

Projekt DES 2011

Razvoj vgrajenega sistema

Končne naloge za skupine

1. Arhitektura

- ▶ dokončati shemo vezja in pripraviti opis

2. Prototip vezja

- ▶ izračun uporov za matriko, spajkanje, preučiti JTAG

3. VHDL vmesniki

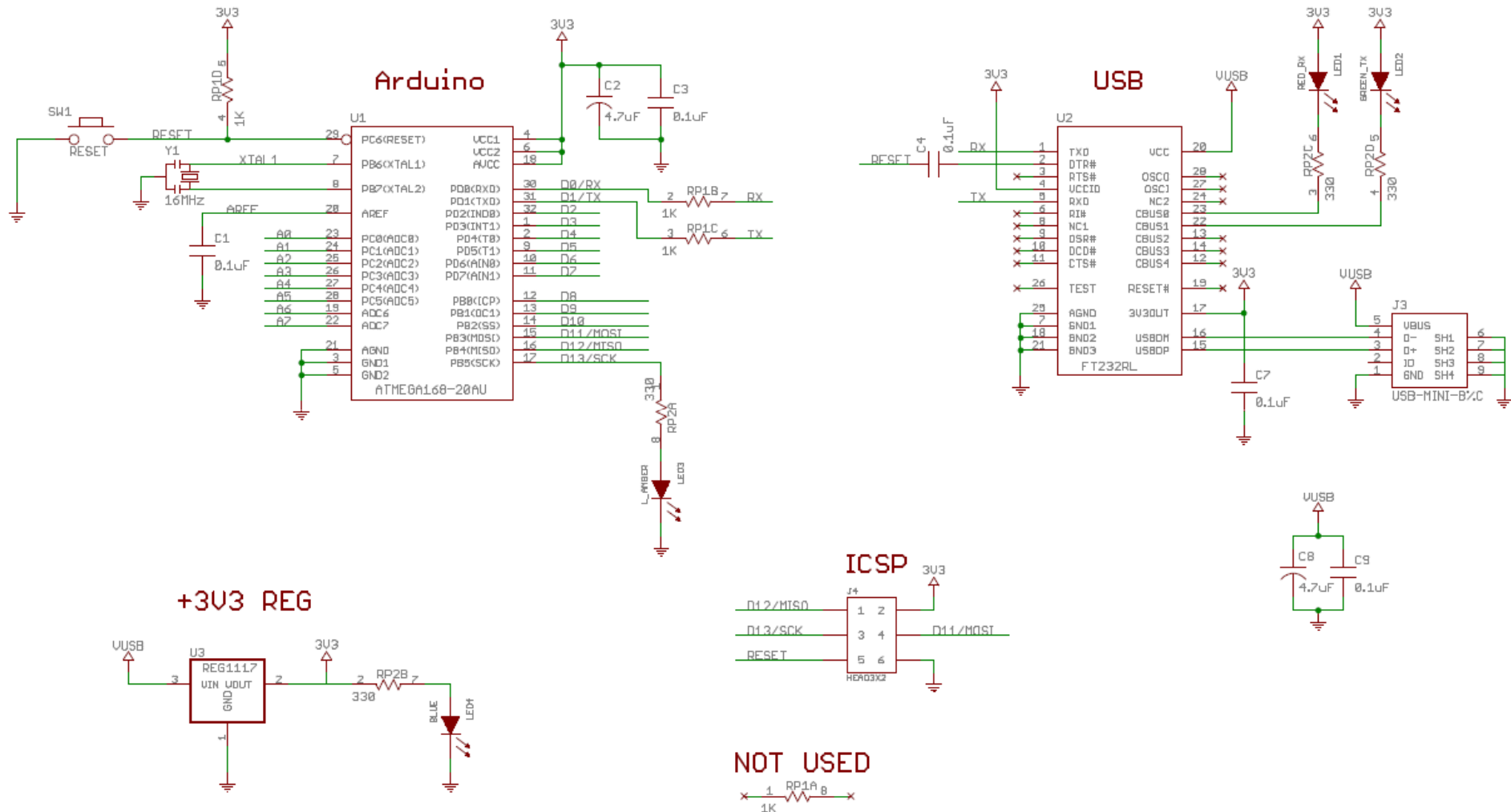
- ▶ izvedba 4-bitnega vmesnika za matriko, prenos podatkov iz PS/2 in tipk

