

# Interactive poster

In dit document wordt beschreven welke technieken er gebruikt/uitgeprobeerd zijn om een interactieve poster te maken.

## Interactive by touch – poster



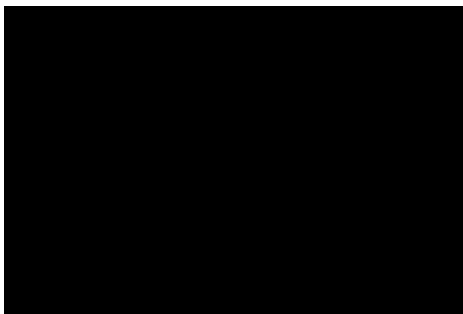
Beschrijving: Een webpagina wordt op een groot scherm getoond. Via deze webpagina kan de gebruiker een actie activeren op aanraking. Als een gebruiker een actie activeert, zal er een nieuwe pagina getoond worden waarop een animatie van Noek te zien (en eventueel te horen) is.

Software: HTML, CSS, Javascript

Hardware: scherm, HDMI kabel

Voordelen: makkelijk te maken en betrouwbaar.

Nadeel: gebruiker moet zorgen voor input in de vorm van het activeren van een knop.



<https://youtu.be/zuN-xXmmedk>

*Serial port communication Arduino -> Visual studio -> media player*

Het is onmogelijk om data naar visual studio te sturen, omdat visual studio geen seriële data kan uitlezen. Het omgekeerde is echter wel mogelijk maar Arduino kan geen video's afspelen. Deze (makkelijke) oplossing op het probleem is dus niet mogelijk.

<https://forum.arduino.cc/index.php?topic=428408.0>

*Interactive by sensor – poster*

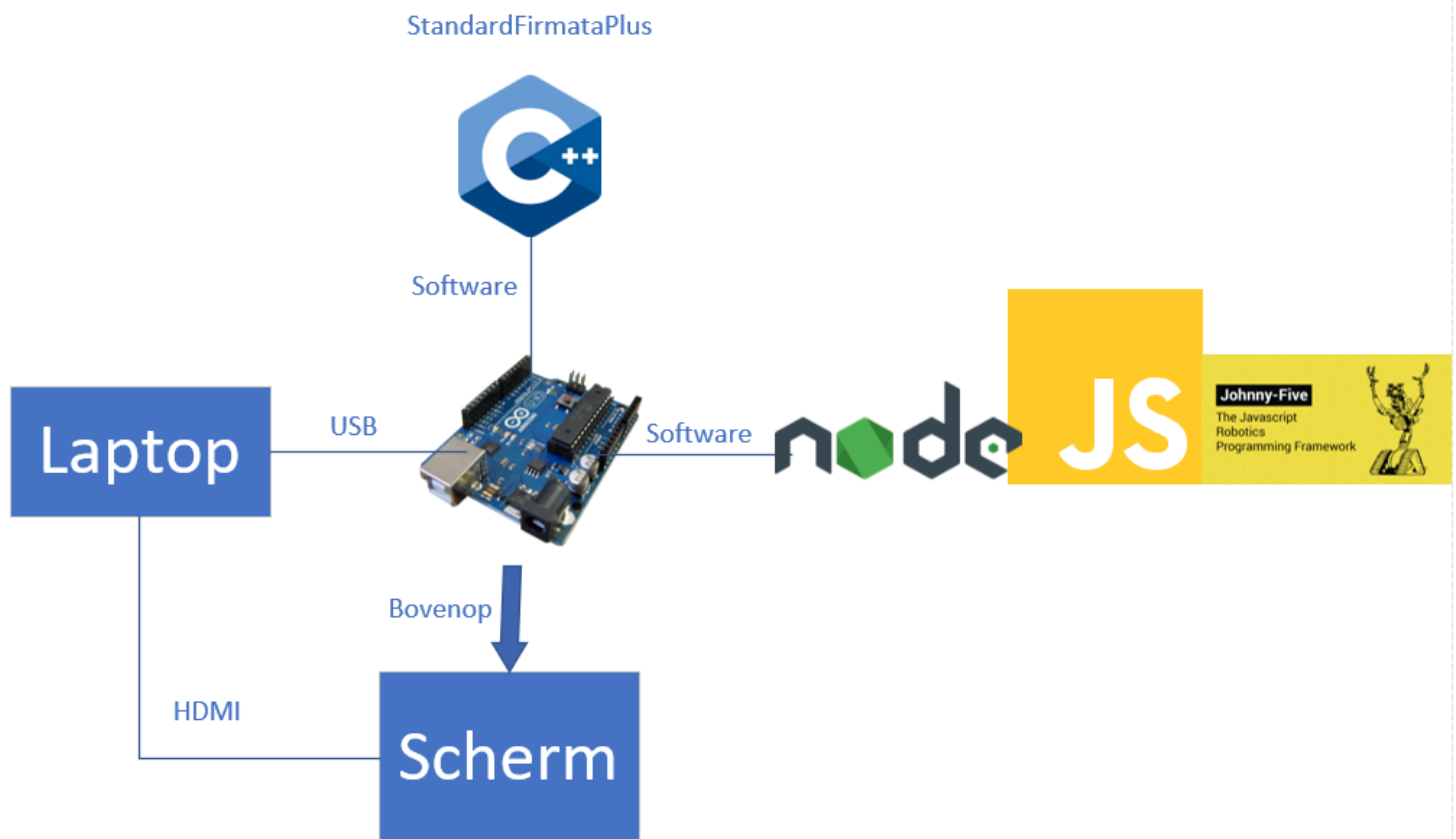
Beschrijving: een groot scherm heeft een Arduino Uno aan zich gekoppeld. Op het circuit board is een sensor (hc-sr04) verbonden naar de Arduino. De Arduino is gekoppeld aan een laptop waaruit de 'commands' worden gestuurd naar de Arduino. De 'commands' worden gemaakt met behulp van Node en de Johnny5 framework. Ook is er Firmata nodig om vanuit de laptop de commands te sturen en deze leesbaar te maken voor de Arduino. De laptop is via HDMI aangesloten op het scherm. Op het scherm worden animaties van Noek afgespeeld als de sensor iemand voorbij ziet lopen.

