

Opis i primjena FriendlyARM Mini2440 računara

Divljan Nikola

Univerzitet u Istočnom Sarajevu
Elektrotehnički fakultet
Istočno Sarajevo, Bosna i Hercegovina
nikoladivljan@yahoo.com

Sadržaj— Rad sadrži opis osnovnih karakteristika, specifikaciju i prednosti korišćenja FriendlyARM Mini2440 razvojnog sistema, te način izbora i pokretanje uređaja na različitim operativnim sistemima. Dat je detaljan opis povezivanja ploče sa okolinom upotrebom Windows platforme koristeći Visual Studio 2008 alat.

Gljučne riječi – *embedded; Windows okruženje; Visual Studio;*

I. UVOD

Danas postoji niz mikrokontrolera sa izuzetnim karakteristikama velikog broja U/I *pinova*, AD/DA konvertora i RS 232 priključkom koji omogućavaju povezivanje sa objektom upravljanja. Kod ovih uređaja, čip je već kreiran za korisnika koji treba da napiše *softver* i pokrene ga na mikrokontroleru koji ga spašava u *fleš* memoriju sve dok se ne zamijeni drugim ili ne izbriše. Na ovaj način korisnik ima kontrolu nad *softverom*. Ovo je jeftino rješenje koje po pravilu ne sadrži HMI (Human machine interface). U istu svrhu, na raspolaganju je i veliki broj FPGA (Field-programmable gate array) kola koja omogućavaju izvršavanje niza operacija i kod kojih korisnik sam osmišljava izgled kola. Ona ne sadrže procesor koji pokreće *softver* sve dok ga korisnik ne kreira. Kolo može biti jednostavno u vidu jednog "I" kola, ali i kompleksno kao višejezgarni procesor. Da bi se osmislio *dizajn*, koristi se HDL (Hardware description language). Mana FPGA kola je što čuvaju svoju konfiguraciju u RAM, a ne u *fleš* memoriji što dovodi do posljedice da jednom kada izgube napajanje, gube i svoju konfiguraciju, tako da se moraju konfigurisati svaki put kada dobiju strujno napajanje. Ovo i nije toliko loše ukoliko postoji *fleš* čip koji će automatski konfigurisati kolo prilikom uključivanja. Ovakva rješenja omogućavaju potpunu kontrolu nad *hardverom*, ali nedostatak je u tome što takođe ne sadrže HMI, kao i princip veoma zahtjevnog programiranja. Upoređujući ova dva uređaja, kod tipičnog mikrokontrolera, postoje *pinovi* koji su namjenski kreirani. Npr. ukoliko postoje samo dva *pina* na mikroprocesoru za serijske *portove*, a korisniku je potrebno više njih, jedino rješenje pored promjene čipa je korišćenje *softvera* koji će oponašati serijski *port*. To radi dobro, osim što ima za posledicu trošenje procesorskog vremena za jednostavnu operaciju slanja bitova, tako da, ako korisnik želi oponašati više *portova*, dolazi do potrošnje kompletnog procesorskog vremena. Sa FPGA kolima, korisnik samostalno kreira kolo, tako da ima mogućnost određivanja na koje *pinove*