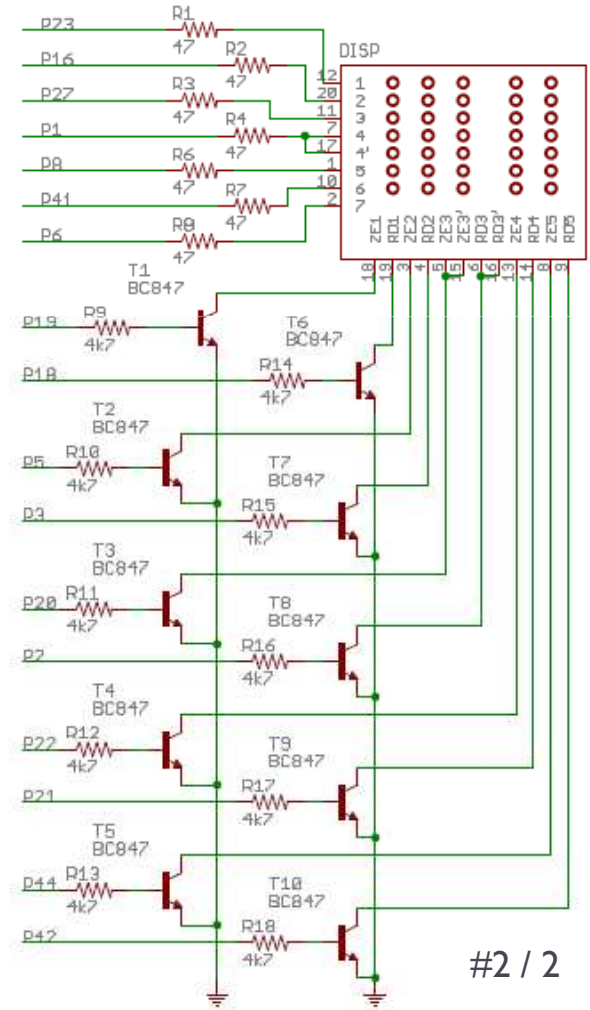
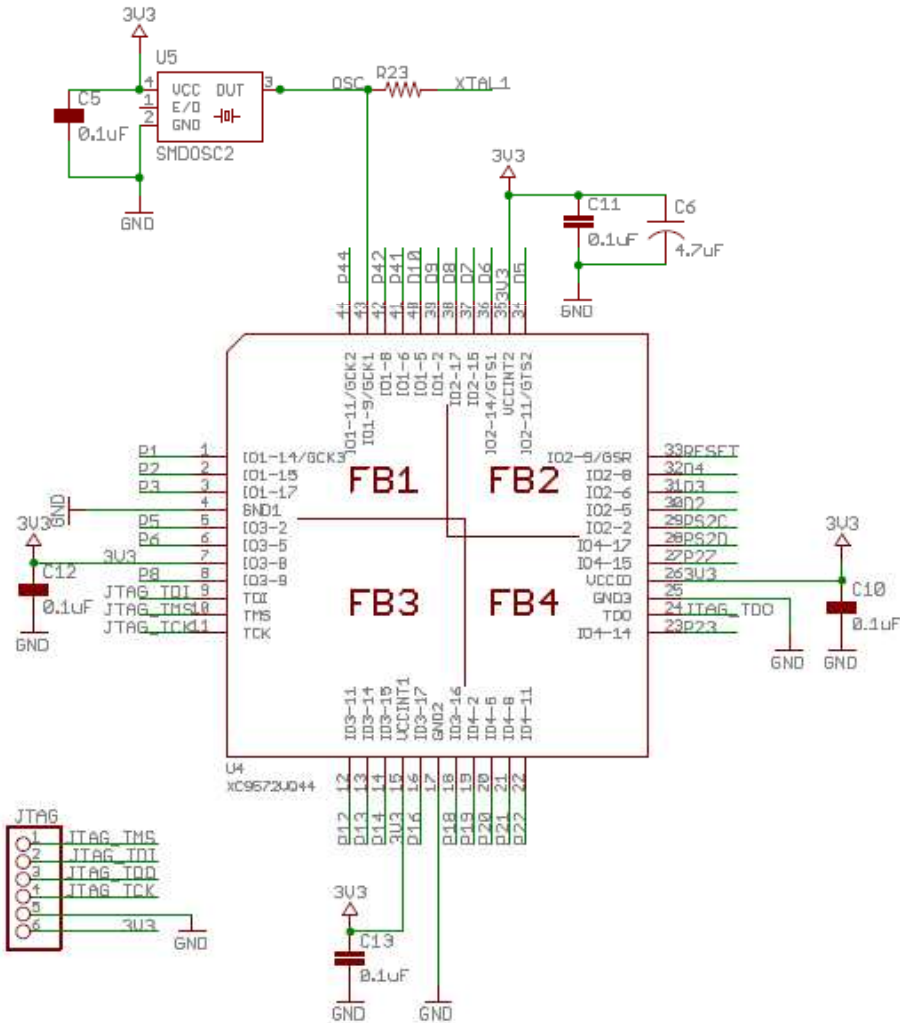
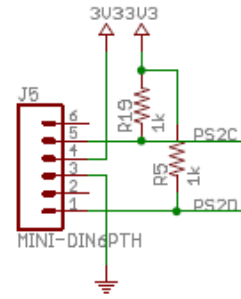
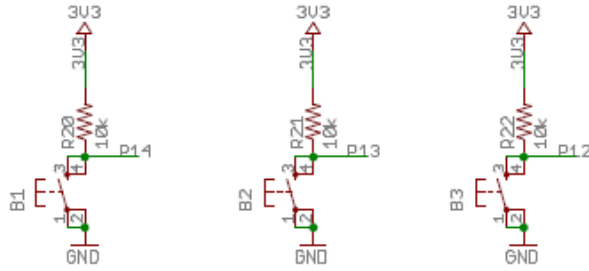
4. Programiranje Arduino

