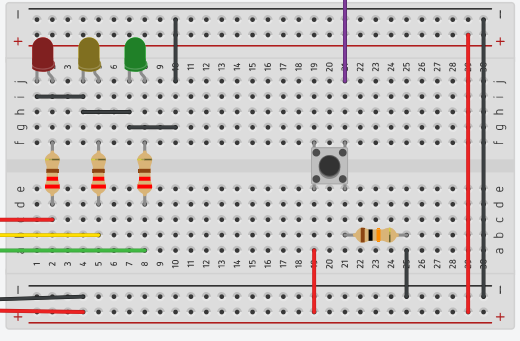
Projektrapport

Trafikljus-simulering



Samuel Bengtsson

Mölk utbildning

Linköping

2018-12-04

Omslagsbild : Trafikljuskrets

Källa : Samuel Bengtsson

Sammanfattning

Det här är en projektrapporten beskriver en simulering av en krets som ska vara någon sorts trafikljus med ett övergångsställe och en knapp. När man trycker på knappen så blir trafikljuset gult och efter ett tag rött, för att sedan gå tillbaks till grönt.

Simuleringen gjordes i Tinkercad som är ett gratisprogram som var mycket lätt att sätta sig in i. Koden till kretsen hittade jag på nätet. Anledningen till varför jag valde att simulera just det här projektet är för att det var enkelt att bygga kretsen i Tinkercad. Målet med det här projektet är att få testa hur simulering fungerar.

Förord

Den här projektredovisningen beskriver lite om simuleringen, svårigheter och vad som jag tyckt har varit lätt. Jag kommer inte beskriva exakt hur kretsen är gjord

I denna projektrapport.

Innehållsförteckning

1 – Inledning

2 – Metod

3 – Resultat

4 – Diskussion

5 - Källförteckning

6 - Bilaga

Inledning

Jag fick till uppgift att testa på att simulera ett tidigare projekt som vi har gjort i skolan. Att simulera är ett bra verktyg för att testa om en krets skulle fungera korrekt, utan att behöva bygga den i verkligheten. I Tinkercad som jag använde för att simulera kretsen så var det väldigt enkelt att dra sladdar, sätta dit komponenter på ett breadboard och ladda upp kod. Eftersom jag hade lite tid på mig så valde jag en lätt krets med lite funktionalitet. Koden laddade jag upp på en arduino som kretsen var kopplad till.

1

Metod

Kretsen består av ett breadboard, en arduino , tre LED:s, en knapp, fyra resistorer och ett antal sladdar. Först gick jag in i Tinkercad och såg att alla virtuella komponenter jag behövde för kretsen fanns att använda. Tinkercad vad mycket lätt att använda och det var bara att dra ut och släppa alla komponenter och sedan dra sladdar mellan. Koden var också väldigt lätt att få in på den “virtuella-arduinon”, det var bara att kopiera in den och ladda upp den.

2

Resultat

Simulering gick lättare än vad jag hare trott att den skulle göra. Jag har inte använt något simuleringsverktyg förutom Cisco - Packet tracer innan. Som resultat av simuleringen fick jag mer förståelse om vilket kraftigt verktyg simulering är, och hur lätt det faktiskt är. Trafikljuset fungerade precis så som det var tänkt innan.

3

Diskussion

Uppgiften jag valde att simulera finns på nätet och jag har testat göra en riktig krets av den tidigare. Det var lätt att bygga kretsen och koden var väldigt enkel att förstå sig på. Jag stötte inte på några riktiga svårigheter under den här simuleringen, det var så lätt att förstå sig på hur Tinkercad kunde användas för att bygga kretsar och ladda upp kod. Jag rekommenderar Tinkercad som simuleringsverktyg.