se konektuje serijski *port*, a ukoliko je potrebno, može kreirati proizvoljan broj serijskih *portova*. Jedino ograničenje predstavlja broj U/I *pinova*, kao i veličina ploče. Bitna karakteristika FPGA kola se ogleda u tome što se može u toku kreiranja *hardvera*, kreirati i *hardver* koji će imati funkciju procesora na kome se može pokrenuti *softver*. Sagledavajući sve karakteristike, jedno od mogućih izuzetno dobrih rješenja za konkretan princip upravljanja objektima je razvojni sistem FriendlyARM koji nudi *embedded* računar malih gabarita sa mogućnošću pokretanja na različitim operativnim sistemima. Uređaj omogućava lako usklađivanje sa okolinom s obzirom na dimenzije i na taj način pruža bolje fizičke karakteristike. Ploča se karakteriše izvršavanjem samo jednog procesa u svakom trenutku, tako da se potpuno iskorištava procesorsko vrijeme i postiže brzina u izvršavanju komandi. *Emedded* računar nudi mogućnost korišćenja *displeja* sa upotrebom *touch screen-a*, te se na taj način dobija na multifunkcionalnosti uz jednostavan pristup standardnim karakteristikama specifičnim za računare opšte namjene poput USB pristupa, LAN, HDD, RS 232, VGA, MIC i SD karticama uz upotrebu korisničkih *pinova* i osjetljivog ekrana različitih dimenzija. Sa druge strane, najveća prednost ploče se ogleda u praktično veoma niskoj cijeni koja zauzvrat nudi mnoge mogućnosti uz odnos cijene i performansi na visokom nivou [1]. Bitno je napomenuti slabiju procesorsku moć FriendlyARM razvojnog sistema, pa se stoga predlažu rješenja koja podrazumijevaju kombinaciju ovog uređaja i mikrokontrolera ili FPGA kola. Veliki broj izračunavanja u realnom vremenu bi se i dalje implementirao na FPGA kolima/mikrokontrolerima dok bi korisnički *interfejs* bio implementiran na FriendlyARM računaru.

II. FRIENDLYARM MINI2440

ARM razvojne ploče su idealna platforma za ubrzanje pristupa programiranja, te olakšanog izvršavanja operacija konačnom korisniku. Kombinacija karakteristika koje nudi ASIC (An application-specific integrated circuit - integrisano kolo dizajnirano samo za određenu upotrebu) i FPGA tehnologija u kombinaciji sa ARM pločom, donose optimalno rješenje u pogledu brzine, tačnosti, fleksibilnosti i troškova. U pogledu razvoja uređaja FriendlyARM nudi niz ploča od kojih je moguće izabrati (Mini2440, Micro2440, Mini2451, Tiny2451, Tiny2416, Mini6410, Tiny6410, Mini210, Mini210s, Tiny210, Smart210, NanoPC-T1, Super4412 i Tiny4412). FriendlyARM Mini2440 predstavlja ploču čiji se

rad bazira na korišćenju Samsung S3C2440 ARM9 mikroprocesora. Dimenzije su samo 10cm x 10 cm, te omogućavaju lako prilagođenje okruženju i upotrebu u brojnim sistemima [2].

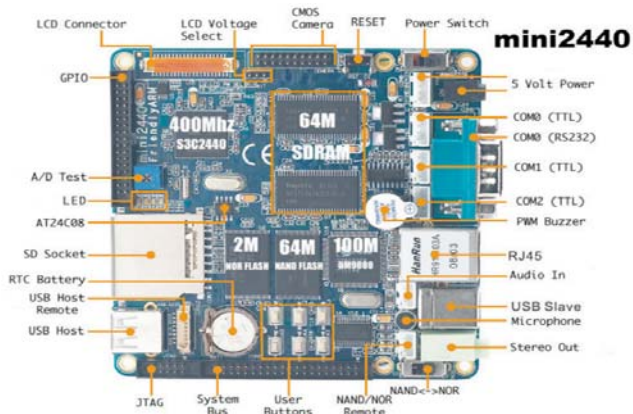
- Dimension: 100 x 100 mm
- CPU: 400 MHz Samsung S3C2440 ARM926T (max freq. 533 MHz)
- RAM: 64 MB, 32 bit Bus
- Flash: up to 1GB NAND Flash
- EEPROM: 256 Byte (I2C)
- Ext. Memory: SD-Card socket
- Serial Ports: 1x DB9 connector (RS232), total: 3x serial port connectors
- USB: 1x USB-A Host 1.1, 1x USB-B Device 1.1
- Audio Output: 3.5 mm stereo jack
- Audio Input: Connector + Condenser microphone
- Ethernet: RJ-45 10/100M (DM9000)
- RTC: Real Time Clock with battery (CR1220)
- Beeper: PWM buzzer
- Camera: 20 pin (2.0 mm) Camera interface
- LCD Interface: 41 pin connector for FriendlyARM Displays and VGA Board
- User Inputs: 6x push buttons and 1x A/D pot
- User Outputs: 4x LEDs
- Expansion: 40 pin System Bus, 34 pin GPIO, 10 pin Buttons (2.0 mm)
- Debug: 10 pin JTAG (2.0 mm)
- Power: regulated 5V (DC-Plug: 1.35mm inner x 3.5mm outer diameter)
- Power Consumption: Mini2440: 0.3 A, Mini2440 + 3.5" LCD: 0.6 A, Mini2440 + 7" LCD: 1 A
- OS Support
 - Windows CE 5 and 6
 - Linux

