. Predstavljaju alternativu uobičajenim funkcijama povratnog poziva (engl. *callback*). Funkcije povratnog poziva imaju dvije mane. Prvo, nisu tipski sigurne, tj. pri korištenju nije sa sigurnošću moguće znati hoće li funkcija povratnog poziva biti pozvana s točnim argumentima. Drugo, funkcija povratnog poziva je čvrsto povezana s pozivnom funkcijom, jer pozivna funkcija mora znati koju povratnu funkciju pozvati.

Signali su način informiranja objekata o odgovarajućim događajima. Automatski su generirani od strane prevoditelja meta-objekata i ne smiju biti implementirani u *.cpp* datoteci. Također ne smiju imati povratni tip, tj. trebaju biti tipa *void*. Utori su članske (engl. *member*) funkcije koje se pozivaju kao odgovor na određeni signal. Mogu pozivati neovisno o signalima, tj. kao obične C++ funkcije. Objekt koji emitira signal nema saznanja o objektima koji taj signal primaju, dok objekt koji prima signal nema saznanja o tome je li povezan na neki signal, što omogućava pravu informacijsku enkapsulaciju.

Mehanizam signala i utora je tipski siguran. Potpis (engl. *signature*), koji definira naziv, parametre i povratnu vrijednost signala mora odgovarati potpisu primajućeg utora. Utor može imati i kraći potpis (manje parametara) nego signal jer može ignorirati višak parametara. To omogućava izvođenje utora s parametrima signala.

Signal može biti povezan s više utora, te može biti povezan s drugim signalom. Utor može biti povezan s više signala. Svaka *QObject* klasa može imati proizvoljan broj signala i utora, no emitiranje signala je moguće samo iz te klase. Spajanje može biti izravno (sinkrono) ili odgođeno (asinkrono). Kada je signal emitiran, povezani utor je uobičajeno odmah izvršen, kao normalan poziv funkcije. Kada se to dogodi, mehanizam je neovisan o petlji događaja (engl. *event loop*) grafičkog korisničkog sučelja (engl. *graphical user interface*). Mehanizam signala i utora je neznatno sporiji od funkcija povratnog poziva, no jednostavnost i fleksibilnost koje pružaju čine ih vrijednim korištenja.

Postojeće *Qt* klase sadrže mnoge predefinirane signale i utore, no također je moguće definirati i vlastite. Vlastiti signali emitiraju se ključnom riječi *emit*. Signalizacija je moguća i između različitih niti, te je moguće i odspajanje signala i utora tijekom izvođenja programa. Grafički prikaz korištenja signala prikazan je na slici 2.6 [6].



Sl. 2.6. Prikaz mehanizma signala i utora [6]

2.3.4. Qmake

Alat za izgradnju *qmake* [7] pojednostavljuje proces izgradnje programskih rješenja na različitim platformama. Dio je *Qt* okvira od početka i smatra se nativnim alatom. Automatizira proces kreiranja *Makefile* datoteka i može se koristiti za izgradnju bilo kojih programskih rješenja ovisno o platformi na kojoj se koristi. *Qmake* pruža projektno orijentirani sustav za upravljanje i izgradnju aplikacija, biblioteka i drugih projektnih komponenata. Takav pristup omogućava kontrolu nad korištenim izvornim datotekama.

Qmake generira *Makefile* datoteku na temelju informacija dostupnih u projektnoj datoteci, koja je kreirana od strane programera. *Qt* projekt je opisan sadržajem projektne datoteke (*.pro*). Obično sadrže listu izvornih datoteka, konfiguracijske informacije i specifične informacije za programsko rješenje, kao što su uključene biblioteke, putanje izvornih datoteka i sl. Pri kreiranju novog projekta unutar *Qt Creator* integriranog razvojnog okruženja, projektna datoteka se generira sa zadanim vrijednostima za izgradnju i pokretanje projekta. Zadane vrijednosti ovise o vrsti projekta. Projektnu datoteku je moguće ručno mijenjati prema vlastitim potrebama. Nakon izmjene potrebno je pokrenuti *qmake* da bi se potrebni dijelovi projekta ažurirali.

Nije nužno korištenje *qmake* alata pri izgradnji *Qt* projekta, no u tom slučaju posao je dodatno otežan jer je potrebno pokretati prevoditelj meta-objekata za svaku datoteku zaglavlja koja sadrži signale ili utore i općenito se ne preporuča.

3. REALIZACIJA KORISNIČKE APLIKACIJE ZA *AMV GRABBER* UREĐAJ

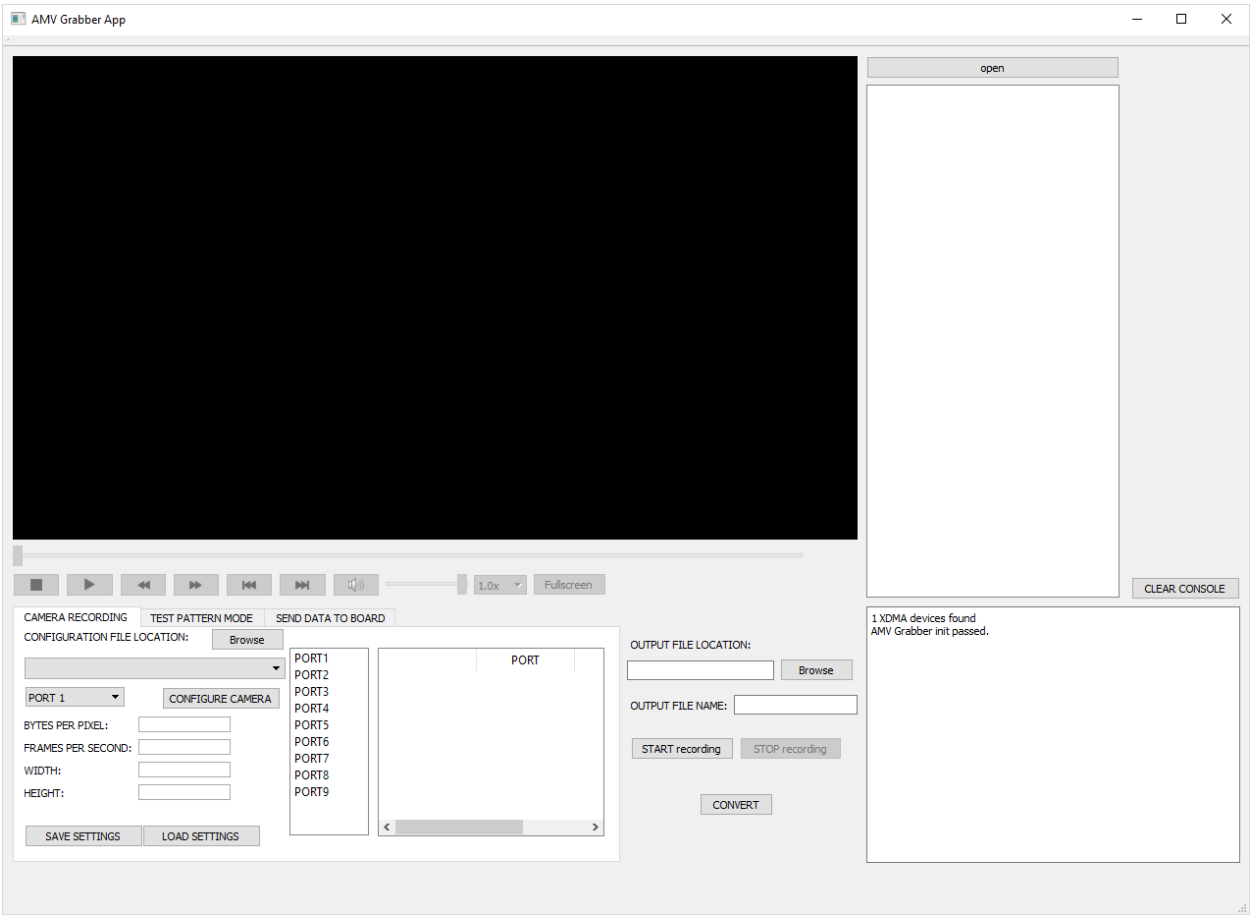
Pri realizaciji korisničke aplikacije korištena je *Qt* verzija 5.10.0 i 64-bitni C++ jezični prevoditelj *Microsoft Visual C++* 15.0. Na *AMV Grabber* uređaju se izvršava programski kod razvijen u C programskom jeziku, a međudjelovanje s korisničkom aplikacijom je omogućeno posredstvom upravljačkog programa temeljenom na *Xilinx* rješenju za izravan pristup memoriji. (engl. *Xilinx direct memory access - XDMA*) [8].

