

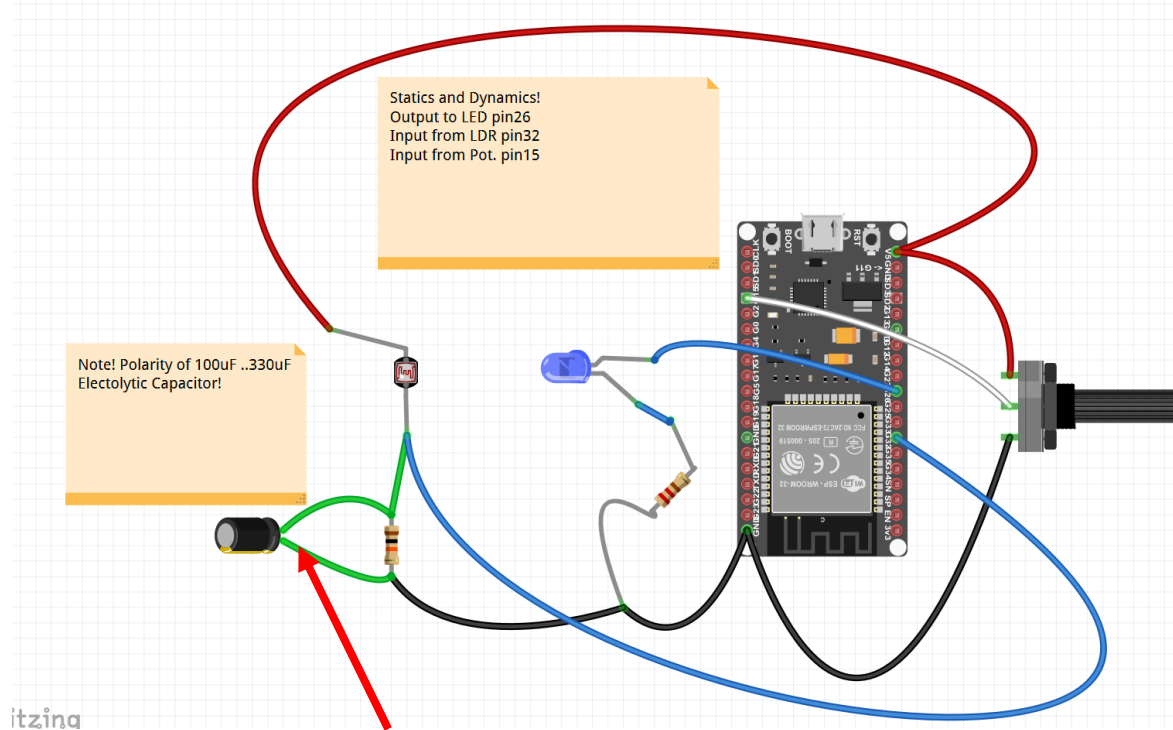
Statics and Dynamics of
ESP32

Tässä tulee ohjeita
harjoitustyön suorittamiseksi.

```
#include <jaksamista.h>
```

5. Statics and Dynamics exercise

Prosessin mallinnus on vaativa tehtävä.



100uF:n elektrolyyttikonkka mukaan kytkentään, jotta saadaan hieman hitaammin reagoiva prosessi.