Software: Node Johnny5, Javascript, C++ (Firmata)

Hardware: Arduino Uno, jumper wires, hc-sr04, scherm, HDMI kabel

Voordelen: de gebruiker zorgt voor de input zonder hiervoor extra energie en tijd aan te besteden.

Nadelen: veel omwegen in de software om alles met elkaar te verbinden. Dit kan nadelig zijn voor de responsiviteit van het product.



POC's

## Creating Node.js server

Met dit POC wordt gekeken of er een server gemaakt kan worden met Node.js zodat daar de interactieve poster daar op kan worden gezet.

Input:

```
C:\Users\ricky\Documents\ICT Media\Media SM 2\DEV\POC codes\Node>node app
Server started on port 3000
Server started on port 2000
```

*Node server activation*

```
const http = require('http');
const fs = require('fs');
const hostname= '127.0.0.1';
const port = 3000;
const port2 = 2000;
fs.readFile('index.html', (err, html)=>{
  if(err){
    throw 'hello';
  }

  const server = http.createServer((req, res) =>{
    res.statusCode = 200;
    res.setHeader('Content-type', 'text/html');
    res.write(html);
    res.end();
  });
  server.listen(port, hostname, ()=>{
    console.log('Server started on port '+port);
  });
});

fs.readFile('index2.html', (err, html)=>{
  if(err){
    throw err;
  }

  const server = http.createServer((req, res) =>{
    res.statusCode = 200;
    res.setHeader('Content-type', 'text/html');
    res.write(html);
    res.end();
  });
  server.listen(port2, hostname, ()=>{
    console.log('Server started on port '+port2);
  });
});
```

*Javascript; server set-up*

← → ↻ ⓘ 127.0.0.1:2000

# This server has port: 2000

*Output*

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>2nd server</title>
</head>
<body>
  <h1>This server has port: 2000</h1>
  <script src="app.js"></script>
</body>
</html>
```

*HTML File #1*

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>1st server</title>
</head>
<body>
  <h1>This server has port: 3000</h1>
  <script src="app.js"></script>
</body>
</html>
```

*HTML File #2*

## LED blink

Met dit POC wordt gekeken of de Johnny5 library/framework daadwerkelijk de Arduino kan laten werken met Javascript.

```
C:\Users\ricky\Documents\ICT Media\Media SM 2\Johnny5>node arduino_led
1576766622940 Available COM3
1576766622974 Connected COM3
1576766626717 Repl Initialized
```

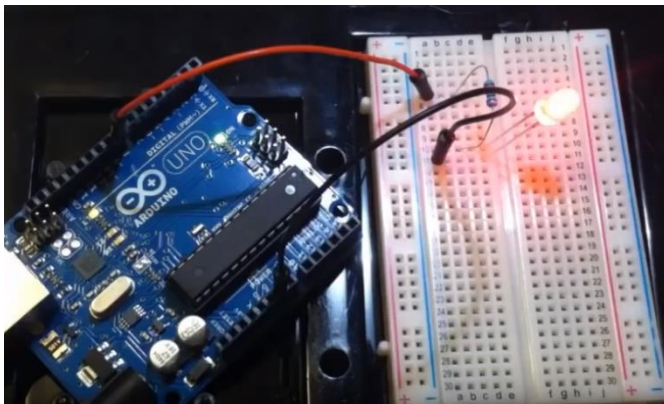
```
var five = require("johnny-five");
var board = new five.Board();

board.on("ready", function(){
  var led = new five.Led(11);

  led.on(2000);
  led.blink(500);

});
```