Za ispravan rad, *AMV Grabber* uređaj mora biti povezan s računalom i programski kod mora biti pokrenut, te upravljački program mora biti omogućen na računalu. Ukoliko jedan od tih zahtjeva nije zadovoljen, u korisničkoj aplikaciji će biti prikazana obavijest o greški i neće biti moguće koristiti dijelove aplikacije namijenjene za upravljanje uređajem. U nastavku teksta bit će opisano korištenje aplikacije za slučaj kada su prethodno navedeni zahtjevi zadovoljeni.

Prilikom pokretanja korisničke aplikacije obavlja se inicijalizacija i povezivanje aplikacije s upravljačkim programom. Kreiraju se četiri ulazno/izlazna komunikacijska kanala koji se koriste za interakciju s *AMV Grabber* uređajem. Kreiraju se kanal za slanje kontrolnih parametara iz aplikacije do uređaja, kanal za dohvaćanje povratne informacije s uređaja u aplikaciji, kanal za prijenos podataka s uređaja do aplikacije tijekom snimanja, te kanal za prijenos podataka s računala do uređaja tijekom reprodukcije.

Korisnička aplikacija sadrži grafičke elemente koji pri interakciji izvršavaju odgovarajući programski kod. Početni zaslon korisničke aplikacije prikazan je na slici 3.1. Za upravljanje *AMV Grabber* uređajem postoje kartice (engl. *tabs*) koje sadrže grafičke elemente za različite načine rada uređaja. Kartica *CAMERA RECORDING* sadrži elemente koji se koriste za konfiguraciju i snimanje kamerama. Kartica *TEST PATTERN MODE* sadrži elemente koji se koriste za konfiguraciju i snimanje ispitnog uzorka (engl. *test pattern*) u kojem se ne koriste kamere. Kartica *SEND DATA TO BOARD* sadrži elemente za reprodukciju snimljenih podataka, tj. prijenos podataka s računala u memoriju uređaja. Na početku su grafički elementi koji se koriste za upravljanje uređajem onemogućeni. Ukoliko su inicijalizacija i povezivanje uspješno izvršeni, odgovarajući grafički elementi su omogućeni i moguće je koristiti funkcionalnosti za upravljanje uređajem. Aplikacija također sadrži grafičke elemente za pretvorbu snimljenih podataka u video format pogodan za prikaz na računalu. Za reprodukciju video sadržaja u aplikaciji postoje grafički elementi za dodavanje video zapisa na popis za reprodukciju, element za prikaz video sadržaja, te

elementi za upravljanje reprodukcijom. Aplikacija sadrži i element koji služi kao konzola za ispis kontrolnih poruka, te gumb *CLEAR CONSOLE* za čišćenje konzole.



Sl. 3.1. Početni zaslon korisničke aplikacije za *AMV Grabber* uređaj

3.1. Upravljanje snimanjem *AMV Grabber* uređajem

3.1.1. Snimanje kamerama

Prije snimanja potrebno je inicijalizirati i konfigurirati kamere, tj. upisati podešavajuće vrijednosti u odgovarajuće registre kamere. Prvi korak je odabir i učitavanje putanje konfiguracijske datoteke. Konfiguracijske datoteke su JSON (engl. *JavaScript Object Notation*) formata i sadrže objekt sa četiri cjelobrojna inicijalizacijska podatka za kameru koja se konfigurira: broj bajtova po elementu slike, broj fps, širina i visina u broju elemenata slike. Također sadrže polje objekata gdje se svaki objekt sastoji od dvije heksadekadske vrijednosti: adresa registra i vrijednost koja se upisuje u taj registar. Aplikacija očekuje postojanje objekta *initialSettings* s ključevima *bytesPerPixel*, *width*, *height* i *framesPerSecond* kojima su pridružene cjelobrojne vrijednosti, te postojanje polja objekta *registers* gdje se svaki objekt sastoji od ključeva *address* i *value* kojima su pridruženi tekstualni nizovi koji predstavljaju heksadekadske vrijednosti. Podaci unutar objekta *initialSettings* moraju

odgovarati odgovarajućim vrijednostima registara unutar polja objekata *registers*. Podudaranje tih podataka je važno zbog dobivanja ispravnog video sadržaja s kamere i zadaća je korisnika osigurati tu podudaranost. Konfiguracijske datoteke moraju se pridržavati definiranog formata da bi se vrijednosti mogle ispravno pročitati u aplikaciji, što je također zadaća korisnika. Primjer konfiguracijske datoteke prikazan je na slici 3.2.

Sl. 3.2. Primjer konfiguracijske datoteke za podešavanje kamere

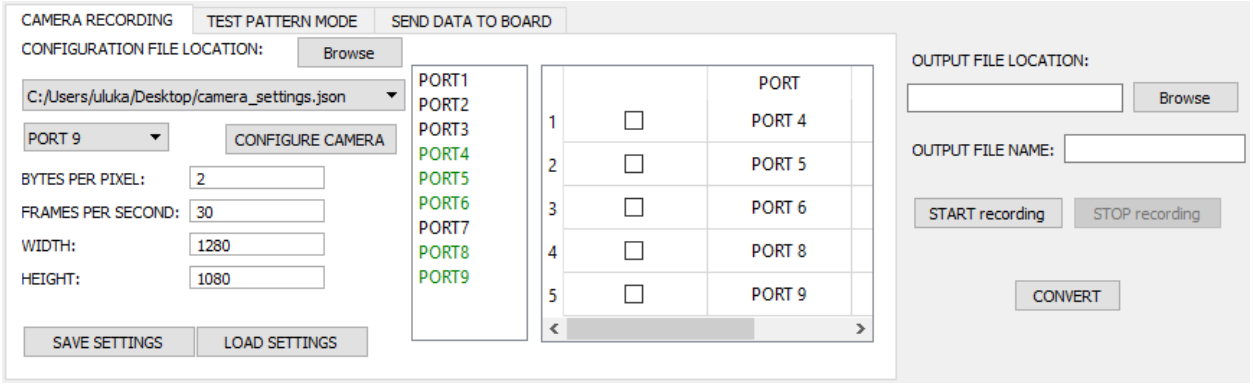
Za pohranu konfiguracijskih i ostalih potrebnih parametera za svaki priključak, kreiran je vektor strukturnog tipa koji sadrži devet strukturnih varijabli, tj. po jednu za svaki priključak *AMV Grabber* uređaja.

Klikom na gumb *Browse* otvara se istraživač datoteka u kojem je moguće odabrati željenu datoteku čija se putanja pohranjuje unutar aplikacije i prikazuje u padajućem izborniku. Nakon učitavanja potrebno je u odgovarajućem padajućem izborniku odabrati priključak na koji je spojena kamera.

Ukoliko nisu odabrani putanja konfiguracijske datoteke i priključak u padajućim izbornicima, klikom na gumb *CONFIGURE CAMERA* izbaciti će se obavijest o greški i konfiguracija se neće pokrenuti. U suprotnom, kreira se objekt radnik (engl. *worker object*) koji sadrži metodu za inicijalizaciju i konfiguraciju kamere, te se kreira nit u kojoj će se ta metoda izvoditi. Nakon toga, obavljaju se povezivanja korištenjem mehanizma signala i utora prema [9] nakon čega se pokreće izvođenje metode u kreiranoj niti i svi ostali gumbi za upravljanje *AMV Grabber* uređajem su onemogućeni, kako bi se izbjegla neželjena ponašanja sustava.