1. Tee kytkentä ja todista PID-säätimen koodin avulla, että kytkentä toimii! 2p.

```
#include <PID_v1.h>

double Setpoint ;
double Input;
double Output ;
//Pins in ESP32
int LED = 26;
int LDR = 32;
int Pot = 15;

//PID parameters
double Kp=0, Ki=10, Kd=0;

PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);

void setup()
{
    Serial.begin(115200);

    Setpoint = 70;
    myPID.SetMode(AUTOMATIC);
    myPID.SetTunings(Kp, Ki, Kd);

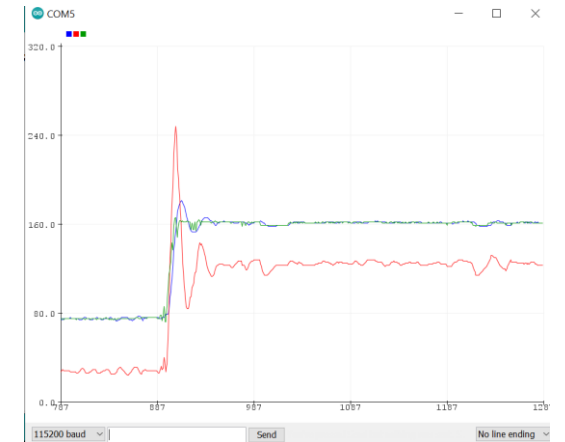
    ledcSetup(0, 1000, 8);
    ledcAttachPin(LED, 0);
}

void loop()
{
    Setpoint = map(analogRead(Pot), 0, 4096, 0, 255);
    Input = map(analogRead(LDR), 0, 4096, 0, 255);

    myPID.Compute();

    ledcWrite(0, Output);

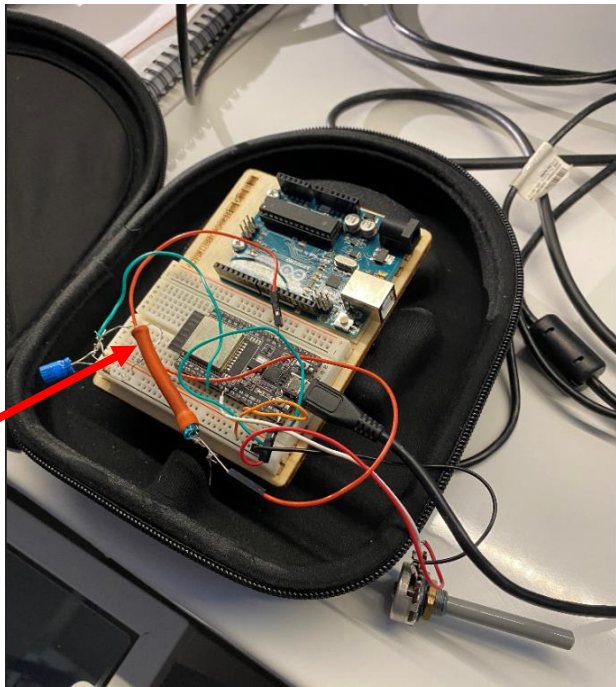
    Serial.print(Input);
    Serial.print(",");
    Serial.print(Output);
    Serial.print(",");
    /*Serial.print(Kp*100);
    Serial.print(",");
    Serial.print(Ki*100);
    Serial.print(",");
    Serial.println(Kd*100);*/
    Serial.print(",");
    Serial.println(Setpoint);
    delay(100);
}
```



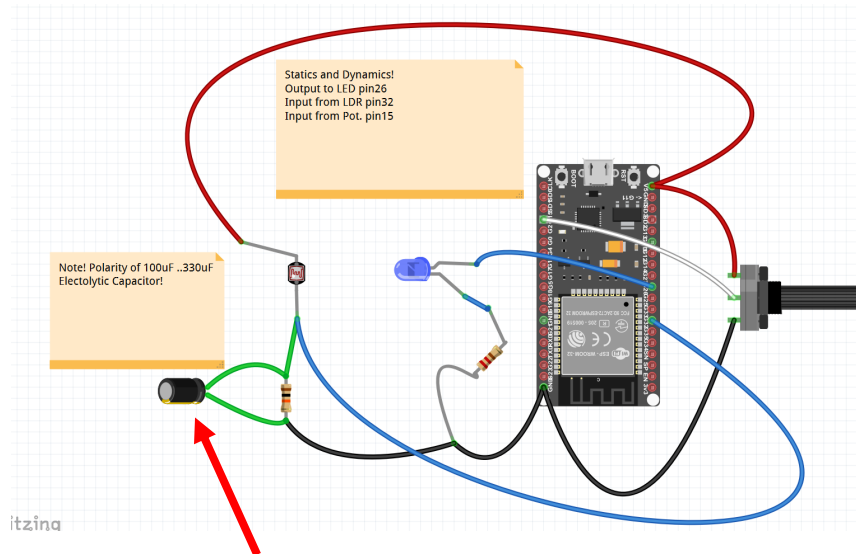
Ulkoinen valo häiritsee hommaa!

Koska valaistus häiritsee kytkentää, laitan kytkennän valoa läpäisemättömään pakettiin! Vain potentiometri jää ulkopuolelle!

Tuolla
kumiletkun
sisällä on LED
ja LDR.
Valitettavasti
LDR saa valoa
eulkopuolelta



Staattiset ominaisuudet testataan tällä softalla!



100 – 330uF:n elektrolyyttikonkka mukaan kytkentään, jotta saadaan hieman hitaammin reagoivaproessi.