Slika 1. Specifikacija uređaja [3]



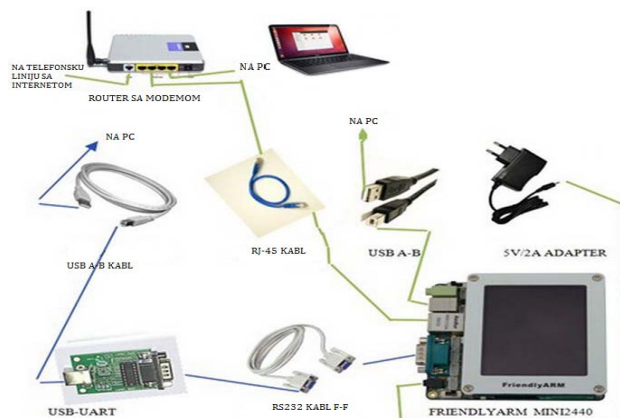
Slika 2. FriendlyARM Mini2440 [4]

Uređaj podržava Android, Windows CE 5.0 i 6.0, te Linux 2.6 operativni sistem uz opcije korišćenja displeja veličine 3.5" i 7" koji rade na 5V uz rezoluciju 1024x768 i 240x320, respektivno. Omogućeno je korišćenje 3 serijska porta od kojih je jedan i RS232, USB pristupa i SD kartice sa mogućnošću zamjene u cilju povećavanja memorije.



Slika 3. Izgled periferije ploče [5]

Prošireni *interfejs* čine GPIO (General-purpose input/output) sa 34 *pina*, te System Bus sa 40 *pinova* od po 2mm. Na periferiji ploče se nalazi 6 korisničkih dugmadi u svrhu ulaznih signala i 4 izlazna implementirana u vidu LED dioda. JTAG (Joint Test Action Group) je podržan sa 20 *pinova* i pojednostavljuje razvoj aplikacije i otklanjanje grešaka uz mogućnost izvršavanja programa korak po korak ili postavljanja tačke do koje će se program izvršiti (breakpoint). *Interfejs* za kameru je podržan sa 20 *pinova*. Najveća prednost ploče se ogleda u praktično veoma niskoj cijeni koja zauzvat nudi mnoge mogućnosti uz odnos cijene i performansi na visokom nivou.



Slika 4. Način povezivanja uređaja [6]