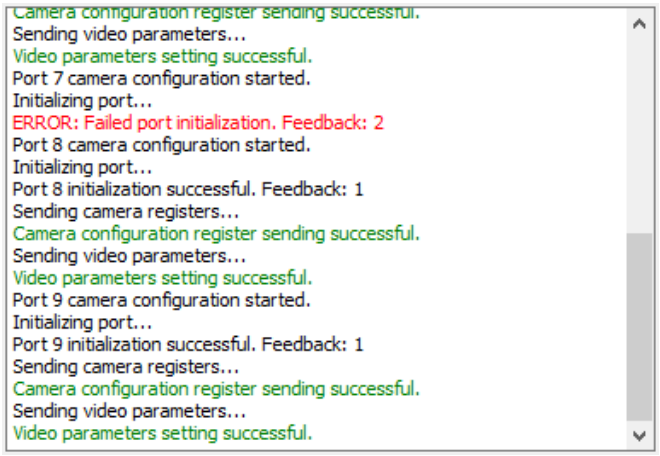
Ukoliko nema priključene kamere na odabranom priključku, dobit će se povratna informacija s uređaja te će se u aplikaciji ispisati poruka o greški. Ukoliko je kamera uspješno inicijalizirana, obavlja se konfiguracija kamera podacima iz odabrane konfiguracijske datoteke gdje se podaci iz objekta *initialSettings* čitaju i šalju do kamere, a zatim pohranjuju unutar vektora za odgovarajući priključak, te se prikazuju u odgovarajućim grafičkim elementima. Nakon toga se podaci iz polja objekata *registers* šalju do kamere jedan po jedan. U slučaju pogreške tijekom konfiguracije, izvođenje se prekida i u aplikaciji se ispisuje poruka s kodom greške. Ukoliko se s uređaja dobije povratna vrijednost koja označava uspješnu konfiguraciju, u aplikaciji se ispisuje poruka o uspješnosti i priključak postaje dostupan za odabir prilikom snimanja. Nakon završetka izvođenja aplikacija se vraća u stanje prije pokretanja konfiguracije.

Slika 3.3. prikazuje primjer priključaka na kojima su uspješno konfigurirane kamere i omogućeni okviri za izbor kojima je moguće te priključke odabrati za snimanje. Slika 3.4. prikazuje primjer kontrolnih poruka u konzoli nakon konfiguriranja kamera. Za pokretanje snimanja potrebno je odabrati priključke s čijih kamera se želi snimati, odabrati putanju odredišne datoteke *OUTPUT FILE LOCATION* klikom na odgovarajući gumb *Browse*, te je potrebno unijeti ime odredišne datoteke. Podaci sa svih kamera koje se koriste u ciklusu snimanja pohranjuju se u istu odredišnu datoteku. Prema [10], ime odredišne datoteke ne smije imati rezervirane znakove, kao što su < > : " / \ | ? *.



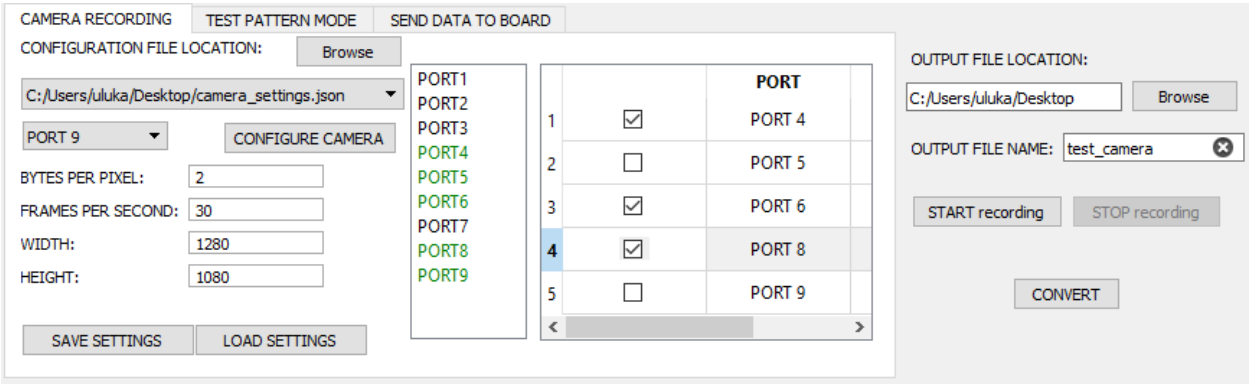
Sl. 3.3. Primjer konfiguriranih priključaka

Ukoliko nije odabran niti jedan priključak, putanja odredišne datoteke ili nije uneseno ime datoteke, klikom na gumb *START recording* izbacit će se obavijest o greški i snimanje neće biti pokrenuto. U suprotnom, podaci pročitani iz konfiguracijske datoteke i spremljeni u vektoru za svaki odabrani priključak upisuju se u odredišnu datoteku u heksadekadskom formatu. Prva 1024 bajta su rezervirana za zaglavlje datoteke. Ti podaci govore s kojih priključaka je snimano i s kojim parametrima, a koriste se pri pretvorbi i slanju podataka s računala na *AMV Grabber* uređaj.

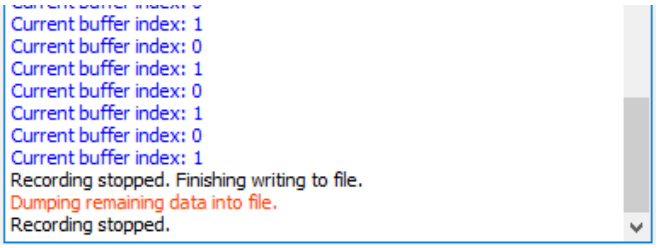


Sl. 3.4. Primjer ispisa kontrolnih poruka nakon konfiguriranja kamera

Nakon uspješnog zapisivanja parametara kreiraju se dva objekta radnika. Jedan objekt radnik sadrži metodu za komunikaciju i čitanje podataka s uređaja, dok drugi sadrži metodu za zapisivanje podataka u odredišnu datoteku. Također se kreiraju dvije niti u kojima će se te metode izvoditi. Nakon toga, obavljaju se povezivanja korištenjem mehanizma signala i utora prema [9] nakon čega se pokreću izvođenja funkcija u kreiranim nitima i svi ostali gumbi za upravljanje *AMV Grabber* uređajem osim gumba *STOP recording* su onemogućeni. Klikom na gumb *STOP recording*, snimanje, prijenos i zapisivanje podataka se zaustavljaju te se aplikacija vraća u stanje prije pokretanja snimanja. Slika 3.5. prikazuje primjer ispravnog stanja prije pokretanja snimanja, dok slika 3.6. prikazuje primjer kontrolnih poruka u konzoli nakon završetka snimanja.



