

Fakultet Tehničkih Nauka
Novi Sad

Arduino Joystick Shield

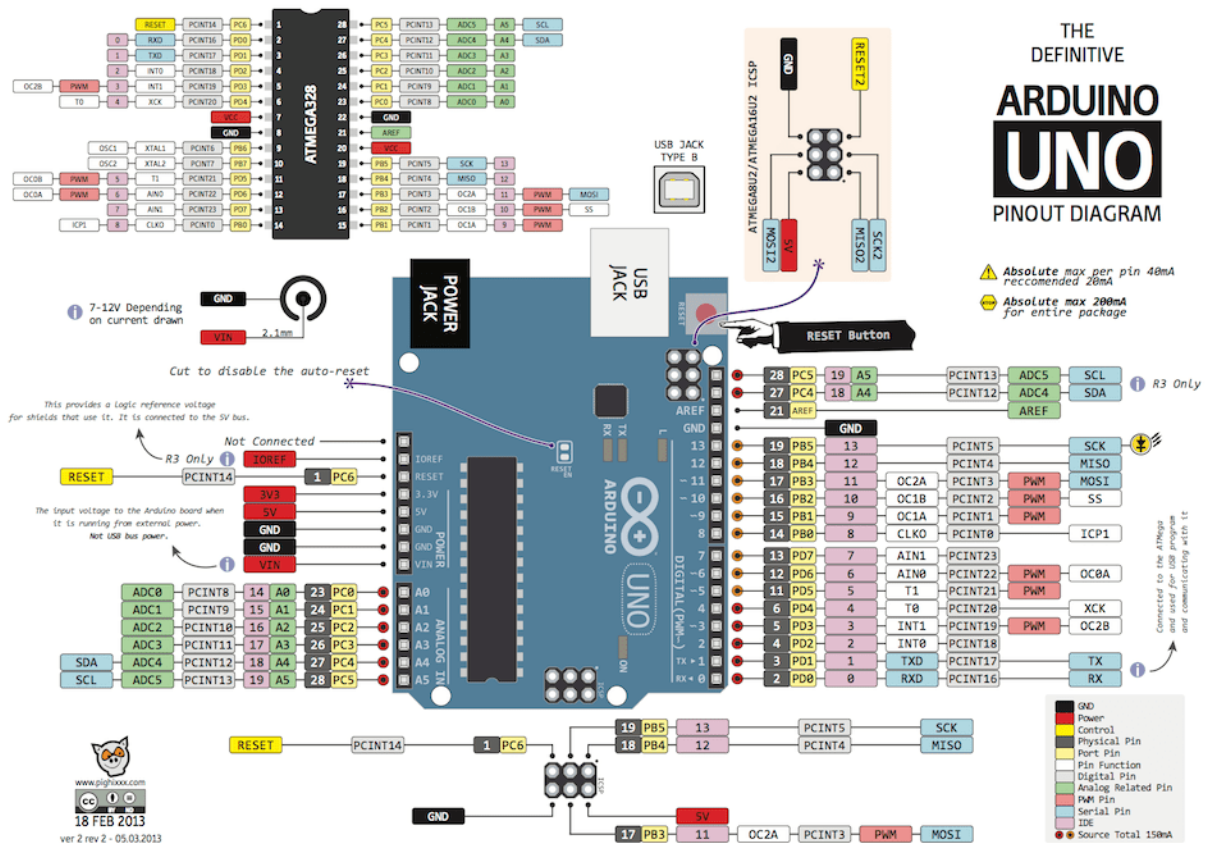
Predmet: Sistemska programska podrška u realnom vremenu 2

Projekat radili: Nikolina Anđelić, Nemanja Havlović, Kristina Pece

1. Uvod

1.1 Arduino

Arduino Uno je open source mikrokontrolerska ploča zasnovana na mikrokontroleru Microchip ATmega328P i razvijena od strane Arduino.cc. Ploča je opremljena pinovima za digitalni i analogni ulaz / izlaz (I / O) koji se mogu povezati sa različitim pločama za proširenje (shields). Ploča ima 14 digitalnih U / I pinova (šest sa PVM izlazom), 6 analognih U / I pinova i programabilna je sa Arduino IDE (Integrirano razvojno okruženje), putem USB kabla tipa B. Može se napajati USB kablom ili spoljnom 9-voltnom baterijom, mada prima napone između 7 i 20 volti.





1.2 Raspberry Pi

Raspberry Pi je single-board kompjuter veličine kreditne kartice koji je razvila u Ujedinjenom Kraljevstvu Fondacija Raspberry Pi sa namerom da promoviše i uči osnove informatike u školama i zemljama u razvoju.

Raspberry Pi 3 model B je najnoviji model pušten na tržište u februaru 2016. Kao operativni sistem na raspolaganju su : Raspbian, Ubuntu MATE, Snappy Ubuntu Core, Windows 10 IoT Core, RISC OS, Debian, Arch Linux ARM.

Koristi 1200 MHz quad-core ARM Cortex-A53 procesor. Posедуje memoriju od 1GB. Za skladištenje koristi MicroSDHC slot kao i njegovi prethodnici.

Grafiku predstavlja Broadcom VideoCore IV ali na većim clock frekvencijama nego njegovi prethodnici koji su radili na 250 MHz.