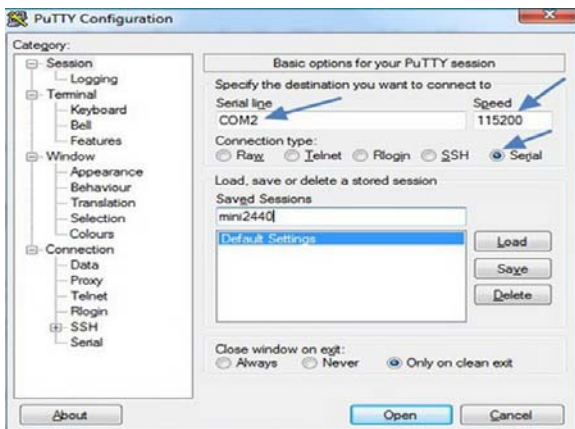
III. NAČIN IZBORA OPERATIVNOG SISTEMA

Uređaj može imati do 256MB RAM memorije, do 1GB NI i 2MB NILI *fleš* memorije uključujući BIOS. NI memorija se koristi za sve prenosive memorijske kartice zbog brzog upisa/brisanja operacija, ali uz spor slučajni pristup memoriji, dok NILI memoriju karakteriše sposobnost brzog slučajnog pristupa memoriji. Individualne karakteristike ćelija *fleš* memorije pokazuju slične karakteristike kao odgovarajuća logička kola. FriendlyARM koristi oba tipa. Uređaj sadrži SuperVivi bootloader (program koji se inicijalno pokreće pri uključivanju uređaja) na NILI *fleš* memoriji i omogućava instalaciju operativnog sistema na NI *fleš* memoriju. Koristi se kao alat za preuzimanje i instaliranje slike operativnog sistema. Povezivanjem (Slika 4.) uređaja sa PC računarom počinje proces instaliranja operativnog sistema. Potrebno je prekidač prebaciti u NILI mod rada i isključiti uređaj. Za potpunost funkcionalnosti strukture neophodna je instalacija *drajvera* za USB i USB-UART ploču, njihova detekcija u Device Manager-u, instalacija PuTTY *emulatora* i DNW WinCE alata za omogućavanje konekcije preko serijskog porta, te za nastavak procesa i uključivanje uređaja [7].

A. Instalacija Linux Qtopia platforme

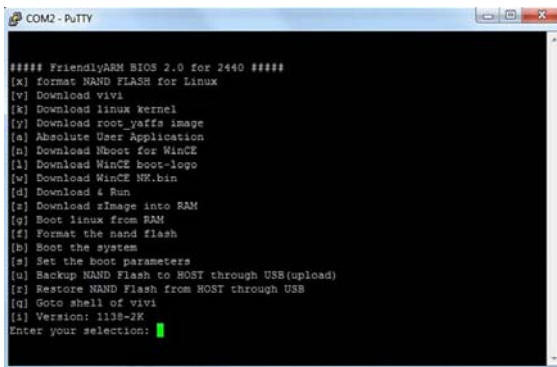
Linux Qtopia koristi rootfs_qtopia_qt4 supervivi-128M, vboot i zImage_X35 *fajlove* koje je potrebno prekopirati na

memoriju PC računara sa instalacionog DVD-a. X35 predstavlja oznaku *displeja* uređaja i ona može biti označena u zavisnosti od tipa uređaja i sa A70, L80, N35, N43, T35, W35 ili VGA sa *rezolucijom* od 1024x768 *piksela*. Pokretanjem *emulatora*, podešavaju se tip konekcije, broj *porta* i brzina koja je izražena u baudima.



Slika 5. Konfiguracija PuTTY emulatora

Spašavanjem navedenih postavki pod nazivom „mini2440“, pokreće se kreirani *fajl* i preko opcije „Load“ učitavaju podešavanja. Pritiskom na taster „Enter“ prikazuje se FriendlyARM BIOS preko PuTTY *terminala*.



Slika 6. FriendlyARM BIOS - PuTTY terminal

Pokretanjem DNW alata pristupa se konfiguraciji serijskog *porta* podešavaju istih podataka kao u PuTTY *emulatoru*.



Slika 7. DNW alat – podešavanje serijskog porta

Pozivanjem opcije „Connect“ iz „Serial Port“ *menija* uspostavlja se konekcija uz uslov da se prvo pokrene PuTTY, a zatim DNW alat. PuTTY emulator čeka na selekciju opcije koju treba izabrati. Potrebno je izvršiti niz koraka:

- a) Odabir opcije „x“ - formatiranje NI *fleš* memorije.
- b) Opcija „v“ -preuzimanje *supervivi_128m fajla*.
- c) DNW alat, opcija USB Port-->Transmit->Transmit šalje *fajl* u PC računar i otvara datu lokaciju nakon čega je potrebno pokrenuti datu *fajl*.
- d) Selekcija „k”- preuzimanje *fajla* “Linux Kernel”, izvršiti “Transmit”, te pokrenuti “zImage_X35” *fajl*.
- e) Opcija „y” – preuzimanje *yaffs root* slike koja je sistemski *fajl* za QTopia-u, pokrenuti “rootfs_qtopia_qt4” *fajl*.
- f) Isključiti uređaj, prekidač prebaciti u NI poziciju, te uključiti uređaj i Linux Qtopia platforma se pokreće.