Sl. 3.5. Primjer ispravnog stanja prije pokretanja snimanja

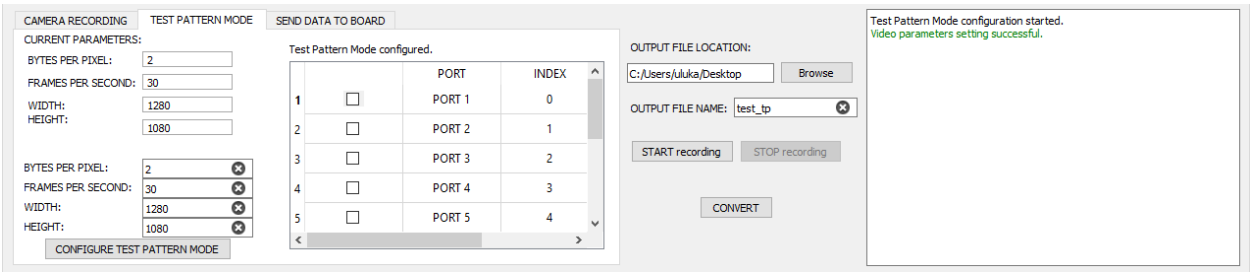


Sl. 3.6. Primjer ispisa kontrolnih poruka nakon zaustavljanja snimanja

3.1.2. Snimanje ispitnog uzorka

Snimanje ispitnog uzorka ne zahtijeva korištenje kamera i korisno je za testiranje jer omogućuje iste procese kao snimanje s kamerama u slučaju kada kamere nisu dostupne. Pri pokretanju aplikacije ispitni uzorak nije konfiguriran i okviri za izbor priključaka za snimanje ispitnog uzorka su onemogućeni. Prvo je potrebno unijeti četiri parametra za konfiguraciju ispitnog uzorka: broj bajtova po elementu slike, broj fps, širina i visina u broju elemenata slike. Ukoliko nisu unesena sva četiri parametra u odgovarajuća polja, klikom na gumb *CONFIGURE TEST PATTERN MODE* izbacit će se obavijest o greški i konfiguracija se neće pokrenuti. U suprotnom, kreira se objekt radnik koji sadrži metodu za konfiguraciju testnog uzorka, te se kreira nit u kojoj će se ta metoda izvoditi. Nakon toga, obavljaju se povezivanja korištenjem mehanizma signala i utora prema [9] nakon čega se pokreće izvođenje metode u kreiranoj niti i svi ostali gumbi za upravljanje *AMV Grabber* uređajem su onemogućeni.

Kod konfiguracije se sva četiri podatka šalju i pohranjuju na *AMV Grabber* uređaj te se također pohranjuju u aplikaciji. U slučaju pogreške tijekom konfiguracije, izvođenje se prekida i u aplikaciji se ispisuje poruka s kodom greške, te se okviri za izbor priključaka za snimanje onemogućuju ukoliko su prethodno bili omogućeni ispravnim konfiguriranjem. Ukoliko se s uređaja dobije povratna vrijednost koja označava uspješnu konfiguraciju, u aplikaciji se ispisuje poruka o uspješnosti i omogućavaju se okviri za izbor priključaka, te se parametri prikazuju u odgovarajućim grafičkim elementima. Nakon završetka izvođenja aplikacija se vraća u stanje prije pokretanja konfiguracije. Slika 3.7. prikazuje primjer nakon uspješne konfiguracije ispitnog uzorka.



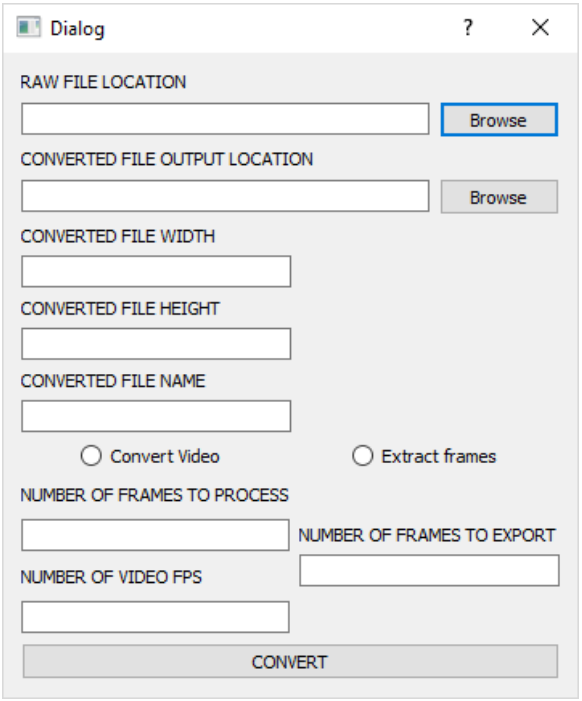
Sl. 3.7. Izgled aplikacije nakon konfiguriranja ispitnog uzorka

Postupak pokretanja i zaustavljanja snimanja testnog uzorka je jednak kao i kod snimanja s kamerama, samo što u ovom slučaju svi priključci imaju jednake parametre.

3.2. Pretvorbe snimljenih podataka

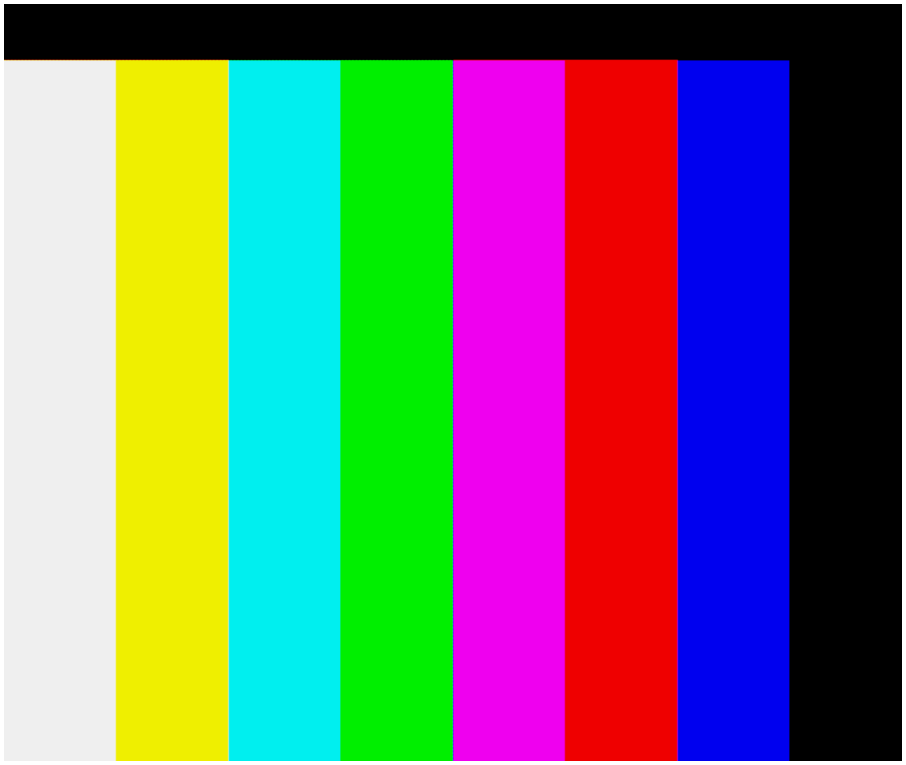
Podaci snimljeni kamerama spremljeni su u odredišnoj datoteci u RAW formatu, kojeg nije moguće reproducirati na računalu. Zato je potrebno obaviti pretvorbu tih podataka u format pogodan za reprodukciju video zapisa. U ovom slučaju, pretvoreni video zapisi su MPG formata. Klikom na gumb *CONVERT* na glavnom zaslonu aplikacije otvara se novi prozor za pretvorbu prikazan na slici 3.8. Osim pretvorbe, moguće je obaviti i izdvajanje željenog broja okvira, gdje se generiraju slikovne datoteke BMP formata za pojedini okvir.

Željena snimljena datoteka odabire se klikom na odgovarajući gumb *Browse*, nakon čega se otvara istraživač datoteka. Nakon odabira, putanja se pohranjuje u polje *RAW FILE LOCATION*. Putanju odredišnih datoteka odabire se klikom na odgovarajući gumb *Browse*, nakon čega se odredišna putanja pohranjuje u polje *CONVERTED FILE OUTPUT LOCATION*. Potrebno je unijeti visinu i širinu u broju elemenata slike koje moraju odgovarati vrijednostima u zaglavlju datoteke da bi se dobio ispravan pretvoreni zapis, te je potrebno unijeti ime odredišne datoteke koje ne smije imati rezervirane znakove, prema [10]. Ukoliko se odabire pretvorba video zapisa, potrebno je unijeti broj okvira koje treba obraditi te fps. Ukoliko se odabere izdvajanje okvira potrebno je unijeti željeni broj okvira. Klikom na gumb *CONVERT* obavlja se provjera jesu li sve vrijednosti zadovoljavajuće postavljene i unesene. Ukoliko jesu, započinje pretvorba u odabranom načinu rada, a u suprotnom se izbacuje obavijest o greški i pretvorba se neće pokrenuti.

