

RoboHand

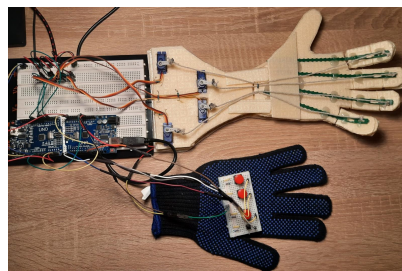
Autor

Olaru Gabriel Iulian 334CC

Introducere

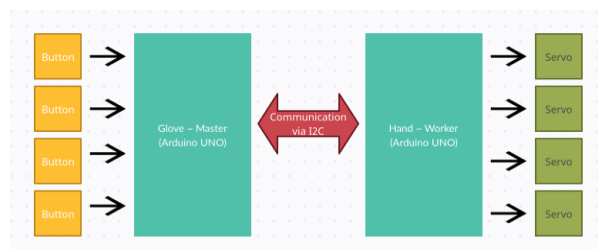
Am ales ca tema a proiectului realizarea unei maini robotice care sa urmareasca actiunile mainii utilizatorului.

Descriere generala



Utilizatorul va avea in palma o telecomanda cu 4 butoane, una pentru fiecare deget cu exceptiua degetului opozant. Prin flexarea degetelor acesta va apasa butonul si va trimite o comanda mainii robotice care va flexa la randul ei degetul indicat de utilizator. Cand utilizatorul va ridica degetul de pe buton (si nu va mai fi flexat) si degetul mainii robotice se va relaxa.

Schema Bloc



Hardware Design

Lista piese:

- 2x Arduino UNO
- 2x Cablu Alimentare
- 4x Butoane Push-Down
- 4X Servomotoare
- Misc: Fire, Breadbord, Rezistente
- Mecanic: Sfoara, Manusa, Hot-Glue, Polistiren, Baghete Plastic

In implementarea proiectului s-au folosit 2 placi arduino uno, ce comunica prin protocolul I2C astfel:

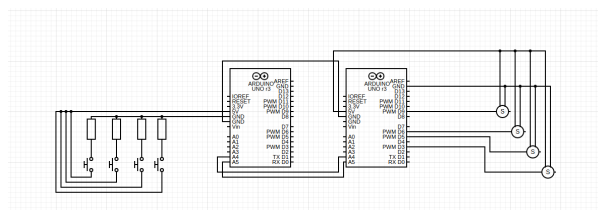
- Prima placa (manusa) serveste pe post de master. Aceasta primeste inputul utilizatorului si il transmite mai departe.
- A doua placa (bratul) serveste pe post de worker. Aceasta primeste comanda de la manusa si actioneaza servomotoarele ce articuleaza degetele bratului mecanic.

Pentru comunicarea dintre cele doua placi Arduino a fost ales protocolul I2C deoarece permite adaugarea foarte usoara a mai multor brate mecanice ce pot fi controlate de aceeași manusa, in cazul unei viitoare expansiuni.

Cele 4 butoane servesc la captarea inputului utilizatorului. Atunci cand acesta flexeaza degetul va apasa un buton. In revers, cand degetul nu va mai fi flexat, butonul nu va mai fi apasat.

Cele 4 servomotoare servesc la controlul flexiei degetelor mainii robotice.

Schema Electrica



Software Design

Software-ul este descris din doua perspective: master respectiv worker.

Master (Manusa):

- Initializeaza starea butoanelor si asociaza acestora pinii digitali de pe placa arduino, setati ca input.
- Monitorizeaza continuu starea butoanelor.
- In cazul schimbarii starii unui buton (este luat in calcul si prevenit un posibil debounce), initializeaza comunicarea I2C si transmite codificat index-ul butonului si noua sa stare. Va incheia apoi comunicarea pentru a nu tine canalul ocupat.
- Codificarea: $*codificare = ab = (10 * a + b)$ unde $a = id_buton + 1$; $b = stare\ on(1)/off(0)*$.
- Starea precedenta va fi stocata in memorie iar starea actuala va fi updatata.

Worker (Mana Robotica):

- Asociaza servomotoarelor pinii digitali capabili de PWM de pe placa.
- Seteaza servomotoarele la pozitia initiala.
- Initializeaza intreruperea aferenta primirii unei comenzi de la manusa.
- In cazul primirii unei comenzi:
 - Decodeaza comanda.
 - Flexeaza / Relaxeaza getetul corespunzator comenzii, dupa starea indicata de comanda, prin actionarea servomotoarelor.

Implementarea este una eficienta, cu putine delay-uri, fiind bazata pe intreruperi si pe trimiterea de mesaje strict necesare intre master si worker, fara congestia canalului de comunicare.

Rezultate obtinute

Toate functionalitatile prezentate au fost implementate, rezultand intr-un proiect interesant, usor de extins pe viitor. Functionalitatile sunt demonstrate in videoclipul din link-ul de mai jos.

<https://www.youtube.com/watch?v=H2pjVKa9vBY>

Concluzii

Implementarea proiectului a fost interesanta, imbinand atat programarea software si design-ul de circuite electrice cat si partea mecanica necesara constructiei mainii. Proiectul m-a ajutat de asemenea sa aprofundez cunostintele dobandite pe parcursul cursului de PM, folosindu-se de concepte din majoritatea laboratoarelor:

- Intreruperi venite din exterior.
- PWM pentru controlul servomotoarelor
- Comunicarea dintre doua placi arduino, prin I2C (motivul alegerii I2C in defavoarea SPI este prezentat mai sus).
- Citirea starii butoanelor si tratarea schimbarii state-ului + tratarea debouncing.

Rezultatul este un proiect interesant, la care imi va face placere sa revin pentru a adauga functionalitati noi si pentru a-l extinde.

Download

- Arhiva ce contine proiectul: [olaru_gabriel_robohand.zip](#)
 - Document pdf: https://ocw.cs.pub.ro/courses/pm/prj2021/amocanu/robohand?do=export_pdf
-

Jurnal

- 26.04 - Crearea paginii
 - 08.05 - Realizarea schemei
 - 10.05 - Comandarea componentelor
 - 27.05 - Inceputul partii software in simulator.
 - 01.06 - Circuitul
 - 02.06 - Partea mecanica (bratul robotic)
 - 03.06 - Finalizarea partii software, testarea si finalizarea proiectului.
-

Bibliografie/ Resurse

- Laboratoarele OCW: <https://ocw.cs.pub.ro/courses/pm>
 - State Change Detection: <https://www.arduino.cc/en/Tutorial/BuiltInExamples/StateChangeDetection>
 - Realizarea schemei electrice: <https://www.circuit-diagram.org/editor/>
 - Retailer: <https://www.optimusdigital.ro/ro/>
-

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/amocanu/robohand>



Last update: **2021/06/03 05:35**