B. Instalacija Windows platforme

Potrebno je generisati Windows CE sliku za Mini2440 koristeći Platform Builder i instalirati Visual Studio 2005 Pro, Windows Embedded CE 6.0, te R2 i R3 *emulatore* za Windows CE 6.0. BSP (Board Support Package) je neophodan za prilagođenje platforme na ploči.

Nakon njegove instalacije, *source* kod se vidi na lokaciji: C:\WINCE600\PLATFORM\Mini2440, a Platform Builder projekat je već kreiran na *hard disku* i nalazi se na na lokaciji C:\WINCE600\OSDesigns\Mini2440. Pokretanjem *.sln fajla* iz Visual Studio-a otvara se stablo projekta.



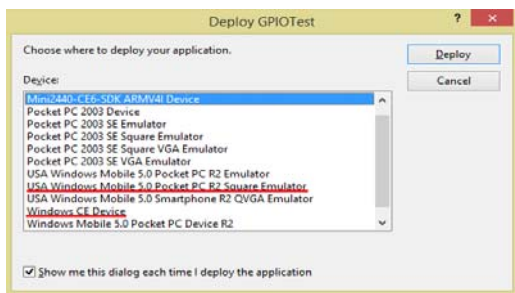
Slika 8. Prikaz instaliranog BSP-a

Za dodavanje komponente u već kreiranu sliku, potrebno je označiti Third Party/BSP/Mini2440:ARMV4I. Biblioteka options.h na lokaciji C:\wince600\platform\mini2440\Src\Inc sadrži definisanje tipa *displeja*. Potrebno je podesiti #define LCD_X35. Ponovnim pokretanjem programa, kreira se slika C:\WINCE600\OSDesigns\Mini2440\Mini2440\RelDir\Mini2440_ARMV4I_Release\nk.bin i ovaj *fajl* je neophodno prebaciti na uređaj. Za nastavak instalacije Windows platforme potrebno je izvršiti niz koraka:

- Odabir opcije „f“ - formatiranje NI *fleš* memorije.
- Opcija „v“ -download vivi.
- Na lokaciji BSP-a otvoriti „supervivi_128m” iz DNW-a.
- Selekcija „n”- download Nboot for Win CE.
- Na lokaciji BSP-a otvoriti „nboot_X35” *fajl*.
- Opcija „l“ – download Win CE boot-logo.
- Na lokaciji BSP-a otvoriti „boot-logo” *fajl*.
- Opcija „w“ – download Win CE NK.bin.
- Na lokaciji kreirane slike, otvoriti *fajl* „nk.bin“.
- Kreira se particija i pokreće Windows CE 6.0.

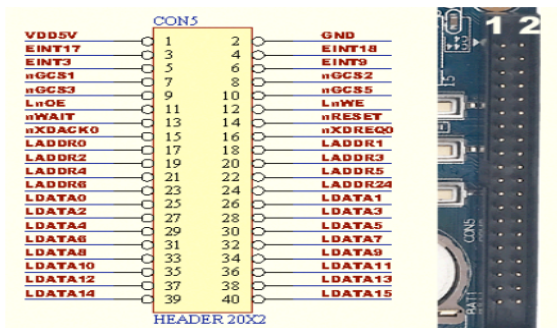
IV. POVEZIVANJE UREĐAJA SA OKOLINOM

Instalacijom Windows platforme omogućeno je programiranje kroz Visual Studio 2005/2008 uz upotrebu SDK WinCE za Mini2440. Kreirana aplikacija se pokreće na uređaju povezanom preko USB-a odabirom opcije Mini2440-CE6-SDK ARMV4I Device. Testiranje programa bez priključivanja uređaja na računar omogućava se instaliranjem *emulatora* R2 i R3 koji će vizuelno improvizovati funkcionisanje aplikacije.