3. Tee mittauksista kuvaaja Output(Input).
Tietenkään se ei ole viereisen näköinen! 2p

2. Testaa staattiset ominaisuudet! Tee taulukko, jossa saat eri Input -muuttujan arvoilla Output -muuttujan arvon! 2p

[illegible]

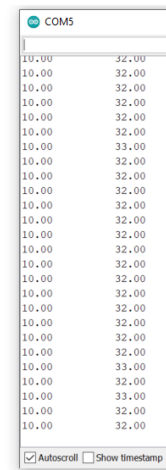
```
double Input;
double Output;

void setup() {
    // put your setup code here, to run once:
    Serial.begin(115200);
    delay(1000);

    Input = 10; //Tätä arvoa muutetaan välillä 0..255
    Output = 0;
    ledcSetup(0, 1000, 8);
    ledcAttachPin(26, 0);
}

void loop() {
    // put your main code here, to run repeatedly:
    ledcWrite(0, Input);
    Output = map(analogRead(32), 0, 4096, 0, 255);

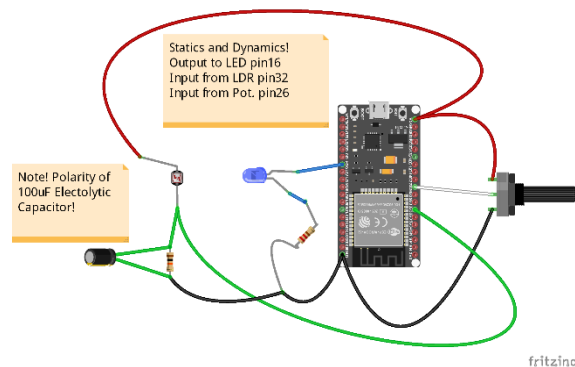
    Serial.print(Input);
    Serial.print(" ");
    Serial.println(Output);
    delay(100);
}
```



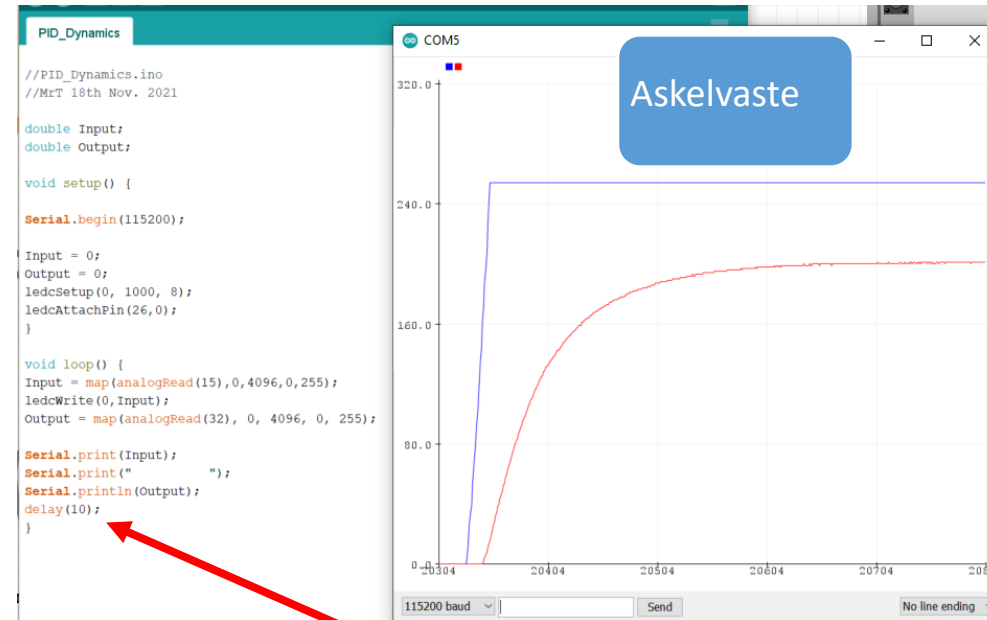
4. Koodi on aika tyhmä kun muuttuja Input on kovakoodattu tuonn
sisälle. Miten tämän testin voisi tehdä fiksummin? 1p

Dynaamiset ominaisuudet testataan askelvaste-testillä!

5. Tee askelvaste seuraavasti! Sääda Input arvo potentiometrillä nollaan. Odota hetki ja kierrä nopeasti potentiometri täysille eli arvoon 255. Koita keksiä miten saat plotter kuvaajasta snapshotin! 2p



Tämän sain napattua kömpelösti ctrl+PrtScr näppäinyhdistelmällä 😊.



Huom! Viive! Tämä vaikuttaa siihen kuinka nopeasti kuvaaja vilistää plotterin näytössä! Laita siihen lukuarvo 10

6. Tämäkin koodi on hieman kömpelö. Potikan vääntelemisen sijaan nyt voisi tuon askelmuutoksen kovakoodata tuonne koodiin. Kuinka sen tekisit? 1p