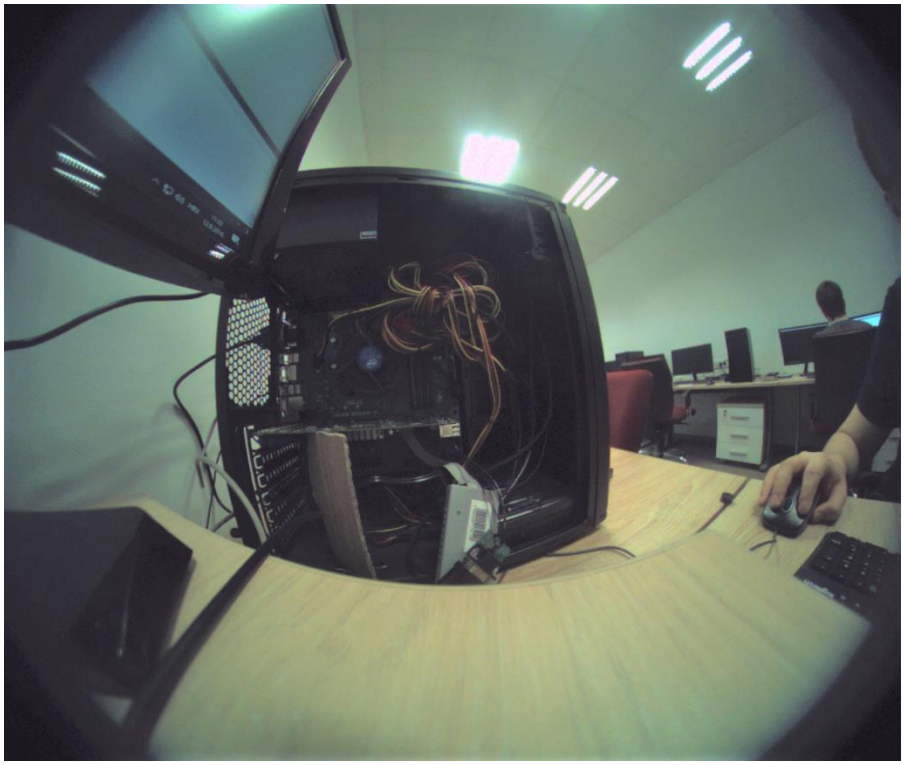


Sl. 3.8. Prozor za pokretanje pretvorbe

Primjer okvira dobivenog izdvajanjem iz datoteke s podacima dobivenim snimanjem ispitnog uzorka prikazan je na slici 3.9., a primjer okvira dobivenog izdvajanjem iz datoteke s podacima dobivenim snimanjem kamerama prikazan je na slici 3.10.



Sl. 3.9. Izdvojeni okvir dobiven snimanjem ispitnog uzorka

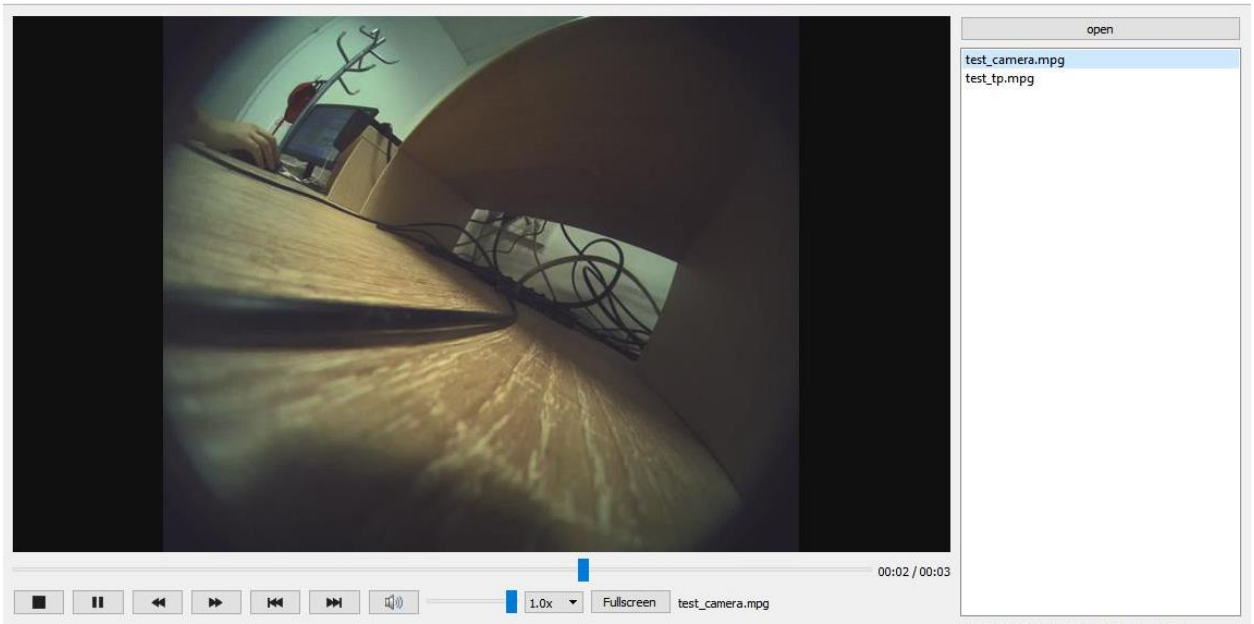


Sl. 3.10. Izdvojeni okvir dobiven snimanjem kamerama

3.3. Prikaz snimljenog video sadržaja u korisničkoj aplikaciji

Qt programski okvir podržava aplikacijska programska sučelja (engl. *application programming interface*) koja sadrže C++ klase i grafičke elemente za rukovanje multimedijskim sadržajima. Da bi se ta programska sučelja mogla koristiti, potrebno ih je uključiti u projektnu datoteku. Za omogućavanje reprodukcije video sadržaja u korisničkoj aplikaciji za *AMV Grabber* uređaj uključeni su *Qt Multimedia* i *Qt Multimedia Widgets* moduli [11].

Sučelje za reprodukciju video sadržaja sastoji se od grafičkog elementa koji služi kao lista za reprodukciju, grafičkog elementa koji služi kao pozadina za prikaz video sadržaja, horizontalnih klizača za premotavanje video sadržaja i kontrolu glasnoće, padajućeg izbornika kojim je moguće mijenjati brzinu reprodukcije video sadržaja, te kontrolnih gumba za dodavanje video zapisa na listu, pokretanje/pauziranje, zaustavljanje, premotavanje naprijed ili nazad, odabir idućeg ili prethodnog video zapisa na listi, onemogućavanje ili omogućavanje zvuka, te gumb za prikaz na cijelom zaslonu. Primjer reprodukcije prikazan je na slici 3.11.