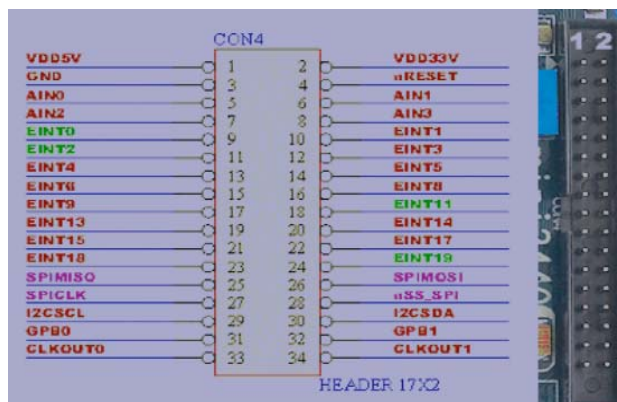
Slika 9. Odabir uređaja prilikom pokretanja aplikacije

System Bus *interfejs* (Slika 10.) obuhvata 40 *pinova* i predstavlja način prenosa informacija između *hardverskih* komponenti (uključujući CPU i RAM) i perifernih uređaja. Osnovna uloga je u proširenju mogućnosti uređaja. Ove izlaze čini 16 linija podataka, 8 adresnih linija i pojedine signalne linije (chip select, read, write...). Omogućeno napajanje za pravilno funkcionisanje je 5V.



Slika 10. Prikaz i naziv System Bus *pinova* [8]

GPIO *interfejs* (Slika 11.) obuhvata 34 *pina* koje čine AIN0-AIN3 za analogno/digitalni test, CLKOUT, SPI *interfejs*, I²C *interfejs*, GPB0 za zvučni signal, GPB1, te EINT0-EINT19 kao ulazni/izlazni signali. JTAG obuhvata 10 *pinova* uz generalnu upotrebu otklanjanja grešaka i korišćenja BIOS-a.



Slika 11. Prikaz i naziv GPIO *pinova* [8]

A. Definisane klase i funkcija

Upotrebom odgovarajućih biblioteka pristupa se iz Visual Studio-a *pinovima* na ploči. U zavisnosti da li se koristi C, C# ili neki drugi programski jezik koriste se definisane klase. U biblioteci za C# svaki *pin* na ploči je definisan sa oznakom *porta* kojem pripada, konfiguracijom, brojnomo oznakom, te vrijednosti (logička jedinica ili logička nula). Slanje i primanje signala se omogućava podešavanjem vrijednosti za dati *pin*.

a) Port

Pinovi na ploči su grupisani po *portovima*. Konkretno, *pinovi* na GPIO označeni sa EINT0, EINT1, EINT2, EINT3, EINT4, EINT5 i EINT6 pripadaju *portu* F, dok je EINT9 *port* G. Svi *pinovi* su grupisani u 10 *portova* (A-J). Klasa “Port” sadrži definisane sve *portove*. *Source* kod klase “Port” se nalazi u dodatku.

b) Konfiguracija *pina*

Prilikom definisanja naznačava se svrha korišćenja (ulaz/izlaz). Potrebno slanje signala na izlaz omogućava “output”, a ulaz “input”. Prilikom definisanja ulaznog signala bira se opcija “pullup” gdje se otpornik koristi za podešavanje ulaza sistema na određen logički nivo ili “pulldown” koji je povezan na masu i drži signal blizu nule. *Source* kod klase “PinConfiguration” se nalazi u dodatku.

c) Brojna oznaka izlaza

Svaki *pin* je definisan oznakom tipa integer. Brojevi za *pinove* na GPIO *interfejsu* (Slika 13.) se nalaze nakon oznake za *port*.

d) Logička vrijednost pina

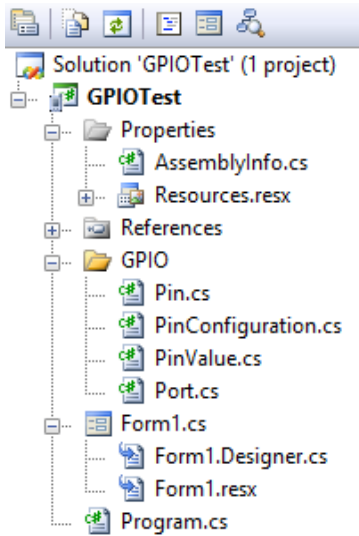
Kreiranje *pina* se završava definisanjem njegove početne vrijednosti prilikom startovanja aplikacije. Vrijednost može biti 0 ili 1, tj. LOW ili HIGH. Sve klase i funkcije su definisane u *namespace*-u koji je nazvan "Mini2440.GPIO". *Source* kod klase "PinValue" se nalazi u dodatku.