- ▶ koda za periodično osveževanje matrike, branje vhodov (PS/2 ali tipk) ter preprosta aplikacija

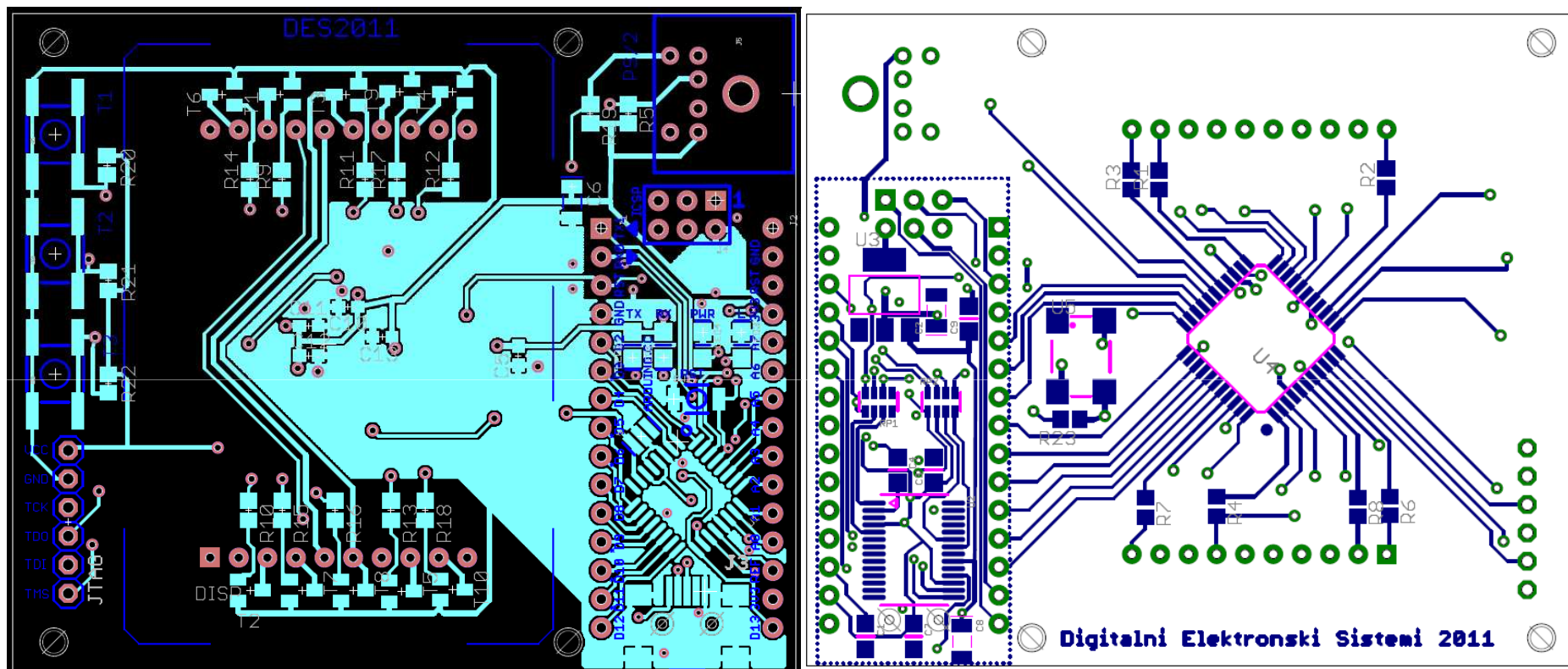


Arduino Nano (predelan na 3.3V)





