

1. App for forenklet utvikling og kontroll av Arduino

Generelt

Appen har åpen kildekode, og [eksisterer på GitHub](#).

Der har den også [en egen wiki \(engelsk\)](#), som forhåpentligvis holdes oppdatert.

Hva den gjør

Appen lar deg kjøre [Processing](#) kode på telefonen din for å kontrollere Arduinoen din via bluetooth.

Sagt ganske enkelt kobler appen deg til en Bluetooth eller Bluetooth LE enhet og laster en eller flere Processing kildefiler du må ha lagt tilgjengelig på nett. Standard er at appen leser fra `folk.ntnu.no/[BRUKERNAVN]/plab/plab.pde`, men dette kan endres i innstillinger til appen (trykk på det lille tannhjulet oppe i høyre hjørne). Appen bruker [Processing.js](#) for å oversette Java koden til Javascript. Den injiserer et Javascript objekt inn i filen som ligger på hjemmeområdet ditt som gjør du kan snakke med bluetooth på en veldig forenklet måte.

Appen er bygd med Apache Cordova, noe som gjør den portabel til flere enheter. Den støtter for øyeblikket Android og iOS, men vi har hatt problemer med å få publisert iOS versjonen. En annen konsekvens av at vi har brukt Apache Cordova er at det du ser egentlig er en web side, men for deg har nok ikke det noe å si.

For mer teknisk info om appen og hvordan den behandler Processing skissa di, se om [Bluetooth og Processing](#), eller let i [kildekoden som ligger som open source på GitHub](#).

Plassering av Processing kode på hjemmeområde

Som standard ser appen etter filer på hjemmeområdet ditt på NTNU. Plasseringen er `folk.ntnu.no/[BRUKERNAVN]/plab/plab.pde`, men dette kan endres i innstillinger i appen (trykk på det lille tannhjulet i høyre hjørnet).

Den siste skissa du kjørte kan alltid kjøres på nytt, selv uten nett. Den mellomlagres i appen.

Opplasting av filer til hjemmeområdet

For å få lasta opp filer til brukerområdet ditt på NTNU er det flere muligheter.

Mulighet 1: Legg til mappen som nettverksmappe. Fordrer at du enten sitter på NTNU nettverket eller har en vpn tilkobling til NTNU sine servere. Før vi rekker å skrive det opp selv, kan [framgangsmåte for å koble til finnes her](#). Velg det systemet du sitter på under "grafisk". Vi anbefaler denne metoden.

Mulighet 2: Skaff deg en filoverføringsklient, utvikle koden lokalt og last den opp. Her finnes det nok mange mulige klienter, men vi kan si [FileZilla](#) ialle fall virker. Enkel tilkobling er å bruke samme server som i mulighet 3 (ssh) (`login.stud.ntnu.no`) via sftp (ssh) protokoll. Denne metoden er også bra. Fungerer like bra om du sitter på NTNU nettverk eller hjemme.

Mulighet 3: ssh deg inn til `login.stud.ntnu.no` og skriv koden direkte der. Sitter du på en Windows maskin kan du bruke et program som [PuTTY](#). Vi anbefaler ikke dette.

Hjemmesideområdet ditt på NTNU kan finnes ved å navigere deg inn i `public_html` (som er en snarvei til `/web/folk/[BRUKERNAVN]`). Opprett en mappe der som heter `plab`, og kall processing skisser du laster opp for `plab.pde`.

Hvor den finnes

For øyeblikket er den kun tilgjengelig for Android på [Play store under navnet PLab](#). Den er forhåpentligvis på vei til app store for iOS, men det gjenstår fortsatt noen utfordringer før Apple aksepterer den uten å levere inn diverse maskinvare.



Kjente problemer/ spesiell oppførsel

Tall

Appen kjører Processing.js, som oversetter vanlig Processingkode til Javascript før kjøring. Dette fører til at heltall ikke eksisterer når du kjører. Så hvis du er helt avhengig at tallet du skal bruke faktisk er et heltall, må du caste det eksplisitt: `int(tall)`

Musbehandling

For å få tak i mus posisjonen, generelt bruk `mouseX` og `mouseY`. For å få hvor musa var forrige gang bildet ble tegnet, bruk `pmouseX` og `pmouseY`.

`mouseDragged()`

Fungerer. Blir kalt når du drar fingeren over skjermen. Blir ikke kalt når du bare trykker uten å flytte fingeren.

`mouseMoved()`

Kalles kun en gang. Første gangen du trykker på skjermen blir den kalt, etter dette er den helt stille. Det gjør at bruksområder blir noe begrenset.

`mouse[EVENTTYPE](MouseEvent me)`

Fungerer ikke. Blir kalt på samme måte som den eventtypen det er snakk om (dragged/moved), men MouseEvent objektet blir ikke lagt med. Å prøve å referere noe i det objektet vil føre til at koden stopper (et exception blir kastet og uten behandling av det stopper bare utførelsen av resten av funksjonen).

Bilder

Fungerer på ingen måte som vanlig Processing. Det er flere grunner til dette, og info om hvorfor finnes på nettet (se lenken under for mer info).

For å få bilder til å virke i appen, må du gi absoluttadressen til bildet. Appen skal bare akseptere url-er som peker til `http://folk.ntnu.no/*`

Denne koden viser det generelle mønsteret vi anbefaler deg å bruke. Bytt ut BRUKERNAVN med brukernavnet ditt/brukeren som har lasta opp bildet, og STI med hvilken mappe det ligger i og navnet på fila.

Processing bilder i app

```
String imageURL = "http://folk.ntnu.no/BRUKERNAVN/STI";
PImage im;

void setup() {
  loadImage(imageURL);
}

void draw() {
  background(0);
  image(im, 0, 0);
}
```

Se info om [Processing.js](#) for mer info om hvordan det generelt fungerer i Processing.js.

Processing.js versjon

Appen bruker en gammel versjon av Processing.js, **versjon 1.4.1**. Den nyeste versjonen i skrivende stund er 1.4.15, men alle versjoner vi har testet over 1.4.1 (dette inkluderer 1.4.15) oversetter touch events til mouse events på en måte vi ikke er enige med. Og det er grunnen til en så gammel versjon benyttes i appen.

Kom med tilbakemeldinger

Vi er alltid åpne for nye forslag for hva som er praktisk og bra å ha med i appen.

Oppdager du noen problemer med den, så si ifra. Vi skal prøve å utbedre det.

Oppdager du noe du savner med den, noe funksjonalitet du skulle ønske var med så si ifra. Vi skal se om vi klarer å legge det med etterhvert.