Testskjema

*PID Funksjonsblokk*

2021

# Introduksjon

**1.1 Hensikt**

***Hva skal testes og hvorfor?***

Komponenten som skal testes er en PID-regulator på GX Works 2. Komponenten skal brukes til å regulere en prosess, i dette tilfellet en tank.

For å sørge for at komponenten kan regulere en fysisk prosess, må det kobles opp et sammensatt system og blant annet stilles inn forskjellige parametere. Man må derfor gjennomføre diverse tester for å sikre at komponenten fungerer som den skal.

**1.2 Utstyrsliste**

***Hvilket utstyr (software/hardware) må være tilgjengelig/konfigurert for å gjennomføre testen?***

*Processshield kretskort. (Se figur 1) 1  
Arduino UNO mikrokontroller. 1  
Mitsubishi PLC (Helst av type FX2N/FX2NC) 1  
 - Kan ikke garantere at det fungerer med andre modeller.  
Analogkort FX0N-3A 1*

*GX Works 2 -*

*Arduino IDE -*

*Programkode PID (se vedlegg/link under) -*

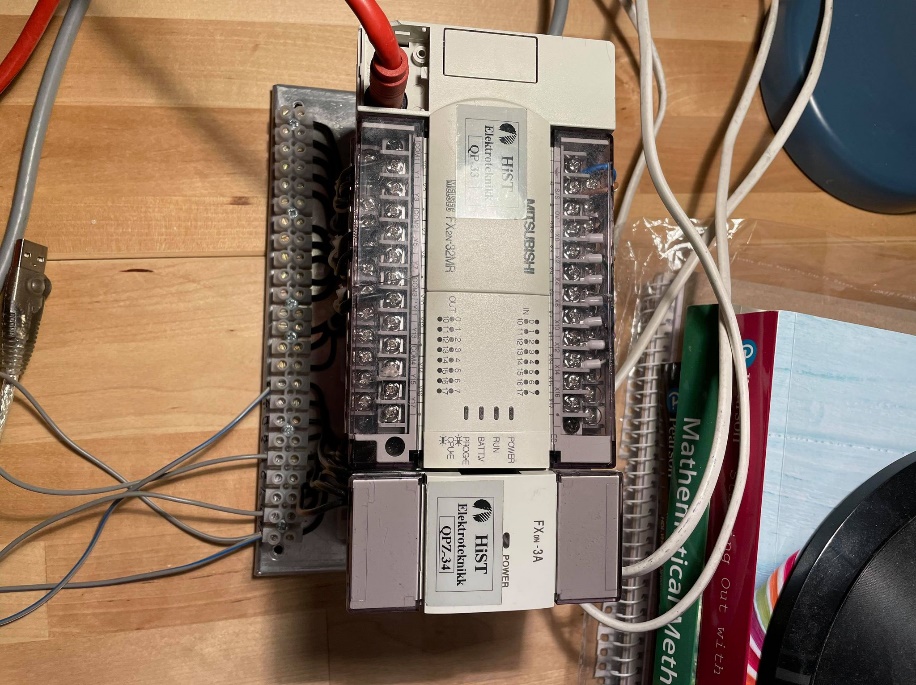
[*https://github.com/huvert/PID\_test.git*](https://github.com/huvert/PID_test.git)

**1.3 Oppsett**

***Her kommer litt mer utfyllende informasjon om oppsettet til testen, gjerne med illustrasjoner. Avsnittet skal gjøre det lettere for testansvarlig å utføre testen på korrekt måte.***

1. **Kalibrering av analogkort (FX0N-3A)**

Man må forsikre seg om at innganger og utganger på analogkortet opererer med samme spenninger som Processhield kretskortet.   
Gå til <https://github.com/huvert/PID_test.git> og last ned dokumentet:  
«FX0N-3A\_kalibrering\_av\_analogkort.pdf». Følg instruksjonene i manuelt og kalibrer både   
inputs og outputs.

1. **Fysiske oppkoblingerEt bilde som inneholder tekst, elektronikk, krets

   Automatisk generert beskrivelseEt bilde som inneholder tekst, elektronikk

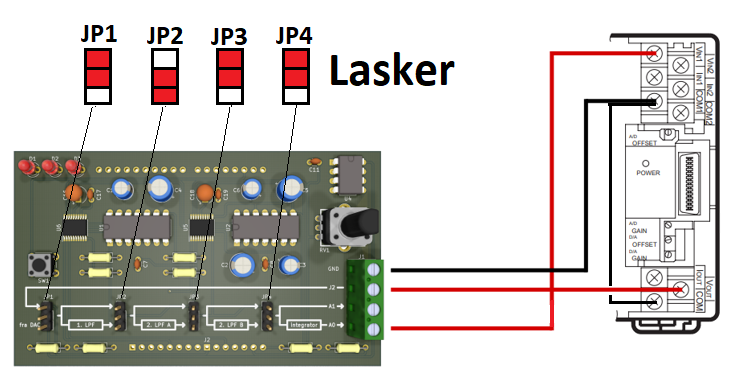
   Automatisk generert beskrivelse**

Figur 1: PLS-en til venstre koblet sammen med analogkortet til høyre.