➤ Tiskano vezje – zgornja in spodnja stran



Pogled na tiskano vezje

➤ Seznam elementov vezja

Part	Value	Device	Package	Description
B1		TAC_SWITCH	TACTILE_SWITCH_SMD-3	tipka SMD
B2		TAC_SWITCH	TACTILE_SWITCH_SMD-3	tipka SMD
B3		TAC_SWITCH	TACTILE_SWITCH_SMD-3	tipka SMD
C1	0.1uF	CAP0805-NP	CAP0805-NP	Kond, SMD 0805
C2	4.7uF	CAP-TANA-P	TAN-A	Kond, tantal, SMD
C3	0.1uF	CAP0805-NP	CAP0805-NP	Kond, SMD 0805
C4	0.1uF	CAP0805-NP	CAP0805-NP	Kond, SMD 0805
C5	0.1uF	CAPC603	C603	Kond, SMD 0603
C6	4.7uF	CAP-TANA-P	TAN-A	Kond, tantal, SMD
C7	0.1uF	CAP0805-NP	CAP0805-NP	Kond, SMD 0805
C8	4.7uF	CAP-TANA-P	TAN-A	Kond, tantal, SMD
C9	0.1uF	CAP0805-NP	CAP0805-NP	Kond, SMD 0805
C10	0.1uF	CAPC603	C603	Kond, SMD 0603
C11	0.1uF	CAPC603	C603	Kond, SMD 0603
C12	0.1uF	CAPC603	C603	Kond, SMD 0603
C13	0.1uF	CAPC603	C603	Kond, SMD 0603
DISP	5X7DISP	5X7DISP	5X7	LED matrika

J3	USB-MINI-B%C	USB-MINI-B%C	USB-MINI-B_2	USB konektor
J4	HEAD3X2	HEAD3X2	HEAD3X2	letvica 3 PIN x 2
J5	MINI-DIN6PTH	MINI-DIN6PTH	MINI-DIN6	mini DIN 6
JTAG		PINH-D-1X6	1X06	letvica 6 PIN
LED1	RED_RX	LED0805	LED0805	rdeča
LED2	GREEN_TX	LED0805	LED0805	zelena
LED3	L_AMBER	LED0805	LED0805	rumena
LED4	BLUE	LED0805	LED0805	zelena
R1	47	R-US_R0805	R0805	Upor, SMD 0805
R2	47	R-US_R0805	R0805	Upor, SMD 0805
R3	47	R-US_R0805	R0805	Upor, SMD 0805
R4	47	R-US_R0805	R0805	Upor, SMD 0805
R5	1k	R-US_R0805	R0805	Upor, SMD 0805
R6	47	R-US_R0805	R0805	Upor, SMD 0805
R7	47	R-US_R0805	R0805	Upor, SMD 0805
R8	47	R-US_R0805	R0805	Upor, SMD 0805
R9	4k7	R-US_R0805	R0805	Upor, SMD 0805
R10	4k7	R-US_R0805	R0805	Upor, SMD 0805
R11	4k7	R-US_R0805	R0805	Upor, SMD 0805
R12	4k7	R-US_R0805	R0805	Upor, SMD 0805
R13	4k7	R-US_R0805	R0805	Upor, SMD 0805
R14	4k7	R-US_R0805	R0805	Upor, SMD 0805
R15	4k7	R-US_R0805	R0805	Upor, SMD 0805
R16	4k7	R-US_R0805	R0805	Upor, SMD 0805
R17	4k7	R-US_R0805	R0805	Upor, SMD 0805
R18	4k7	R-US_R0805	R0805	Upor, SMD 0805
R19	1k	R-US_R0805	R0805	Upor, SMD 0805
R20	10k	R-US_R0805	R0805	Upor, SMD 0805
R21	10k	R-US_R0805	R0805	Upor, SMD 0805
R22	10k	R-US_R0805	R0805	Upor, SMD 0805
R23		R-US_R0805	R0805	Upor, SMD 0805

RPI	IK	RES4NT	RES4NT	Upor Array 4x
RP2	330	RES4NT	RES4NT	Upor Array 4x
SWI	RESET	PBI57	I57SW	Mikro tipka SMD
T1	BC847	BC847	SOT23	NPNTTRANSISTOR
T2	BC847	BC847	SOT23	NPNTTRANSISTOR
T3	BC847	BC847	SOT23	NPNTTRANSISTOR
T4	BC847	BC847	SOT23	NPNTTRANSISTOR
T5	BC847	BC847	SOT23	NPNTTRANSISTOR
T6	BC847	BC847	SOT23	NPNTTRANSISTOR
T7	BC847	BC847	SOT23	NPNTTRANSISTOR
T8	BC847	BC847	SOT23	NPNTTRANSISTOR
T9	BC847	BC847	SOT23	NPNTTRANSISTOR
T10	BC847	BC847	SOT23	NPNTTRANSISTOR
U1	ATMEGA328-20AU	ATMEGA328-20AU	QFP032	AVR
U2	FT232RL	FT232RL	SSOP28	USB vmesnik
U3	REG1117	REG1117	SOT223	3.3V 800mA LDO Regulator
U4	XC9572VQ44	XC9572VQ44	VQ44	CPLD
U5	SMDOSC2	SMDOSC2	DXO-57	Oscilator 8MHz SMD
Y1	16MHz	RESONATORMU	RESONATOR	se ne montira

Resonator se ne montira, ker je vir ure skupen z vezjem CPLD (SMDOSC preko R23)