Sl. 3.11. Primjer reprodukcije video sadržaja

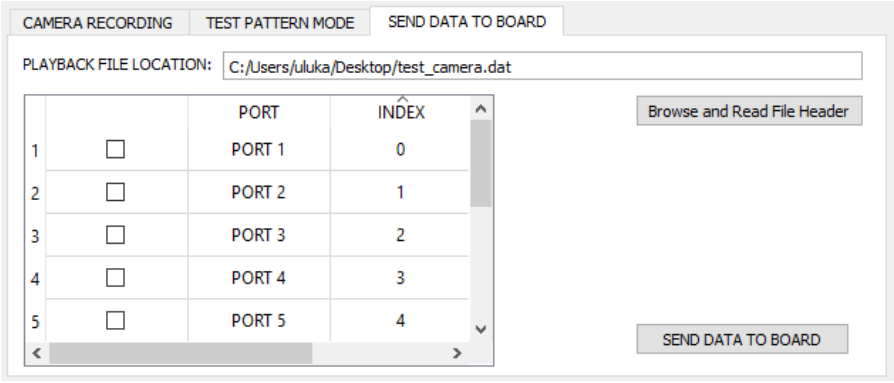
Ukoliko na listi za reprodukciju nema dodanih video zapisa, svi grafički elementi su onemogućeni osim gumba za dodavanje video zapisa. Kako je pretvaranjem u aplikaciji omogućena pretvorba samo u video zapise MPG formata, tako je učitavanje i reprodukcija omogućena samo za video zapise tog formata. Nakon dodavanja barem jednog video zapisa na listu za reprodukciju, svi elementi su omogućeni i moguće ih je koristiti.

Qt ne sadrži grafički element koji služi kao pozadina za prikaz, pa je to omogućeno prema [12]. Također je kreiran prilagođeni model koji nasljeđuje standardnu *QVideoWidget* klasu. U modelu su implementirane metode za izlazak iz prikaza na cijelom zaslonu dvostrukim miša ili pritiskom na tipku *escape*.

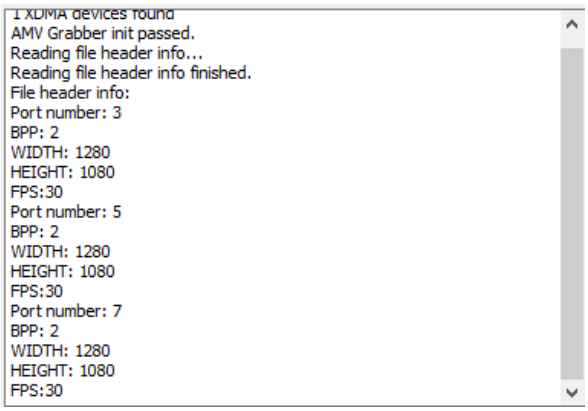
3.4. Reprodukcija snimljenih podataka

Reprodukcija snimljenih podataka znači prijenos podataka koji su pohranjeni na računalu u RAW formatu na *AMV Grabber* uređaj kako bi se oni mogli proslijediti na izlazne priključke uređaja.

Prvo je potrebno učitati datoteku sa snimljenim podacima u RAW formatu. Klikom na gumb *Browse and Read File Header* se obavlja čitanje zaglavlja datoteke upisanog prije snimanja. Ako je čitanje uspješno, pročitani parametri se zajedno s putanjom datoteke pohranjuju u aplikaciji, ispisuju u konzoli i omogućavaju se okviri za izbor priključaka na koje se želi poslati podatke. U slučaju da datoteka ne sadrži zaglavlje ili je čitanje neuspješno, ispisuje se poruka o greški i okviri za izbor se onemogućuju. Slika 3.12. prikazuje primjer uspješnog čitanja zaglavlja datoteke nakon čega su omogućeni okviri za izbor priključaka, dok slika 3.13. prikazuje ispis parametara u konzoli.

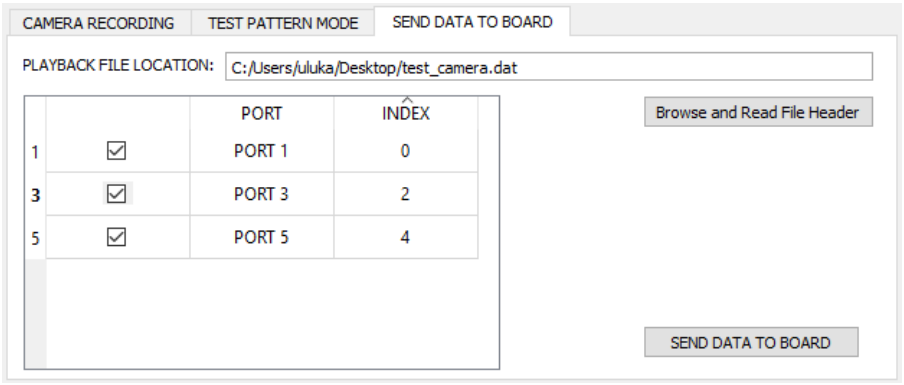


Sl. 3.12. Primjer okvira za izbor nakon uspješnog čitanja zaglavlja datoteke



Sl. 3.13. Primjer ispisa parametara nakon uspješnog čitanja zaglavlja datoteke

Korisnik ima mogućnost izabrati onoliko priključaka na koje želi slati podatke koliko ih je pročitano iz zaglavlja datoteke. To je prikazano na slici 3.14., gdje se za tri pročitana priključka može odabrati tri priključka na koje se želi pročitane podatke proslijediti. Ako je broj odabranih priključaka za slanje jednak broju pročitanih priključaka, ostali okviri za izbor postaju nedostupni.



Sl. 3.14. Primjer odabira priključaka za slanje podataka

Ukoliko nije odabran niti jedan priključak za slanje podataka, klikom na gumb *SEND DATA TO BOARD* izbacit će se obavijest o greški i slanje podataka neće biti pokrenuto. U suprotnom, kreiraju se tri objekta radnika. Prvi objekt sadrži metodu za osluškivanje povratnih informacija s *AMV Grabber* uređaja, drugi sadrži metodu za čitanje podataka iz datoteke, a treći sadrži metodu za slanje podataka na uređaj. Također se kreiraju tri niti u kojima će se te metode izvoditi. Nakon toga, obavljaju se povezivanja korištenjem mehanizma signala i utora prema [9] nakon čega se pokreću izvođenja funkcija u kreiranim nitima i svi ostali gumbi za upravljanje *AMV Grabber* uređajem su onemogućeni. Aplikacija samo obavlja prijenos podataka i informacija o priključcima u memoriju uređaja. Daljnja obrada tih podataka zadaća je uređaja.

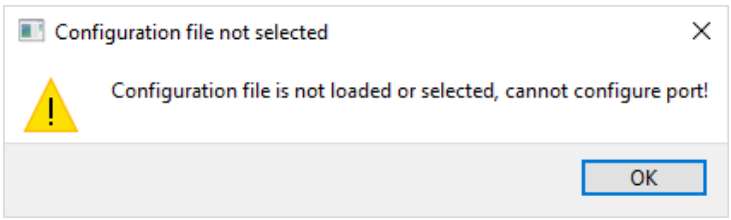
4. TESTIRANJE FUNKCIONALNOSTI PROGRAMSKOG RJEŠENJA

Testiranje je obavljano pri svakom dodavanju novih funkcionalnosti u aplikaciju, gdje su se otkrivale i popravljale greške i nestabilnosti. Na kraju razvoja aplikacije provedena su konačna testiranja funkcionalnosti gdje je promatrano ponašanje aplikacije u različitim situacijama. Prvo je testirano stanje korisničke aplikacije kada *AMV Grabber* uređaj nije povezan s računalom, te kada je uređaj povezan, a upravljački program nije pokrenut. Promatrano je odgovaraju li ponašanja aplikacije očekivanim stanjima. Ponašanje aplikacije u oba slučaja treba biti isto, a očekivana stanja aplikacije su:

- Gumbi za kontrolu uređaja su onemogućeni.
- U konzoli se ispisuje poruka o greški.
- Aplikacije ne ispada iz rada zbog toga što nije uspjelo povezivanje s uređajem.
- Moguće je koristiti funkcionalnosti za pretvorbu prethodno snimljenih podataka.
- Moguće je koristiti funkcionalnosti za prikaz video sadržaja.

Prvim testiranjem sva su stanja zadovoljena i utvrđeno je da se aplikacija izvodi prema očekivanjima.

