10.11.2025

ELE201 - Semesterprosjekt

RC-bil

Abdo, Abdulrahman, Oskar, Thomas
GRUPPE 5

Innhald

1. Prosjektbeskriving

Målet med prosjektet er å utvikle en fjernstyrt bil (RC-bil) som kontrollerast via en PS4-kontroller. PS4-kontrolleren koplast til en PC via Bluetooth, og et Python-program på PC-en leser kontrollens signal. Python-skriptet sender deretter styringskommandoar via UART (Bluetooth-modul HC-05) til en STM32F767 mikrokontroller, som styrar motorane og servoen. Prosjektet fokuserer på kommunikasjon mellom PC og mikrokontroller, og motorstyring ved bruk av L293D.

2. Teori og bakgrunn

Prosjektet nyttar UART-kommunikasjon mellom PC og STM32F767, der Bluetooth-modulen fungerer som trådlaus bru. Motorstyring implementerast ved bruk av L293D H-bru for å kontrollere retning og hastigheit på to DC-motorar. Ein micro servo (FS90) brukast til å styre vinkelen på forhjula ved å bruke ein 3d printa arm. Python-skriptet på PC-en bruker PS4-kontrollere sin input (via pygame-biblioteket) for å sende kommandoar til bilen.

3. Utstyr og metode

3.1 Utstyr

Komponent	Antall	Funksjon
DC-motor	2	Framdrift og bakdrift
Micro servo FS90	1	Styring av forhjul
L293D	1	Motorstyring (H-bru)
9V batteri	1	Straumforsyning
HC-05 Bluetooth-modul	1	Trådlaus kommunikasjon mellom PC og STM32
PS4-kontroller	1	Brukarinput via PC
STM32F767 Nucleo-144	1	Mikrokontroller
Hjul	4	Bilens hjulsett
Breadboard	1	For kopling av kretsane

Verktøy: Python, STM32CubeMX, Serial Bluetooth Terminal (Android-app) og 3D-printer

3.2 Metode

1. PS4 kontroller sender styringssignal til PC via Bluetooth
2. PC med Python program, leser input frå PS4 kontroller og gjer den om til UART kommandoar M <venstreHjulFart> <høgreHjulFart> S <servoVinkel>.
3. Deretter tar mikrokontrolleren imot kommandoane via UART og styrer motorane via L293D og servoen.

3.3 Kommunikasjon og styring

Digital I/O: Styrer motorretninga gjennom L293D

Analog I/O: Styrer motorfart og servovinkel

Seriel I/O: Overfører kommandoar mellom PC-en og mikrokontrolleren

Bluetooth (HC-05): trådlaus overføring mellom PC og mikrokontroller

Python programmet sender konstant kommandoar til STM32, som tolkar og oppdaterer motorar og servo.

Her er korleis dei forskjellege komponentane kommuniserer med kvarandre.

PS4-kontroll -> PC (Sender styringssignal via Bluetooth)

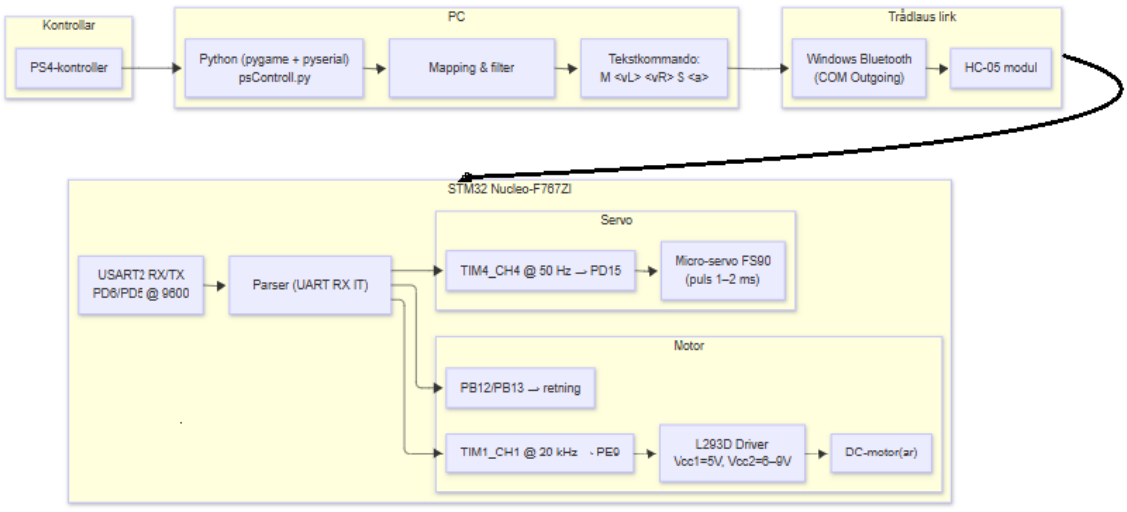
PC (Python) -> STM32 (Sender kommandoar via HC-05 (UART))