Arhitektura

- ▶ Predstavi glavne komponente digitalnega sistema
 - ▶ shema vezja
 - ▶ spremembe v sistemu Arduino Nano (napajanje)
 - ▶ vhodni signali (tipke, PS/2)
 - ▶ vezava LED matrice
- ▶ Statični red (logični napetostni nivoji) in dinamični red (ura za AVR in CPLD)
 - ▶ Kje še potrebujemo sinhronizacijo z uro?
- ▶ Napajanje vezja in predvidena poraba moči



Prototip vezja

- ▶ Predstavi testiranje matrice na CPLD razv. sistemu
- ▶ Izračun uporov za krmiljenje stolpcev
 - ▶ napajanje IOH pri toku $\sim 15\text{mA}$, upoštevaj padec napetosti na LED pri 15mA in na transistorju v nasičenju (V_{CESat})
- ▶ Potek spajkanja vezja
- ▶ Programiranje: SPI, JTAG

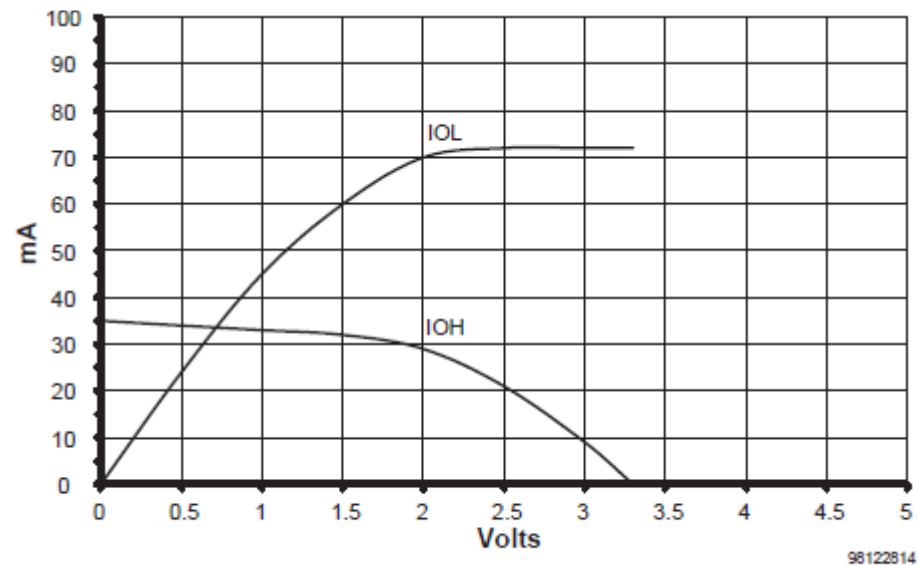


Figure 16: XC9500XL

Programiranje Arduino

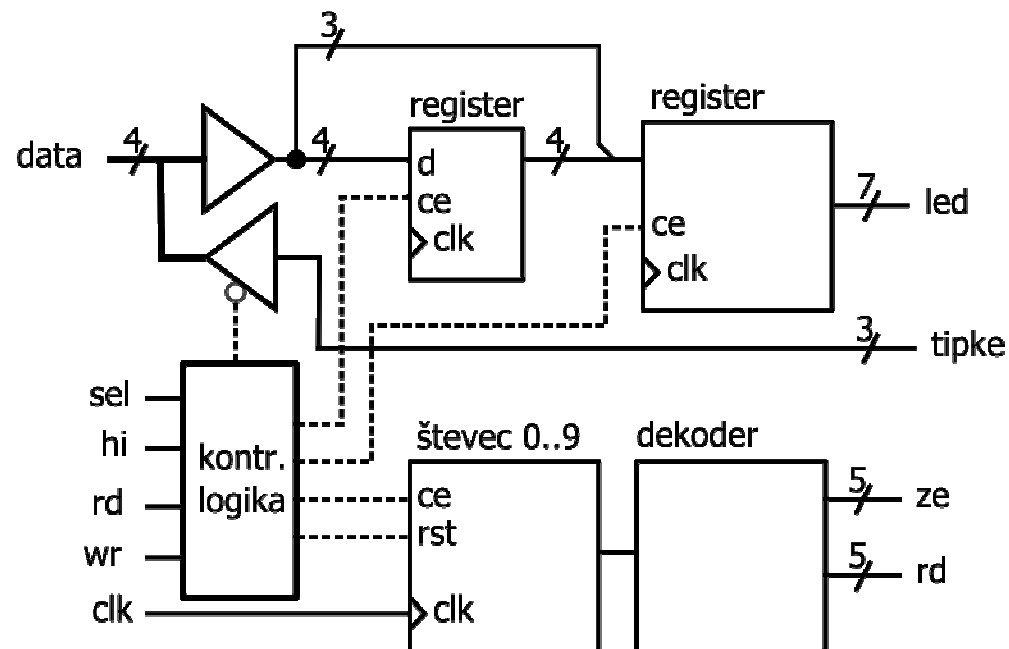
- ▶ Procesor naj 50x v sekundi osveži LED matriko
 - ▶ za celo matriko se 10x izvede funkcija, ki prenese en stolpec
 - ▶ 2 ms intervali
- ▶ Knjižnica MsTimer2
 - ▶ ::set nastavi periodo
 - ▶ ::start požene časovnik
- ▶ Prenos podatkov
DDRD=0xF0;
PORTD=d; PORTB=0x01...
- ▶ Branje tipk (na 2 ms)
 - ▶ vključi v funkcijo za osveževanje

```
FlashLed   
#include <MsTimer2.h>  
  
// Switch on LED on pin 13 each second  
  
void flash() {  
    static boolean output = HIGH;  
  
    digitalWrite(13, output);  
    output = !output;  
}  
  
void setup() {  
    pinMode(13, OUTPUT);  
  
    MsTimer2::set(500, flash); // 500ms period  
    MsTimer2::start();  
}  
  
void loop() {  
}
```