Korisnička aplikacija kreirana je za upravljanje *AMV Grabber* uređajem i većina mogućnosti aplikacije je onemogućena kada uređaj nije povezan s računalom, pa je provjeru potpune ispravnosti rada aplikacije potrebno provesti kada je uređaj povezan i kada je upravljački program pokrenut. Također je potrebno pobrinuti se da se programska podrška koja se izvršava na uređaju podudara s programskom podrškom upravljačkog programa i korisničke aplikacije, tj. da podržavaju međudjelovanje. Nakon prvog slučaja (bez povezanog uređaja s računalom ili s povezanim uređajem ali bez pokrenutog upravljačkog programa) uređaj je povezan i upravljački program je pokrenut te je obavljeno drugo testiranje. Testirano je pojavljuju li se obavijesti o greški kada korisnik klikne na odgovarajući gumb ako nisu zadovoljeni uvjeti za daljnje izvođenje, te odgovara li izvođenje programskog koda na uređaju događajima u aplikaciji i obrnuto. Slika 4.1. prikazuje primjer prozora s obavijesti o greški.



Sl. 4.1. Primjer prozora s obavijesti o greški

Prilikom testiranja korišten je kabel za serijsku komunikaciju kojim se ostvaruje dodatna komunikacija između uređaja i računala. Korišteno je *PuTTY* [13] programsko rješenje za ispis poruka u konzoli. Serijskom komunikacijom moguće je pratiti izvođenje programskog koda na uređaju i promatrati podudara li se ono s događajima u aplikaciji.

Drugo testiranje sastojalo se od više koraka, gdje se svakim korakom testirao dio korisničke aplikacije. Promatrano je odgovaraju li ponašanja aplikacije očekivanim stanjima. Prvo je testirano upravljanje snimanja kamerama, gdje su očekivana stanja aplikacije:

1. Klikom na gumb za konfiguraciju kamere, ako nije odabrana konfiguracijska datoteka pojavi se obavijest o greški i konfiguracija se ne pokreće.
2. Klikom na gumb za konfiguraciju kamere s učitanom konfiguracijskom datotekom koja ne zadovoljava definirani format, konfiguracija se ne izvodi ispravno i pojavi se obavijest o greški.
3. Klikom na gumb za konfiguraciju kamere sa zadovoljenim prethodno navedenim stanjima, konfiguracija se pravilno izvodi na uređaju i u aplikaciji.
4. Klikom na gumb za pokretanje snimanja, ako nije odabran barem jedan konfigurirani priključak pojavi se obavijest o greški i snimanje se ne pokreće.
5. Klikom na gumb za pokretanje snimanja, ako nije odabrana putanja odredišne datoteke pojavi se obavijest o greški i snimanje se ne pokreće.
6. Klikom na gumb za pokretanje snimanja, ako nije uneseno ime datoteke bez rezerviranih znakova pojavi se obavijest o greški i snimanje se ne pokreće.
7. Klikom na gumb za pokretanje snimanja sa zadovoljenim prethodno navedenim stanjima, snimanje se pravilno izvodi na uređaju i u aplikaciji, a gumbi za upravljanje uređajem osim gumba za zaustavljanje snimanja su onemogućeni.
8. Klikom na gumb za zaustavljanje, snimanje, prijenos i zapisivanje podataka na uređaju i aplikaciji se zaustavljaju, te se uređaj i aplikacija vraćaju u stanje prije pokretanja snimanja.

Testiranjem snimanja kamerama sva su stanja zadovoljena i utvrđeno je da se aplikacija izvodi prema očekivanjima.

Nakon testiranja snimanja kamerama, testirano je snimanje ispitnog uzorka. Izvođenje se većinom podudara sa snimanjem kamerama. Očekivana stanja aplikacije kod upravljanja snimanja testnog uzorka su:

1. Klikom na gumb za konfiguriranje ispitnog uzorka, ako nisu unesena sva četiri parametra u odgovarajuća polja, pojavi se obavijest o greški i konfiguracija se ne pokreće.

2. Klikom na gumb za konfiguraciju kamere sa zadovoljenim prethodno navedenim stanjima, konfiguracija se pravilno izvodi na uređaju i u aplikaciji.
3. Ostala očekivana stanja su jednaka kao točke 4 - 8 za upravljanja snimanjem kamerama.

Testiranjem snimanja ispitnog uzorka sva su stanja zadovoljena i utvrđeno je da se aplikacija izvodi prema očekivanjima.

Testiranje pretvorbi snimljenih podataka provedeno je za podatke dobivene snimanjem uređajem. Promatrano je odgovaraju li ponašanja aplikacije očekivanim stanjima i izvode li se algoritmi pretvorbe ispravno, odnosno dobije li se očekivani rezultat pretvorbe. Očekivana stanja aplikacije kod upravljanja pretvorbama snimljenih podataka su:

1. Klikom na gumb za pretvorbu snimljenih podataka, ako nije odabrana željena snimljena datoteka ili ako nije odabrana putanja odredišne datoteke, pojavi se odgovarajuća obavijest o greški i pretvorba se ne pokreće.
2. Klikom na gumb za pretvorbu snimljenih podataka, ako nisu unesena visina, širina ili ime datoteke bez rezerviranih znakova, pojavi se odgovarajuća obavijest o greški i pretvorba se ne pokreće.
3. Klikom na gumb za pretvorbu, ako nije odabran jedan od dva moguća načina pretvorbe pojavi se obavijest o greški i pretvorba se ne pokreće.
4. Klikom na gumb za pretvorbu, ako je odabrana pretvorba vide zapisa, a nisu uneseni broj okvira koje treba obraditi ili broj fps, pojavi se odgovarajuća obavijest o greški i pretvorba se ne pokreće.
5. Klikom na gumb za pretvorbu, ako je odabrano izdvajanje okvira, a nije unesen broj okvira, pojavi se odgovarajuća obavijest o greški i pretvorba se ne pokreće.
6. Pokretanjem pretvorbe u željenom načinu rada sa zadovoljenim prethodno navedenim stanjima, pretvorba se pravilno izvodi u aplikaciji.

Testiranjem pretvorbe snimljenih podataka sva su stanja zadovoljena, dobiju se ispravni rezultati pretvorbe za oba načina, te je utvrđeno je da se aplikacija izvodi prema očekivanjima.

Testiranje prikaza snimljenog video sadržaja provedeno je za podatke dobivene pretvorbom u aplikaciji. Promatrano je odgovaraju li ponašanja aplikacije očekivanim stanjima. Očekivana stanja aplikacije kod prikaza snimljenog video sadržaja su:

1. Svi grafički elementi za upravljanje prikazom video sadržaja osim gumba za dodavanje video zapisa su onemogućeni ako na popisu za reprodukciju nema dodanih video zapisa.
2. Dodavanjem barem jednog video zapisa na popis za reprodukciju, svi grafički elementi za upravljanje su omogućeni i moguće ih je koristiti.

3. Svaki grafički element ispravno obavlja namijenjenu radnju.

Testiranjem prikaza snimljenog video sadržaja otkriveno je da na računalu mora biti instaliran skup audio i video kodera-dekodera (engl. *codec*) kako bi prikaz bio moguć u aplikaciji. U ovom slučaju instaliran je *K-Lite Codec Pack*, nakon čega je prikaz omogućen. Osim toga, sva su stanja zadovoljena, te je utvrđeno da se aplikacija izvodi prema očekivanjima.

Posljednja je testirana reprodukcija snimljenih podataka, kod koje su očekivana stanja aplikacije:

1. Klikom na gumb za otvaranje i čitanje datoteke sa snimljenim podacima, ako datoteka ne sadrži zaglavlje ili je čitanje neuspješno, pojavi se odgovarajuća obavijest o greški i prijenos podataka do uređaja nije moguće pokrenuti.
2. Klikom na gumb za pokretanje prijenosa na uređaj, ako nije odabran odgovarajući broj priključaka, pojavi se obavijest o greški i prijenos se ne pokreće.
3. Klikom na gumb za pokretanje prijenosa na uređaj sa zadovoljenim prethodno navedenim stanjima, prijenos se pravilno izvodi na uređaju i u aplikaciji, a gumbi za upravljanje uređajem su onemogućeni.