B. Kreiranje novog projekta

Opcijama (File-->New-->Project) u Visual Studio-u prikazuje se prozor koji omogućava izbor tipa projekta i programskog jezika. Odabirom C# programskog jezika, te "Windows Forms Application" kao *template*-a, uspješno je kreiran projekat.

C. Rad sa Windows formama

Nakon kreiranja novog projekta prikazuje se radna površina. Desnim klikom na bilo koji element sa radne površine (uključujući i nju) i odabirom opcije Properties omogućavaju se dodatna podešavanja. Sa lijeve strane se nalazi alat (Toolbox), tj. elementi koji se koriste na radnoj površini. Omogućavanje korišćenja GPIO biblioteke se izvršava jednostavnim ubacivanjem foldera „GPIO“ u strukturno stablo aplikacije i uključivanjem namespace-a Mini2440.GPIO u kôdu aplikacije preko komande „using“.



Slika 12. Struktura projekta

D. Definisane pinova

Nakon uključivanja potrebnih biblioteka definisanje *pina* predstavlja kreiranje promjenljive tipa klase Pin. U funkciji *Form_load* (pokreće se odmah prilikom startovanja aplikacije) potrebno je kreirati novi objekat tipa "Pin" kojem se definišu vrijednosti *port*-a, konfiguracija, broj i vrijednost ulaza/izlaza. Kreiranjem dugmeta i metode *button_Click()* kod koji se nalazi u njoj će se izvršiti nakon klika na dugme. U

namespace-u Mini2440.GPIO je kreirana funkcija *ToogleValue()* (engl. Toogle-žabica) koja provjerava stanje na *pinu* (0 ili 1) i podešava suprotnu vrijednost. Na ovaj način je *pin* klikom na dugme promijenio vrijednost. Isti pristup je ka svim *pinovima*, s tim da se samo mijenja *port* i broj izlaza. *Source* kod primjera realizacije slanja signala na *pinove* prikazan je u dodatku.

Rb.	Izlaz	Port Broj	Rb.	Izlaz	Port Broj
1	VDD5V 5V		18	EINT11	GPG3
2	VDD33V		19	EINT13	GPG5
3	GND		20	EINT14	GPG6
4	nRESET		21	EINT15	GPG7
5	AIN0 A/D		22	EINT17	GPG9
6	AIN1 A/D		23	EINT18	GPG10
7	AIN2		24	EINT19	GPG11
8	AIN3		25	SPIMISO	GPE11
9	EINT0	GPF0	26	SPIMOSI	GPG6
10	EINT1	GPF1	27	SPICLK	GPE13
11	EINT2	GPF2	28	nSS_SPI	GPG2
12	EINT3	GPF3	29	I2CSCL	GPE14
13	EINT4	GPF4	30	I2CSDA	GPE15
14	EINT5	GPF5	31	GPB0	TOUT0
15	EINT6	GPF6	32	GPB1	TOUT1
16	EINT8	GPG0	33	CLKOUT0	GPH9
17	EINT9	GPG1	34	CLKOUT1	GPH10

Slika 13. GPIO interfejs

Rb.	Kamera	Port Broj	Rb.	Kamera	Port Broj
1	I2CSDA	GPE15	2	I2CSCL	GPE14
3	EINT20	GPG12	4	CAMRST	GP12
5	CAMCLK	GP11	6	CAM_HREF	GP10
7	CAM_VSYNC	GP9	8	CAM_PCLK	GP8
9	CAMDATA7	GP7	10	CAMDATA6	GP6
11	CAMDATA5	GP5	12	CAMDATA4	GP4
13	CAMDATA3	GP3	14	CAMDATA2	GP2
15	CAMDATA1	GP1	16	CAMDATA0	GP0
17	VDD33V 3.3V		18	VDD_CAM	
19	VDD18V 1.8V		20	GND	

Slika 14. Kamera-interfejs

Korisnički tasteri šalju signale na *pinove* EINT8, EINT11, EINT13, EINT14, EINT15 i EINT19.