Figur 2: Baksiden av kretskortet til høyre som skal kobles til Arduino Uno til venstre. Pinnene til venstre for kretskortet skal inn i inngangene til høyre for Arduino Uno.

Figur 3: Kobling mellom Arduino Uno og PC via USB-kabel.

Figur 4: PLS-en koblet til PC via USB-kabel.



Figur 5: Fysisk kobling mellom Processhield på Arduino UNO og AD/DA omformer på PLC for testing av PID.

1. **Laste opp programkoder**

Selve PID-regulatoren er ikke en fysisk komponent, men et program som skal operere i en PLS. Man må derfor først laste opp denne programkoden til PLS-en fra en PC.(Vedlegg: [Mitsubishi\_PLC\_PID\_test](https://github.com/huvert/PID_test/tree/main/Mitsubishi_PLC_PID_test))

Arduino Uno-en som benyttes med Processhield kretskortet kan i tillegg til å benyttes som en prosess, brukes til å plotte prosessverdien som kommer ut ifra prosessen (det førsteordens lavpassfilteret). Man må derfor også her laste dette opp til Arduino Uno fra en PC. (Vedlegg: [Arduino\_PID\_test](https://github.com/huvert/PID_test/tree/main/Arduino_PID_test))

# Testskjema

**Teknisk informasjon før igangsetting:**

|  |  |  |  |
| --- | --- | --- | --- |
| Testskjema oppsett godkjent: (Marker med kryss) | Godkjent: | | Ikke godkjent: |
| Hva som skal testes: | PID funksjonsblokk i GX Works | | |
| Demonstrasjonsansvarlig: |  | | |
| Kunde/tester: |  | | |
| Andre  tilstedeværende: | **Prosjektmedarbeidere:** | | |
| **Veiledere:** | | |
| Dato og sted: |  | | |
| Testoppsummering: | **Testing av PID regulator blokk på GX Works2**  Se til at PID-regulatoren og andre tilleggsfunksjoner fungerer som den skal. | | |
|
| Kriterier for godkjenning: | ***Hovedfunksjonalitet som skal demonstreres***   * *PID-reguleringen* * *Auto-manuell styring* | | |
|
| Resultat: (Marker med kryss) | Godkjent: | Ikke godkjent: | |

|  |  |
| --- | --- |
| Dato: | Sted: |
| Testansvarlig: |  |
|  |
| Kunde: |  |
|  |

**Testutførelse:**

|  |  |  |
| --- | --- | --- |
| 1. ***Kalibrere AD/DA spenning*** | **Forventet resultat** | **Godkjennelse** |
| Beskrivelse:  *For å unngå skade av testkort må vi forsikre oss om at inn og utspenninger på analogkortet til PLS-en opererer i riktig område.* | | |
| 1. Koble analogkortet til PLS-en | Se figur 1 |  |
| 1. Kalibrer analogkortet slik at arbeidsområdet ligger mellom 0 til 5V både ut og inn. | Se: [Guide\_til\_kalibrering](https://github.com/huvert/PID_test/blob/main/FX0N-3A_kalibrering_av_analogkort.pdf) |  |
| Kommentar:  Analogkortet er nå klart til å kobles til Arduino Uno og kretskortet. | | |

|  |  |  |
| --- | --- | --- |
| 1. ***Fysisk oppkopling*** | **Forventet resultat** | **Godkjennelse** |
| Beskrivelse:  *Koble sammen det analoge kretskortet til PLS-en* | | |
| 1. Koble laskene på kretskortet til det førsteordens lavpassfilteret | Se figur 5 |  |
| 1. Koble kretskortet til Arduino Uno | Se figur 2 |  |
| 1. Koble Arduino Uno til en PC via USB-kabel | Se figur 3 |  |
| 1. Koble PLS-en til en PC via USB-kabel | Se figur 4 |  |
| 1. Koble opp kretskortet til analogkortet på PLS-en. Pass på at lasker/jumpere er riktig plassert. | Se figur 5 |  |
| Kommentar:  Hele det fysiske oppsettet for testingen og er nå klar til å programmeres. | | |
|

