

Scanner 3D

Introducere

Proiectul meu constă în creearea unui scanner 3d de mici dimensiuni. Cu autorul acestuia vom putea scana obiecte de mici dimensiuni din jurul nostru pentru a putea obține un fișier .obj al acestora. Cu acest fișier putem după să facem o multitudine de lucruri. Ideea din spatele proiectului a pornit de la jocul Minecraft, într-un moment în care ma jucam cu colegii. Pe Minecraft există anumite modalități, utilizând programe externe, de a importa fisiere în joc care să te ajute la construirea diverselor lucruri, oferind-o matrita pe post de ajutor. Aceste fisiere, cu extensia .litematica, pot fi obținute din fisiere .obj. Deci toată idea a pornit de la dorinta mea de a construit că mai exact diverse lucruri de prin casa în Minecraft, într-un mod că mai exact. Evident, aceasta este doar un mod de a utiliza proiectul meu. Alte aplicații ale proiectului meu constau în abilitatea de a printa 3d diferite obiecte, cum ar fi figurine de mici dimensiuni pentru machete sau de a importa obiecte din jurul nostru în programe care lucrează cu aceste tipuri de fisiere, cum ar fi Blender pentru editare scene 3d.

Prezentarea pe scurt a proiectului vostru:

- ce face
- care este scopul lui
- care a fost ideea de la care ați pornit
- de ce credeți că este util pentru alții și pentru voi

Descriere generală

Scanner-ul 3d are o placă de bază pe care se poziționează obiectul pe care dorim să îl scanăm. Senzorul de infraroșu scanăza în timp ce placă de bază se roteste. Dupa o rotație completă, senzorul își modifică poziția verticală și continua procesul până cand nu o să mai scanzeze nimic.

O schemă bloc cu toate modulele proiectului vostru, atât software cât și hardware însotită de o descriere a acestora precum și a modului în care interacționează.

Exemplu de schemă bloc: <http://www.robs-projects.com/mp3proj/newplayer.html>

Hardware Design

Lista de piese:

Motor Pas cu Pas 28BYJ-48 - 2 bucati

Driver motor ULN2003 - 2 bucati

Senzor de Distanță Infraroșu 20-150 cm GP2Y0A02YK0F - 1 bucata

Modul Slot Card Compatibil cu MicroSD - 1 bucata

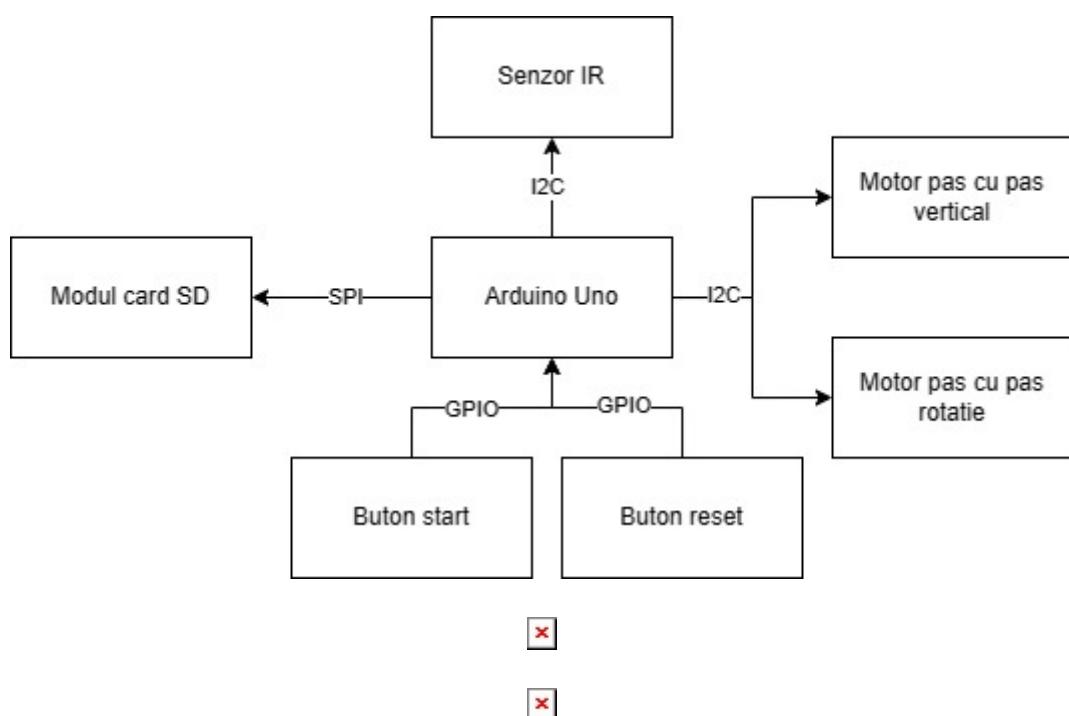
Buton - 2 bucati

Rezistență 10kΩ - 2 bucati

Fire

Placa breadboard

Arduino Uno



Pini utilizati:

1. Senzorul de distanță cu infraroșu utilizează pinul A0 deoarece el este un senzor analog.
2. Modulul de card SD utilizează pinii 10, 11, 12, 13 deoarece aceștia sunt pinii desemnați de Arduino pentru protocolul SPI. Pinul 12 este pentru MISO, pinul 11 este pentru MOSI, pinul 13 este pentru SCK, iar pinul 10 este pentru CS. Cel din urmă putea fi oricare altul, l-am ales pentru că este aproape de ceilalți pini.
3. Cele două drivere de motor utilizează pinii 2-5, respectiv 6-9.
4. Cele două butoane utilizează pinii 0 și 1. Chiar dacă aceștia sunt pini și pentru RX/TX, acest lucru nu este o problemă, deoarece nu utilizez interfața serială în timp ce proiectul funcționează, deci nu poate interacționa cu cele două butoane.