V. ZAKLJUČAK

FriendlyARM Mini2440 je idealan u kombinaciji sa FPGA kolima i naprednim mikrokontrolerima gdje su ova kola zadužena za brza izračunavanja u višestrukim i zahtjevnim procesima. Kombinacija omogućava korišćenje uređaja kao glavnog nadzornog računara koji izvršava proces upravljanja preko *displeja* praktično čineći vizualizaciju procesa koji krajnjem korisniku omogućava jednostavno upravljanje zahtjevnim operacijama izborom opcija iz *menija* na *touch displeju*. Ističe se karakteristika povezivanja sa računarom opšte namjene sa mogućnošću direktnog uređivanja programskog koda, te pokretanja i testiranja aplikacije na ploči. Napredni cilj istraživanja je povezivanje ovog računara sa nekim procesom što bi faktički predstavljalo izradu *scada* sistema.

LITERATURA

- [1] <https://embeddedmicro.com/tutorials/mojo/what-is-an-fpga> , 2015
- [2] <http://www.friendlyarm.net/> ,2015
- [3] <http://www.friendlyarm.net/products/mini2440>
- [4] <http://www.friendlyarm.net/sites/products/mini2440-35.jpg>
- [5] http://www.friendlyarm.net/sites/products/mini2440_3.jpg
- [6] https://alselectro.files.wordpress.com/2013/01/mini_setup.jpg ,2015
- [7] <https://alselectro.wordpress.com/2012/12/08/FRIENDLY-ARMMINI-2440-INSTALLING-LINUX-QTOPIA/>
- [8] http://www.friendlyarm.net/dl.php?file=mini2440_manual_20100609.pdf

a) *Klasa Port*

```
using System;
using System.Linq;
using System.Collections.Generic;
using System.Text; //uključene biblioteke
```

```
namespace Mini2440.GPIO
{
    public enum Port : int
    {
        PORT_A = 0,
        PORT_B = 1,
        PORT_C = 2,
        PORT_D = 3,
        PORT_E = 4,
        PORT_F = 5,
        PORT_G = 6,
        PORT_H = 7,
        PORT_I = 8,
        PORT_J = 9
    }
}
```

b) *Klasa PinConfiguration*

```
//uključene biblioteke
```

```
namespace Mini2440.GPIO
{
    public enum PinConfiguration : int
    {
        OUTPUT = 0,
        INPUT_WITH_PULLUP = 1,
        INPUT_WITHOUT_PULLUP = 2
    }
}
```

c) *Klasa PinValue*

```
//uključene biblioteke
```

```
namespace Mini2440.GPIO
{
    public enum PinValue : int
    {
        LOW = 0,
        HIGH = 1
    }
}
```

```
//uključene biblioteke
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Windows.Forms;
using System.Runtime.InteropServices;
using Mini2440.GPIO; //←uključivanje biblioteke za uređaj
```

```
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        private Pin izlaz; //definisanje promjenljive

        public Form1()
        {
            InitializeComponent();
        }
        private void Form1_Load(object sender, EventArgs e)
        {
            izlaz = new Pin(Port.PORT_F, 0,
                PinConfiguration.OUTPUT, PinValue.LOW);
        }
        private void button1_Click(object sender, EventArgs e)
        {
            if (izlaz.ToggleValue() == PinValue.LOW)
            {
                button1.Text = "Uključi";
            }
            else
            {
                button1.Text = "Isključi";
            }
        }
    }
}
```

ABSTRACT

The paper contains a description of the main features, specifications and advantages of FriendlyARM Mini2440 development board, the selection and intended use of the device on different operating systems. There is a detailed description of the ways of connecting the system with the environment using a Visual Studio 2008 on Windows platform.

Description and usage of FriendlyARM Mini2440

Divljan Nikola