STM32 -> L293D og Servo (Styrer motorar og servovinkel)

L293D -> Motorar (Gjer bevegelsar basert på signal frå STM32)

3.4 Arbeidsflyt

1. Start Python programmet og kople til PS4-kontroll
2. Les joystick- og knappedata
3. Oversett input til kommandoar (M og S)
4. Send kommando via Bluetooth til STM32
5. STM32 tolkar kommandoar og oppdaterer motorar og servo
6. Prosessen blir gjentatt



3.5 Nettverksdel

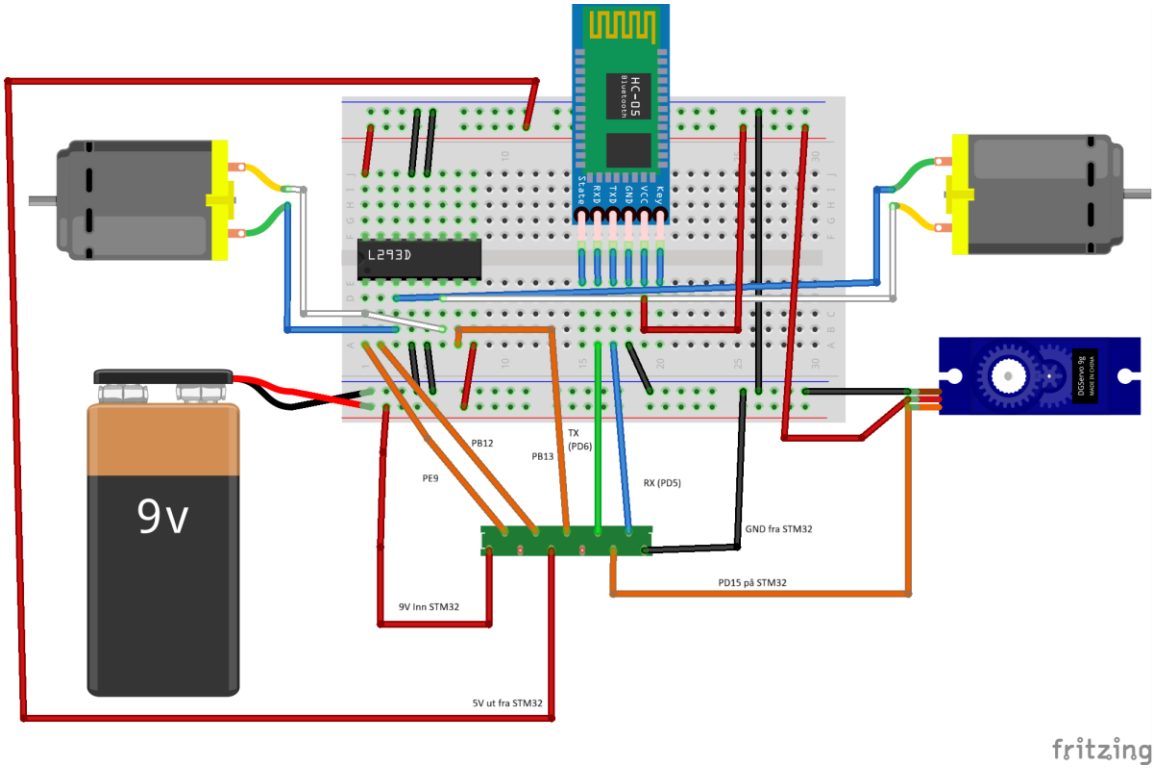
Her er korleis vi ville segmentert nettverket slik at det passar til

Networks	Needed Hosts	Subnet	First	Last
Sensor Network:	1000 | 1024	10.128.0.0/22	10.128.0.1	10.128.3.254
Management Network	6 | 8	10.128.4.0/29	10.128.4.1	10.128.4.6
Server Network	3 | 8	10.128.4.8/29	10.128.4.9	10.128.4.14
Link R1-R2:	2 | 4	10.128.4.16/30	10.128.4.17	10.128.4.18

Device	Interface	IP-Address
R1	Sensor LAN	10.128.0.1
R1	Management LAN	10.128.4.1
R1	R2 Link	10.128.4.17
R2	R1 Link	10.128.4.18
R2	Server LAN	10.128.4.9
S1	Sensor LAN	10.128.0.2
S2	Management LAN	10.128.4.2
S3	Server LAN	10.128.4.10

4. Løysningsdesign

Systemet består av tre hovuddelar: PS4-kontroller, PC med Python-skript og STM32F767-mikrokontroller. Python-skriptet leser signala frå kontrollaren og oversett dei til kommandoar som sendes til STM32 via Bluetooth (UART). STM32 mottar kommandoane og styrer to DC-motorar og ein servo. Den eine motoren er invertert for å gi korrekt drift framover og bakover.