|  |  |  |
| --- | --- | --- |
| 1. ***Laste opp programkoder for PID*** | **Forventet resultat** | **Godkjennelse** |
| Beskrivelse:  *Laste opp programkoder fra GX Works til PLS og fra Arduino IDE til Arduino UNO* | | |
| 1. Åpne Vedlegg ([PID\_test](https://github.com/huvert/PID_test)) |  |  |
| 1. Last ned filene som ZIP og pakk ut |  |  |
| 1. Derifra åpne:[Arduino\_PID\_test](https://github.com/huvert/PID_test/tree/main/Arduino_PID_test) |  |  |
| 1. Åpne «[Arduino\_PID\_test.ino](https://github.com/huvert/PID_test/blob/main/Arduino_PID_test/Arduino_PID_test.ino)» i Arduino IDE |  |  |
| 1. Velg riktig COM port i IDE. |  |  |
| 1. Kompiler og last opp kode. |  |  |
| 1. Åpne SerialPlotter (CTRL + SHIFT + L) |  |  |
| 1. Åpne fra ZIP mappen «[Mitsubishi\_PLC\_PID\_test](https://github.com/huvert/PID_test/tree/main/Mitsubishi_PLC_PID_test)» |  |  |
| 1. Åpne program «[PID\_program\_testing.gxw](https://github.com/huvert/PID_test/blob/main/Mitsubishi_PLC_PID_test/PID_program_testing.gxw)» |  |  |
| 1. Kompiler program |  |  |
| 1. Trykk «write to PLC», NB: velg riktig COM port. |  |  |
| 1. Trykk så «Start monitoring» |  |  |
| Kommentar:  Du har nå åpen Arduino IDE med Serial plotter koblet til kretskortet og Processhield, og GX Works 2 koblet til PLS-en. | | |
|

|  |  |  |
| --- | --- | --- |
| 1. ***Justere P parameter*** | **Forventet resultat** | **Godkjennelse** |
| Beskrivelse:  *Test P-leddet ved å bruke regulatoren som en P-regulator* | | |
| 1. Still Kp til 2.0. |  |  |
| 1. Deaktiver I og D ved å sette Ti og Td til 0.0 |  |  |
| 1. Still SP til 0.0. | PV skal gå mot 0. |  |
| 1. Gjør et sprang ved å sette SP til 130.0 | Plot på Arduino skal ligne på bilde. Se [Bilder](https://github.com/huvert/PID_test/tree/main/Mitsubishi_PLC_PID_test/PID_testskjema_bilder) |  |
| Kommentar:  *Man skal se ett sving og ha et stasjonært avvik.* | | |
|
|

|  |  |  |
| --- | --- | --- |
| 1. ***Justere I parameter*** | **Forventet resultat** | **Godkjennelse** |
| Beskrivelse:  *Test I-leddet ved å bruke regulatoren som en PI-regulator* | | |
| 1. Deaktiver D-leddet ved å sette Td til 0.0 |  |  |
| 1. *Still Kp til 2.0, Ti til 4.0.* |  |  |
| 1. Still SP til 0.0 | PV skal gå mot 0. |  |
| 1. Gjør et sprang ved å sette SP til 130.0 | Plot på Arduino skal ligne på bilde. Se [Bilder](https://github.com/huvert/PID_test/tree/main/Mitsubishi_PLC_PID_test/PID_testskjema_bilder) |  |
| Kommentar:  Prosessen skal nå ikke lenger ha et stasjonært avvik. Sammenlign med P-regulatoren. | | |
|

|  |  |  |
| --- | --- | --- |
| 1. ***Justere D parameter*** | **Forventet resultat** | **Godkjennelse** |
| Beskrivelse:  *Test D-leddet ved å bruke regulatoren som en PD-regulator* | | |
| 1. Deaktiver I-leddet ved å sette Ti til 0.0 |  |  |
| 1. Still Kp til 2.0 og Td til 1.0. |  |  |
| 1. Still SP til 0.0 | PV skal gå mot 0. |  |
| 1. Gjør et sprang ved å sette SP til 130.0 | Plot på Arduino skal ligne på bilde. Se [Bilder](https://github.com/huvert/PID_test/tree/main/Mitsubishi_PLC_PID_test/PID_testskjema_bilder) |  |
| Kommentar:  Prosessen skal nå regulere mer aggressivt sammenlignet med P-regulatoren. | | |
|

|  |  |  |
| --- | --- | --- |
| 1. ***Test Manuell /Auto kjøring*** | **Forventet resultat** | **Godkjennelse** |
| Beskrivelse:  *Etter at man har forsikret seg om at PID-reguleringen fungerer kan man teste overgangen mellom manuell og auto kjøring. Se til at MV holder verdien den hadde når man går over til manuell og at Set-punktet holder verdien den hadde og I leddet justeres når man går over til auto.* | | |
| 1. *Kjør PID-regulatoren fra tidligere deltest med en SP som ikke er 0.* |  |  |
| 1. Bytt over fra auto til manuell ved å sette Bit-en «MANAUTO» som går inn på «Man» til 1 | Variabelen «manuelt\_paadrag» skal få samme verdi som pådraget var da man skiftet fra Auto til Manuell |  |
| 1. Endre manuelt\_paadrag | Endringen i pådraget skal gi utslag i prosessen. |  |
| 1. Bytt over fra manuell til auto ved å endre MANAUTO til 0 | SetPoint skal bli lik prosessverdien i det man byttet over, og MV forblir den samme ved at Ti endres. |  |
| Kommentar:  Observer at overgangen til og fra man/auto ikke lager noe hopp i MV (Utgangen på regulatoren). | | |
|