Vmesnik v CPLD

- ▶ Blokovna shema vmesnika za LED matriko in tipke
 - ▶ predstavi vezje, simulacijo, poročilo o zasedenosti
 - ▶ Kako bi v vezje vključili podatke iz PS/2 vmesnika ?



Rezultati – arhitektura

- ▶ **Vezje se razlikuje od Arduino Nano**
 - ▶ napajanje 3V3, izvor takta je zunanji oscilator 8MHz
 - ▶ preko ISP potrebno nastaviti varovalke (fuse) za izbiro frekvence in izvora ure
 - ▶ preko ISP naloži nalagalnik (Bootloader) za AVR 328P_8MHz
- ▶ **Napajanje iz USB z dodatnim regulatorjem**
 - ▶ prekini izvor napajanja iz USB – RS232 vmesnika
 - ▶ preko USB nastavi konfiguracijo vmesnika za večji tok (500mA)
- ▶ **Menjava signalnih povezav**
 - ▶ PB3 ni povezan na CPLD, uporabi PD3



Rezultati – prototip vezja

- ▶ V izdelavi dve prototipni tiskani vezji
- ▶ Ročno spajkanje SMD elementov
 - ▶ polariteta elementov (tantalovi kond., LED, oscilator, IC)
 - ▶ kratek stik / slab stik na mini USB konektorju
 - ▶ preveč spajke lahko povzroči hladen (prekinjen) spoj
- ▶ Postopek programiranja (Firmware)
 - ▶ CPLD preko USB - JTAG programatorja (Xilinx ISE)
 - ▶ nastavitve parametrov USB vmesnika (gonilnik in progr. na PC)
 - ▶ AVR varovalke in nalagalnik preko USB - ISP (AVR Studio)



Rezultati – CPLD vmesnik

- ▶ Implementacija vmesnika s števcem (integer)

Macrocells Used	Pterms Used	Registers Used	Pins Used
59/72 (82%)	111/360 (31%)	45/72 (63%)	29/34 (86%)

- ▶ Optimizacija števca (integer range 0 to 10)

Macrocells Used	Pterms Used	Registers Used	Pins Used
31/72 (44%)	53/360 (15%)	17/72 (24%)	29/34 (86%)

- ▶ Vmesnik s pomikalnim registrom za izbiro stolpcev

Macrocells Used	Pterms Used	Registers Used	Pins Used
27/72 (38%)	60/360 (17%)	23/72 (32%)	29/34 (86%)



Rezultati – aplikacija

- ▶ Program sestavljajo vsaj 3 funkcije
 - ▶ `setup()` – inicializacija vrat, časovnika, spremenljivk
 - ▶ `prenos()` – prenos 4-bitnih podatkov za LED matriko in branje stanja tipk v globalno spremenljivko
 - ▶ `loop()` – glavna zanka v kateri teče aplikacija
- ▶ Komunikacija preko vrat (**PORTD – PD, PORTB – PB**)
 - ▶ signal sel je na PD 3 (ne na PB 3)
 - ▶ z maskiranjem nastavi le signale, ki jih je potrebno, ostali naj ostanejo v zadnjem stanju: `DDRD = DDRD | B11111000;`
 - ▶ pred branjem tipk je potrebno spremeniti DDRD in vpisati 0 na zgornje 4 bite PORTD, sicer ne beremo prave vrednosti...