4

Källförteckning

<https://www.tinkercad.com>

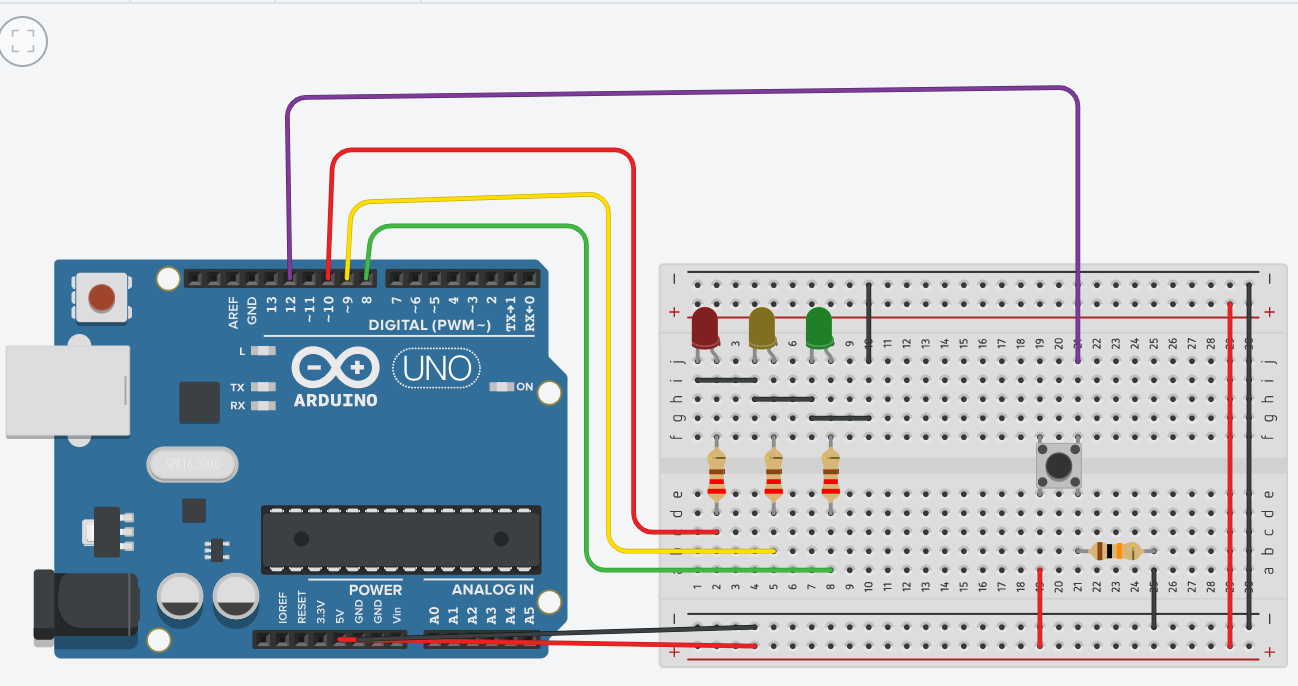
<https://www.makeuseof.com/tag/arduino-traffic-light-controller/>

<https://www.tinkercad.com/things/erR1CIkn8Wj-traffic-light-circuit>

5

Bilaga

Kopplingsschema:



Länk till simulering : <https://www.tinkercad.com/dashboard?type=circuits&collection=designs>

Källkod:

int red = 10;

int yellow = 9;

int green = 8;

int button = 12; // switch is on pin 12

void setup(){

pinMode(red, OUTPUT);

pinMode(yellow, OUTPUT);

pinMode(green, OUTPUT);

pinMode(button, INPUT);

digitalWrite(green, HIGH);

}

void loop() {

if (digitalRead(button) == HIGH) {

// if the switch is HIGH, ie. pushed down - change the lights!

changeLights();

delay(15000); // wait for 15 seconds

}

}

void changeLights(){

// green off, yellow on for 3 seconds

digitalWrite(green, LOW);

digitalWrite(yellow, HIGH);

delay(3000);

// turn off yellow, then turn red on for 5 seconds

digitalWrite(yellow, LOW);

digitalWrite(red, HIGH);

delay(5000);

// red and yellow on for 2 seconds (red is already on though)

digitalWrite(yellow, HIGH);

delay(2000);

// turn off red and yellow, then turn on green

digitalWrite(yellow, LOW);

digitalWrite(red, LOW);

digitalWrite(green, HIGH);

delay(3000);

}

6