După o analiză, am realizat că motoarele pe care le utilizam erau prea slabe, având un cuplu aproximativ de 40 N x mm. Astfel, am ales să utilizez în locul lor motoare Nema 17, modelul 17HS4401, împreună cu driverele A4988, care au un cuplu de 40 N x cm.

Noua lista de piese:

Motor Pas cu Pas 17HS4401 - 2 bucati

Driver motor A4988 - 2 bucati

Senzor de Distanță Infraroșu 20-150 cm GP2Y0A02YK0F - 1 bucata

Modul Slot Card Compatibil cu MicroSD - 1 bucată

Butoane - 2 bucăți

Rezistență 10kΩ - 2 bucăți

Condensator 100 uF - 1 bucată

Fire

Placa breadboard

Arduino Uno

Pini noi utilizati:

3. Driverele pentru motoare utilizeaza pinii 9-8 si 7-6.

4. Pinii pentru cele doua butoane vor fi mutati pe pinii 2 si 3.



Software Design

<https://github.com/FraughtCoin/PM-3D-Scanner>

Aici se gaseste codul pentru proiect.

Am utilizat bibliotecile SPI.h, SD.h si GP2Y0A02YK0F.h. SPI.h si SD.h au fost utilizate pentru ca utilizez un adaptor de card SD care utilizeaza SPI. GP2Y0A02YK0F.h este o biblioteca pentru a facilita utilizarea senzorului de infrarosu cu acelasi nume.

Elementele de noutate reprezinta montajul fizic



Desi nu am putut testa codul final deoarece proiectul nu este in stare functionala din cauza ansamblului care ar fi trebuit sa realizeze miscarea pe verticala. Totusi, am testat toate celelalte componente individual, acestea functionand conform planului (si motorul care are treaba cu ansamblul de miscare vertical functioneaza corect, doar ca nu funcitioneaza ansamblul).

Explicatia codului:

-void setup(): in setup() initiez toate componentetele si pinii pe care ii utilizeaz, initiez cardul SD si caut un nume de fisier in care sa pot salva mai tarziu datele si resetez anumite valori la valoarea initiala, deoarece functia setup() va mai fi apelata si din alte parti din cod, pentru a putea reseta lucruri.
-void loop(): in loop() avem codul care ruleaza; avand doua butoane, mi-am creat doua variabile globale, scanning si reseting, cu ajutorul carora setez anumite conditii in functie de care stiu ce sa fac cand apas un anumit buton (daca programul nu scaneaza si nu este in bucla de resetare, prin apasarea butonului de start pot incepe scanarea, butonul de start nemaifacand nimic daca aceste conditii nu sunt indeplinite); in timpul scanarii, motorul roteste discul cu 1 pas, dupa care senzorul scaneaza si citeste distanta care este scrisa pe cardul SD, lucru care se intampla de 200 de ori, numarul de pasi pentru o rotatie completa a motorului; apoi, motorul vertical ar ridica ansamblul cu 1 cm, dupa care procesul eset reluat, pana se ajunge la limita de 10 cm.
-void rotateMotor(int pinNo, int steps, int vertical): aceasta functie roteste motorul respectiv de un numar dat de pasi; daca vertical este 1, se vor tine minte numarul de rotiri pentru a putea da reset in cazul in care dorim sa nu mai scanam un obiect.
-void writeToSD(double distance): functia scrie o valoare pe cardul SD. -void resetPosition(): functia aduce ansamblul vertical in pozitia originala.

Rezultate Obținute

Concluzii

Nu este atat de usor sa creezi ceva de la 0, este nevoie de o analiza o pieselor si componentelor disponibile, ce ofera fiecare in parte, cum se utilizeaza, ce necesita fiecare. De-a lungul proiectului, din cauza neatentiei la aceste detalii, am ajuns sa intampin o multitudine de probleme, de la piese care nu pot fi utilizate pentru scopul meu, pana la probleme la alimentarea acestora. Alimentarea poate sa fie o problema atat din punct de vedere al tensiunii, cat si al curentului. Este nevoie de o sursa capabila sa le ofere pe ambele.

Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună ✅.

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se

Încarcă fișierele este de tipul :pm:prj20???:c? sau :pm:prj20???:c?:nume_student (dacă este cazul).

Exemplu: Dumitru Alin, 331CC → :pm:prj2009:cc:dumitru_alin.

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2024/avaduva/ctudorache3101> 

Last update: **2024/05/27 06:58**