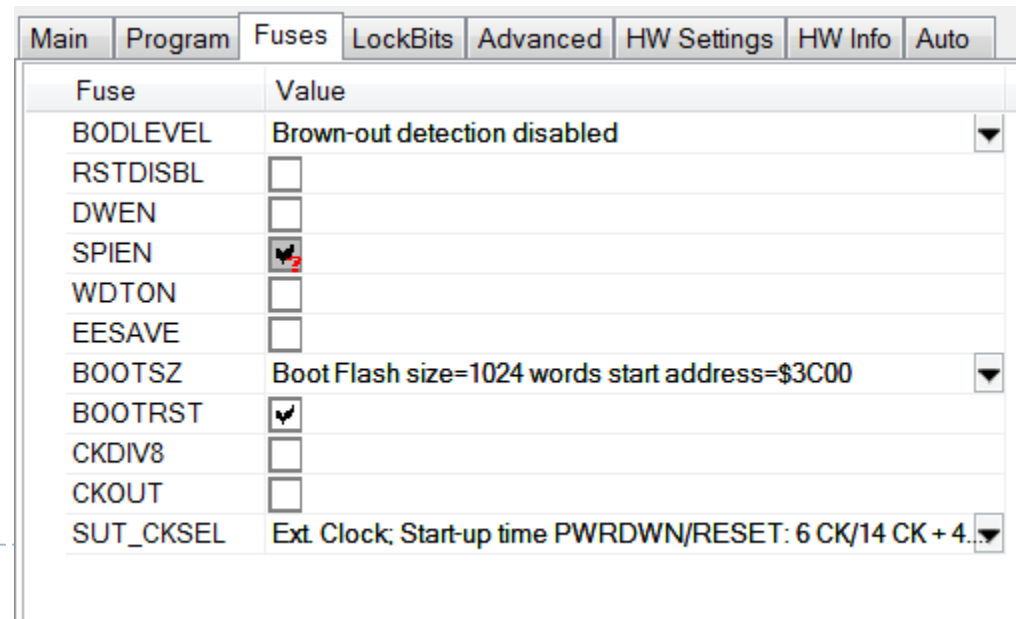
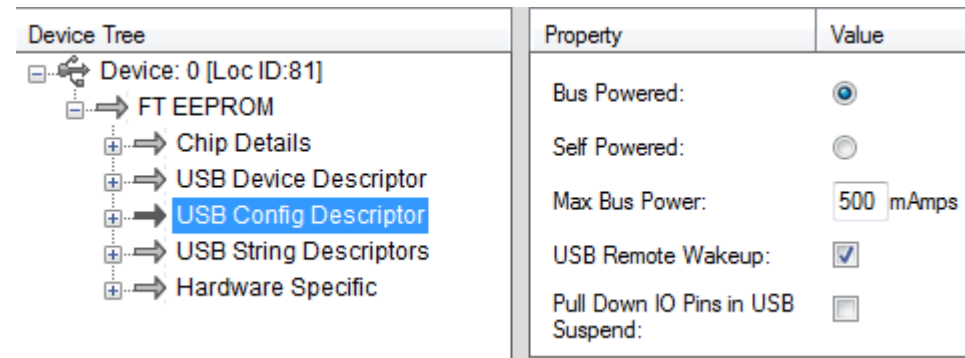
Možnosti nadgradnje arhitekture

- ▶ Prosti so še vsi analogni vhodi procesorja
 - ▶ senzorji, npr. temperatura, pospeškomer, ...
- ▶ Baterijsko napajanje
 - ▶ DC-DC stikalni pretvornik, npr. iz 2.4-3V na 3.3V
- ▶ Zmogljivejši AVR bi lahko sam krmilil 5x7 matriko (brez vmesnika CPLD)
 - ▶ za večje matrike bi bil vseeno potreben vmesnik
- ▶ Dodatni komunikacijski vmesniki
 - ▶ npr. brezžični moduli na RS232



Programska oprema in programatorji

- ▶ USB vmesnik: FTDI CDM gonilnik, FT_PROG.exe
- ▶ Xilinx ISE in JTAG USB programator
- ▶ Arduino
 - ▶ Bootloader *.hex
 - ▶ prog. razv. okolje
- ▶ AVR Studio in AVR ISP mkII programator



Optimizacija vezja

- ▶ Za osnovni vmesnik s tipkami zadošča 50% manjše programirljivo vezje XC9536
 - ▶ zasedenost 38% makrocelic
 - ▶ cena v povprečju 30% nižja
- ▶ Zadoščal bi mikrokrmilnik s 16kB Flash namesto 32kB
 - ▶ zasedenost demo programa 1.5kB / 30kB
 - ▶ cena v povprečju 30% nižja
- ▶ USB vmesnik je potreben le za razvoj, lahko bi ga spustili v produkcijski verziji
 - ▶ nalaganje programa preko ISP



Delo v razvojnem okolju Arduino

▶ Verify / Compile

- ▶ transformacija kode, vključi definicije konstant za izbran mikrokrmilnik, vključi vstopno funkcijo main()
- ▶ prevajanje C kode z AVR GCC prevajalnikom

▶ Prednosti razvojnega okolja

- ▶ enostavno delo, podpira pravilno strukturo programa (funkciji setup() in loop())
- ▶ napisanih precej knjižnic, ki podpirajo različne periferne enote in zunanja vezja (senzorji, prikazovalniki...)
- ▶ Obsežen forum z vprašanji in odgovori