Snaga za napajanje ovog modela iznosi 4.0W

Joystick button	Associated Pin on Arduino board
KEY A	D2
KEY B	D3
KEY C	D4
KEY D	D5
KEY E	D6
KEY F	D7

1.4 SPI Protokol

Naziv SPI je skraćenica od Serial Peripheral Interface i namenjen je komunikaciji između integrisanih kola, kao i za relativno sporu komunikaciju sa perifernim jedinicama koje se nalaze na istoj štampanoj ploči.

SPI protokol je osmislila i patentirala Motorola.

Inicijalno SPI je osmišljen kako bi povezo mikrokontroler sa periferijama koristeći 4 žice i prilično je jednostavan.

SPI komunikacija podrazumeva da uvek u sistemu postoji jedinstveni master, dok je moguće postojanje većeg broja slave uređaja.

SPI protokol ne definiše maksimalnu brzinu komunikacije (za razliku od asinhronne serijske komunikacije, kod SPI je sve sinhronizovano sa SCLK signalom).

SPI ne omogućava nikakvu potvrdu prijema podatka, kao ni kontrolu toka komunikacije.

SPI master čak nije svestan ni da li je slave uređaj uopšte prisutan na SPI magistrali.

1.4.1 SPI signali

SCLK signal je taktni signal koji uvek generiše master i vodi se ka svim slave uređajima. Svi SPI signali su sinhronizovani u odnosu na ovaj signal.

SS signal predstavlja odabir slave-a (engl. Slave Select) Služi da odabere jedan od povezanih slave uređaja u datom trenutku, sa kojim će master da komunicira i da kaže slave uređaju da može da počne sa radom.

MOSI (Master Out Slave In) je linija podataka od mastera ka svim slave uređajima.

MISO (Master In Slave Out) je linija podataka od slave uređaja ka master uređaju - u slučaju datog zadatka se ne koristi jer ne postoji potreba da Raspberry Pi šalje podatke Arduino.

2. Zadatak

Potrebno je realizovati sistemsku programsku podršku za Arduino Joypad Shield povezan preko SPI protokola na RPI. Sa Arduino strane vodeći (Master), a sa RPI strane, SPI treba biti prateći (Slave). SPI protocol a Arduino realizovati bit-banging metodom koristeći `digitalWrite()` funkciju. SCK signal neizmenično menjati između 0 i 1, sa kašnjenjem između. Koristiti `delay()` funkciju za kašnjenje. MOSI signal sa podacima menjati na opadajuću ivicu SCK signala. Na RPI detektovati preko GPIO prolaza rastuću ivicu SCK signala i u njoj uzorkovati vrednosti MOSI signala.

3. Opis rešenja

3.1 Povezivanje

RPI	ARDUINO
25(GND)	GND
23(GPIO 11-CLK)	13(CLK)
19(GPIO 10-MOSI)	11(MOSI)
24(GPIO 8-CE0_N)	10(SS)

3.2 Raspberry Pi

3.2.1 Funkcija za inicijalizaciju modula (*`gpio_driver_init()`*)

Ova funkcija se poziva prilikom uvezivanja modula u kernel. U njoj se registruje novi chardev. Vršiti se inicijalizovanje odgovarajućih pinova. Zatim se inicijalizuju prekidne rutine.

3.2.2 Funkcije prekidne rutine

3.2.2.1 *`irqreturn_t h_irq_gpio08`*

Prekidna rutina je podešena tako da se poziva na opadajuću ivicu SS signala. Kada SS signal dobije vrednost 1, MOSI prestaje da se očitava

I dobijeni podaci se porede se željenim vrednostima, nakon čega se u kernel ispisuje koje dugme je pritisnuto I informacija se smešta u driver buffer.

3.2.2.2 *irqreturn_t h_irq_gpio11*

Prekidna rutina je podešena tako da se poziva na rastuću ivicu SCK signala. Kada je pozvana I SS signal je jednak 0, prima podatke MOSI signala od Arduina.

3.2.3 Funkcija za prebacivanje podataka korisničkoj aplikaciji (*gpio_driver_read()*)

Pozivom funkcije *copy_to_user* unutar *gpio_driver_read()*, podaci iz driver buffer-a koji predstavljaju informaciju o tome koje dugme je pritisnuto, se šalju I ispisuju u terminal.

3.2.4 Funkcija za brisanje modula (*gpio_driver_exit()*)

Ova funkcija se poziva kada se modul uklanja iz kernel. U njoj se oslobađaju handler-i i pinovi, memorija koja je prethodno zauzeta i oslobađa se *chardev*.

3.2.5 Ostale funkcije

Ostale funkcije preuzete su iz primera vežbi 6 I 7 I uglavnom služe za postavljanje pinova na Raspberry Pi ploči.

3.3 Arduino

3.3.1 Algoritam

1. U byte bang se smešta vrednost pritisnutog tastera
2. SS se postavlja na 0 što daje dozvolu za početak slanja informacija i poziva se *bitBangData* funkcija
3. SCK se postavlja na 1
4. MOSI šalje podatke bit po bit
5. Nakon poslatih 8 bita, SS se postavlja na 1 što predstavlja kraj slanja

4. Zaključak

Zadatak projekta je realizacija Linux rukovodilac za Raspberry Pi 2 uređaj. Rukovodilac treba da omogući SPI komunikaciju između Raspberry Pi-a i Arduina, gde je Arduino Master a Raspberry Pi Slave.

Na Arduinu koristimo Joystick Shield, koji ima više tastera koji trebaju da budu registrovani pri pritisku na taster i poslati na Raspberry Pi.

Raspberry Pi treba da očita vrednosti pritisnutog tastera i da ispiše na ekranu koji je taster pritisnut.