Univerzitet u Novom Sadu

Fakultet Tehničkih Nauka

Novi Sad

Departman za računarstvo i automatiku

Odsek za računarsku tehniku i računarske komunikacije

UART Protocol

Predmet: Operativni sistemi u realnom vremenu

Kandidati: Žarko Ostojić

Bogdan Petrić

Stefan Radovanović

Žarko Marić

Mentor: Miloš Subotić

Milorad Trifunović

Novi Sad, januar 2024.

1. Uvod i opis problema

Zadatak je da implementiramo UART na raspberry pi koristeći :

- ioctl() for addr/protocol  
- read()/write() data  
- Bit banging with gpio.h  
- hardware periph

Za bazu ćemo koristiti vežbe 7 :

* Tasks done:
  + copy motor\_ctrl driver to uni\_coms driver and rename
  + only header in include, main.c and gpio.c/.h
  + in main.c rename stuff and cut not needed, only gpio.
  + Check sw\_pwm.c for gpio init and use
    - TX – out from write()
    - RX – in to read()
  + bit-bang UART
    - Blocking write/read
    - No buffering
    - Poll gpio
    - no timer, use delay for 1ms
  + Test
    - Loop to itself
    - Need two threads from user space, 1st write(), 2nd read()

2. Teorijske osnove

# Uart (universal asynchronous receiver/transmitter) je fizičko kolo na mikrokontroleru. Njegova uloga je da prenosi I prima podatke u seriskoj komunikaciji. Prednost mu je što su za komunikaciju potrebne samo dve žice.

# https://www.circuitbasics.com/wp-content/uploads/2016/01/Introduction-to-UART-Basic-Connection-Diagram.png

# Uart komunikacija je asinhrona, što znači da ne postoji sistemtski takt koji usklađuje slanje I primanje. Umesto toga, koriste se biti za početak I kraj slanja (start I stop bit). Uart postavlja žicu na visoki napon kada nema transporta podataka. Pri započinjanju komunikacija, uart koji šalje podatke spušta napon žice na niski napon I tako uart koji prima podatke zna da je transfer započet. Kako nema takta, koristi se baud rate (brzina transvera podataka, izražen u bts (bits per second)). Na kraju komunikacija žica se postavlja na visoki napon.

3. Pokretanje rešenja

* Build-ovanje i pokretanje rešenja:
  + Na oba RPi pokrenuti dva terminal u jednom terminalu namestiti putanju do drajvera, a na drugom do testne aplikacije
  + Build-ovanje:
    - Terminal za drajver:
      * Uneti make build
      * Uneti make start
    - Terminal test:
      * ./waf configure
      * ./waf build
  + Pokretanje:
    - U terminalu za test na jednom RPi unosimo ./build/test\_receive
    - Na drugom RPi u terminalu za test unosimo ./build/test\_send

4. Rešenje:

U pitanju je kernel-space aplikacija koja se oslanja na gpio.h biblioteku za postavljanje i čitanje Rpi pinova. Logika je implementirana pomoću maski i bitwise operatora.

5. Testiranje i testna aplikacija

Za testiranje je izrađena user-space aplikacija. Potrebno je povezati dva RPI žicom. Moguće je koristiti jednu žicu za jednosmernu komunikaciju, ili dve žice za dvosmernu komunikaciju (rx jednog RPI se poveže na tx drugog, druga žica obrnuto). Rešenje je takođe moguće testiraju samo na jednom Rpi, ako se njegov tx žicom poveže na njegov rx.

Primer testiranja:

<https://www.youtube.com/watch?v=lM_K9dE2U9s>

6. Zaključak

Prednost korišćenja UART protokola jeste nezahtevanje system clocka, kao i laka implementacija na različitim uredjajima, a mana je što radi kao jedan master jedan slave, tj. nema povezivanja više uredjaja.

# literatura i izvori informacija :

<https://www.mbtechworks.com/hardware/raspberry-pi-UART-SPI-I2C.html>

<http://www.simplyembedded.org/tutorials/msp430-uart/>

<https://www.arduino.cc/reference/en/language/functions/communication/spi/>

<https://docs.arduino.cc/learn/communication/wire>