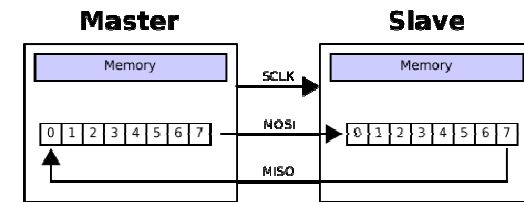
▶ Osnovni program enostavno nadgradimo

- ▶ potek aplikacije zapišemo v loop(), za občasne dogodke (npr. podatki iz PS/2) pa uporabimo prekinitve



Nalaganje programa in debugiranje

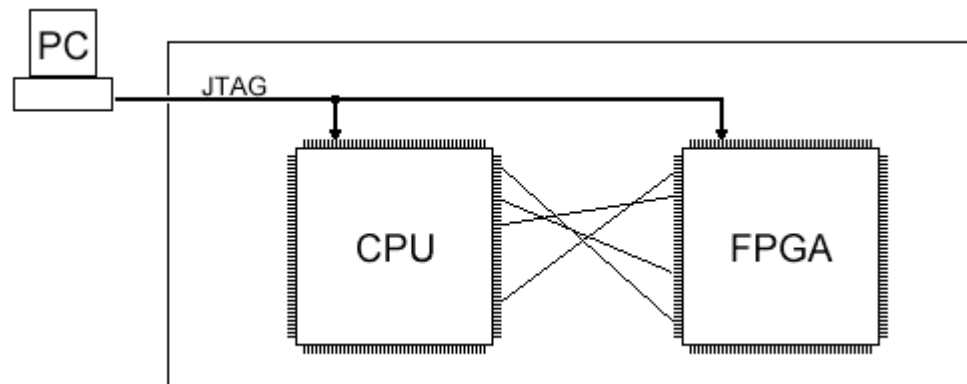
- ▶ **Nalaganje v Flash preko vmesnika SPI**
 - ▶ potrebujemo programator (npr. USB – SPI)
 - ▶ RESET je aktiven v času nalaganja programa



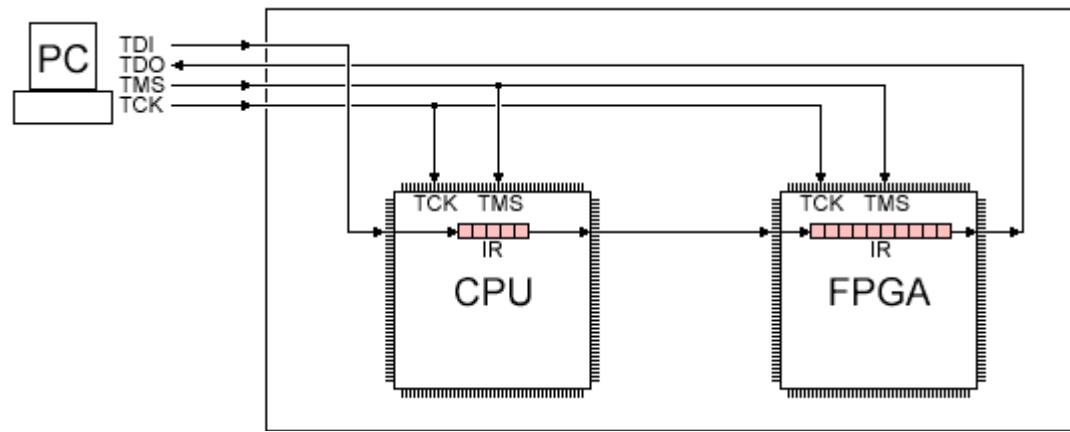
- ▶ **Uporaba posebnega programa v mikrokrmilniku**
 - ▶ Bootloader enkrat naložimo s programatorjem, nato pa omogoča prenos programa preko periferne enote
 - ▶ Arduino Bootloader preko RS232 in vmesnika USB-RS232
 - ▶ ob resetu procesorja in/ali ob določenem zaporedju, ki pride preko RS232 se aktivira Bootloader in začne z nalaganjem programa v Flash

Debugiranje preko JTAG

- ▶ JTAG je IEEE standard (1149.1) za periferno testiranje vezij
- ▶ Primer:
 - ▶ proizvodno testiranje povezav med dvema integriranima vezjema
 - ▶ zaradi majhnih ohišij ni mogoče testiranje s sponkami
 - ▶ preko JTAG odklopimo priključke od notranjosti vezja, nastavljamo stanja na priključkih in beremo stanja na drugem vezju



JTAG testiranje in programiranje



- ▶ Zaporedni prenos ukazov in podatkov
- ▶ Vezava v verigo (JTAG chain)
- ▶ Poleg ukazov po standardu še dodatni ukazi za programiranje vezja CPLD ali FPGA
- ▶ Programiranje in debugiranje mikrokrmilnikov preko JTAG
- ▶ Npr. ARM-7
 - ▶ ustavitev izvajanja ukazov in dostop do vseh registrov preko vmesnika JTAG