5. Resultat og testing

Vi begynte med å kople DC motorane og servoen til ein app på Android (Serial Bluetooth Terminal) via å bruke bluetooth-modul (HC-05) der vi kunne teste om kommandoane funka eller ikkje. Når alt på appen funka så laga vi eit Python-skript som tok imot akkurat same kommando som appen. Deretter satt vi opp styring av bilen ved hjelp av ein PS4-kontroller, og når alt dette fungerte flytta vi alle komponentane over på den 3D-printa bilen. Etter vi gjorde dette så møtte vi på fleire problem, deriblant at framhjula ikkje gjekk rundt, servoen vi brukte slutta å fungere så vi fekk ikkje svinge som vi ønska, og at vi måtte bytte en jumper på mikrokontrolleren for å bruke VIN pinnen.

Figur 4: Testoppsett

6. Diskusjon og vurdering

Prosjektet fungerte som planlagt, med presis styring og stabil Bluetooth-kommunikasjon. Utfordringar inkluderte tuning av PWM-signaler for hastigheitskontroll og synkronisering mellom PC og mikrokontroller. Moglege forbetringar inkluderer betre batterikapasitet og implementering av tovegs kommunikasjon (tilbakemelding frå bilen til PC, som deretter kunne bli sendt til ein webserver). Ein ting vi kunne ha gjort for å forbetre det enda meir er å legge til ein av/på brytar som vi kunne bruke til å skru bilen av og på sånn at vi slepp å kople av og på batteriet.

7. Konklusjon

Vi fekk til å lage ei fjernstyrt bil med ein STM32F767 som basis. Bilen kan styrast ved hjelp av UART- og Bluetooth-kommunikasjon ved å bruke ein PC som eit mellomledd for PS4-kontrollen og HC-05 Bluetooth modulen. Prosjektet fungerte som vi håpte, og bilen reagerte greitt på kommandoane frå kontrollen. Vi lærte mykje om korleis kommunikasjon mellom PC og mikrokontroller fungerer, og korleis ein kan styre motorar og servo ved hjelp av PWM. Det var nokre utfordringar undervegs, som med servoen og straumtilkoplinga, men alt i alt blei vi fornøgde med resultatet og kva vi lærte.

8. Bruk av kunstig intelligens i prosjektet

Vi har brukt KI fleire gongar under prosjektet som eit slags hjelpemiddel undervegs. Først brukte vi KI til å få idear til korleis vi kunne kople ein PS4-kontroller til STM32-kortet. Då foreslo KI at vi kunne bruke Python som mellomledd, slik at PC-en kunne lese signal frå kontrollen og sende kommandoar vidare til mikrokontrolleren via Bluetooth. Det blei eigentleg grunnideen bak heile styringssystemet vårt.

Etterpå brukte vi KI til å finne ut av kva for aksar og knappar som høyrde til dei ulike funksjonane på PS4-kontrolleren, slik at vi kunne styre gass, brems og ratt riktig i koden. Undervegs i prosjektet brukte vi og KI til småting og feilsøking, som når vi fekk feilkodar i STM32-prosjektet, eller når noko ikkje fungerte som det skulle i Python-skriptet. Då fekk vi gode forklaringar og tips som gjorde at vi kom oss vidare mykje raskare.

9. Referansar

Datablad for STM32f767: https://www.st.com/resource/en/datasheet/stm32f765zi.pdf

Datablad for L293D: https://www.st.com/resource/en/datasheet/l293d.pdf

Datablad for HC-05 : https://components101.com/sites/default/files/component_datasheet/HC-05%20Datasheet.pdf

HVL - ELE201 Microcontrollers and Data Networks: https://fjnn.github.io/hvl-ele201/lectures/l7-motordrive

10. Vedlegg

Kode til bilen og Python-skript: https://github.com/h597236/ELE201RCBil

Youtube link: https://youtube.com/shorts/RwyVuzv6o0o?si=eI0RXEnbgqYyiK8G