*Input*



*Output*

## Proximity - HC-SR04

Met dit POC wordt er gekeken of de waardes van de sensor kunnen worden uitgelezen met Node.js. Als dit mogelijk is, kan de volgende stap genomen worden; namelijk het schrijven van if-statements om de poster een actie te laten uitvoeren bij een bepaalde waarde van de sensor.

```
const { Board, Proximity } = require("johnny-five");
const board = new Board();

board.on("ready", () => {
  const proximity = new Proximity({
    controller: "HCSR04",
    pin: 7
  });

  proximity.on("change", () => {

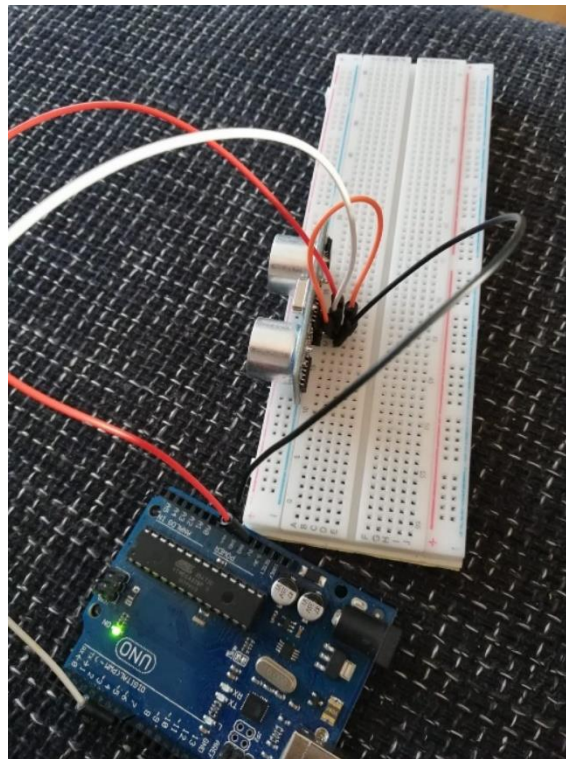
    const {centimeters} = proximity;
    console.log("Proximity: ");
    console.log("  cm  : ", centimeters);
    console.log("-----");

  });
});
```

```
C:\Users\ricky\Documents\ICT Media\Media SM 2\Johnny5>node sensor
1576750134477 Available COM3
1576750134488 Connected COM3
1576750138226 Repl Initialized
```

Input

```
cm  :  9.9
-----
Proximity:
cm  :  10
-----
Proximity:
cm  :  9.4
-----
Proximity:
cm  :  9.5
-----
Proximity:
cm  :  9.1
-----
Proximity:
cm  :  8.7
-----
Proximity:
cm  :  9
-----
```



Output

## Final step

Tenslotte moet de poster nog op de browser getoond worden. Hiervoor moet de javascript file verbonden worden met de HTML file.

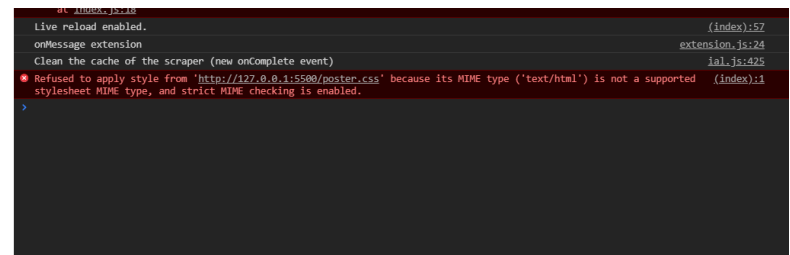
```
var http = require('http'),
    fs = require('fs');
fs.readFile('posternoek.html', function (err, html) {
  if (err) {
    throw err;
  }
  http.createServer(function(request, response) {
    response.writeHead(200, {"Content-Type": "text/html"});
    response.write(html);
    response.end();
  }).listen(80);
});

http.on("ready", () => {
  console.log("Does this work?");
});
```

Raak een knop aan om te activeren



[Dansen](#)  
[Zingen](#)  
[Zwaaien](#)  
[Complimentje](#)  
[Er was eens een...](#)



Aangezien Node niet in een browser runt moet er gebruik gemaakt worden van een framework/library.

Er is eerst geprobeerd met een speciaal NPM package genaamd 'nodebowl' toch een browser te krijgen die samen kan werken met Node. De package was echter nog in 'development' dus was het niet bruikbaar.

Helaas was er geen tijd meer voor verdere iteratie en is de poster niet gerealiseerd tot een volledig functionerend product.

Als er in de toekomst weer gewerkt moet worden met Node, Johnny-Five een browser en Arduino wordt dit onderzoek hervat. Onderstaande link bevat nuttige informatie over het 'runnen' van node.js apps in een browser.

<https://blog.cloudboost.io/how-to-run-node-js-apps-in-the-browser-3f077f34f8a5>