Testiranjem prijenosa snimljenih podataka na uređaj sva su stanja zadovoljena i utvrđeno je da se aplikacija izvodi prema očekivanjima.

4.1. Moguća poboljšanja programskog rješenja

Nakon realizacije svih funkcionalnosti aplikacije utvrđeno je da aplikacija obavlja sve tražene zadatke i da podržava međudjelovanje s programskom podrškom upravljačkog programa i *AMV Grabber* uređaja.

Sva moguća poboljšanja funkcionalnosti korisničke aplikacije ovise o poboljšanjima na strani upravljačkog programa i uređaja jer je potrebno uskladiti sve tri strane. Glavno poboljšanje bi moglo biti vezano za prijenos podataka s računala na uređaj, gdje bi se korisniku pružala mogućnost bolje kontrole cijelog procesa iz aplikacije. Realizirano rješenje predstavlja prvu verziju sustava.

5. ZAKLJUČAK

Cilj ovog rada bio je razvoj korisničke aplikacije za *AMV Grabber* uređaj koji služi za snimanje i reprodukciju podataka koji simuliraju realno okruženje vozila u zatvorenom prostoru. U ovom radu predstavljeni su *AMV Grabber* i modeli kamera koje su se koristili tijekom razvoja i testiranja korisničke aplikacije. Za razvoj aplikacije korišten je C++ programski jezik unutar *Qt* programskog okvira.

U radu su predstavljene realizirane funkcionalnosti aplikacije i način njihova korištenja s uređajem. Predstavljena su dva načina snimanja, gdje se podaci šalju s kamera na računalo, te je predstavljen prijenos podataka s računala na uređaj. Predstavljene su dodatne funkcionalnosti, koje je moguće izvoditi bez uređaja. Moguće je vršiti pretvorbu snimljenog sadržaja u slikovne okvire ili u video zapis kojeg je moguće reproducirati unutar aplikacije.

Aplikacija je dio sustava koji se još sastoji od *AMV Grabber* uređaja i upravljačkog programa koji su razvijani zajedno s korisničkom aplikacijom. Aplikacija opisana u ovom radu dio je prve verzije cijelog sustava, u kojem su realizirani svi početni zahtjevi. U radu su opisana moguća poboljšanja aplikacije i uvjeti za njihovu realizaciju.

LITERATURA

[1] Verband der Automobilindustrie e.V, Automation - From Driver Assistance Systems to Automated Driving https://www.vda.de/dam/vda/publications/2015/automation.pdf [13. 09. 2018.]

[2] I. Kohli, Who Needs Frame Grabbers, Anyway? https://www.qualitymag.com/articles/91008-who-needs-frame-grabbers-anyway [13. 09. 2018.]

[3] I. Kohli, Vision & Sensors: Frame Grabbers and Imaging Boards: Machine Vision Still Needs Frame Grabbers, https://www.qualitymag.com/articles/90625-vision-sensors-frame-grabbers-and-imaging-boards-machine-vision-still-needs-frame-grabbers [13. 09. 2018.]

[4] Qt Creator Manual, http://doc.qt.io/qtcreator/ [13. 09. 2018.]

[5] The Meta-Object System, http://doc.qt.io/qt-5/metaobjects.html [13. 09. 2018.]

[6] Signals & Slots, http://doc.qt.io/qt-5/signalsandslots.html [13. 09. 2018.]

[7] qmake Manual, http://doc.qt.io/qt-5/qmake-manual.html [13. 09. 2018.]

[8] DMA for PCI Express (PCIe) Subsystem, https://www.xilinx.com/products/intellectual-property/pcie-dma.html#overview [13. 09. 2018.]

[9] M. Posch, How To Really, Truly Use QThreads; The Full Explanation, https://mayaposch.wordpress.com/2011/11/01/how-to-really-truly-use-qthreads-the-full-explanation/ [13. 09. 2018.]

[10] Naming Files, Paths, and Namespaces, https://docs.microsoft.com/en-us/windows/desktop/fileio/naming-a-file [13. 09. 2018.]

[11] Qt Multimedia 5.11., https://doc.qt.io/qt-5.11/qtmultimedia-index.html [13. 09. 2018.]

[12] Using Custom Widgets with Qt Designer, http://doc.qt.io/qt-5/designer-using-custom-widgets.html [13. 09. 2018.]

[13] PuTTY - a free SSH and telnet client for Windows, https://www.putty.org/ [13. 09. 2018.]

SAŽETAK

Razvoj ADAS algoritama zahtijeva stalno i temeljito testiranje i verifikaciju ispravnosti rada. Radi smanjenja troškova i povećanja obujma testiranja u sigurnim uvjetima, razvijaju se sustavi koji simuliraju realnu okolinu vozila u zatvorenim prostorima. U ovom diplomskom radu razvijena je korisnička aplikacija za upravljanje uređajem koji podržava snimanje i reprodukciju video signala s devet kamera. Aplikacija je razvijena za Microsoft Windows operacijski sustav korištenjem Qt programskog okvira. Aplikacija podržava dva načina snimanja: snimanje kamerama i snimanje ispitnog uzorka, gdje se podaci prenose s uređaja na računalo i pohranjuju u odredišnu datoteku. Također podržava prijenos tih podataka s računala natrag na uređaj. Aplikacija omogućuje pretvorbu snimljenih podataka u formate pogodne za prikaz na računalu i omogućuje reprodukciju pretvorenih video zapisa. Aplikacija je testirana s uređajem i upravljačkim programom nakon čega je zaključeno da su zadovoljeni svi zahtjevi zadatka.

Ključne riječi: ADAS, Qt, snimanje, reprodukcija, pretvorba

USER APPLICATION FOR A DEVICE FOR RECORDING AND PLAYBACK OF VIDEO CONTENT FOR VERIFICATION OF ADAS ALGORITHM

ABSTRACT

Development of ADAS algorithms requires continually and thorough testing and verification of correctness of work. To reduce costs and increase the volume of testing in safe conditions, systems that simulate real environment of vehicles in indoor spaces are developed. In this master's thesis, user application for controlling the device which supports recording and reproduction of video signal from nine cameras. Application is developed for Microsoft Windows operating system using Qt framework. The application supports two recording modes: recording with cameras and recording of test pattern, where data is transferred from device to computer and stored in destination file. It also supports transfer of this data from computer back to device. Application enables conversion of recorded data into formats suitable for display on computer and enables playback of converted videos. Application was tested with the device and the driver after which it was concluded that all the requirements of the task were met.

Key words: ADAS, Qt, recording, playback, conversion

ŽIVOTOPIS

Luka Umiljanović rođen je 21. 11. 1994. godine u Našicama. Od 2001. do 2009. godine pohađa Osnovnu školu Dore Pejačević u Našicama. Nakon završetka osnovne škole, 2009. godine upisuje prirodoslovno-matematičku gimnaziju u Srednjoj školi Isidora Kršnjavoga u Našicama, gdje kroz 4 godine uči programiranje u različitim programskim jezicima. Srednju školu završava redovito 2013. godine te u istoj upisuje sveučilišni preddiplomski studij računarstva na Elektrotehničkom fakultetu u Osijeku, danas zvanom Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek. Nakon završetka preddiplomskog studija 2016. godine, na istom fakultetu upisuje sveučilišni diplomski studij računarstva, smjer programsko inženjerstvo. U rujnu 2017. godine postaje stipendist Instituta RT-RK Osijek, gdje pod sumentorstvom izrađuje svoj diplomski rad.