

Softont user manual

Jaar 3 periode 2

Roy Dupon - 1824571

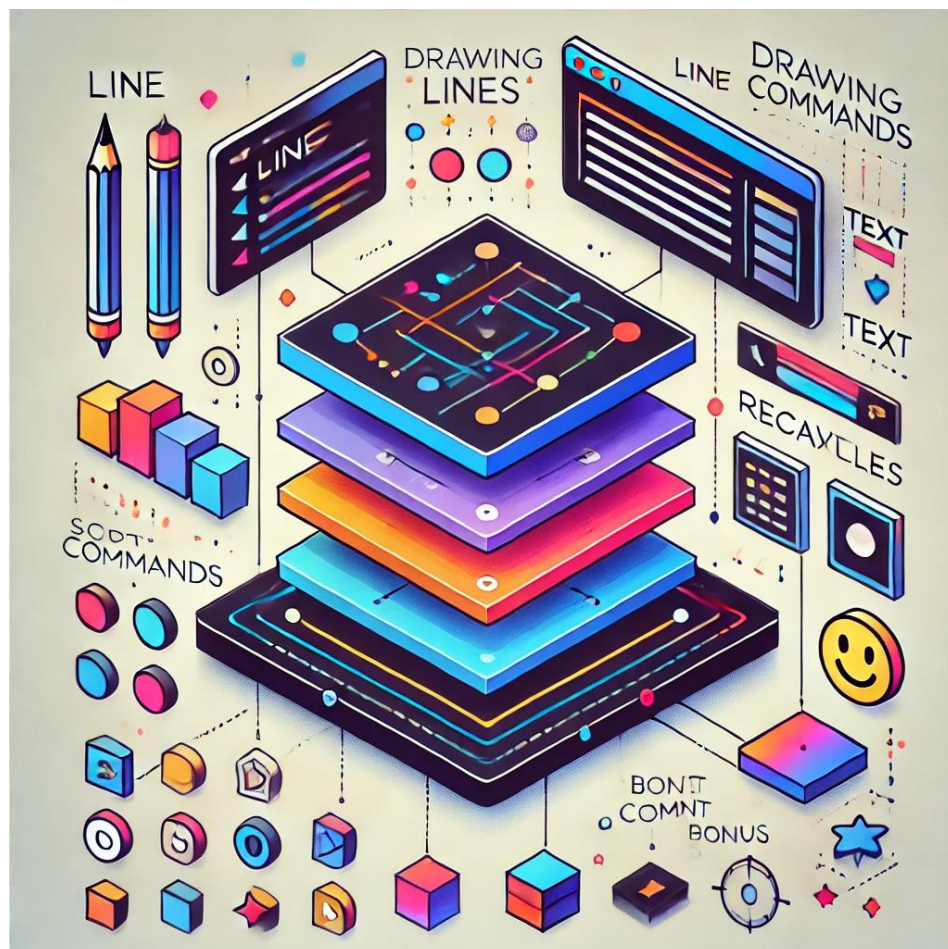
Niels Redegeld - 1827596

Chris van Wijk - 1834625

Electrical Engineering EV5A

Docent

Michiel Scager en Franc van der Bent



Het bestuur van de Stichting Hogeschool Utrecht te Utrecht aanvaardt geen enkele aansprakelijkheid voor schade voortvloeiende uit het gebruik van enig gegeven, hulpmiddel, werkwijze of procedure in dit verslag beschreven. Vermenigvuldiging zonder toestemming van de auteur(s), de school of het bedrijf (indien van toepassing) is niet toegestaan.

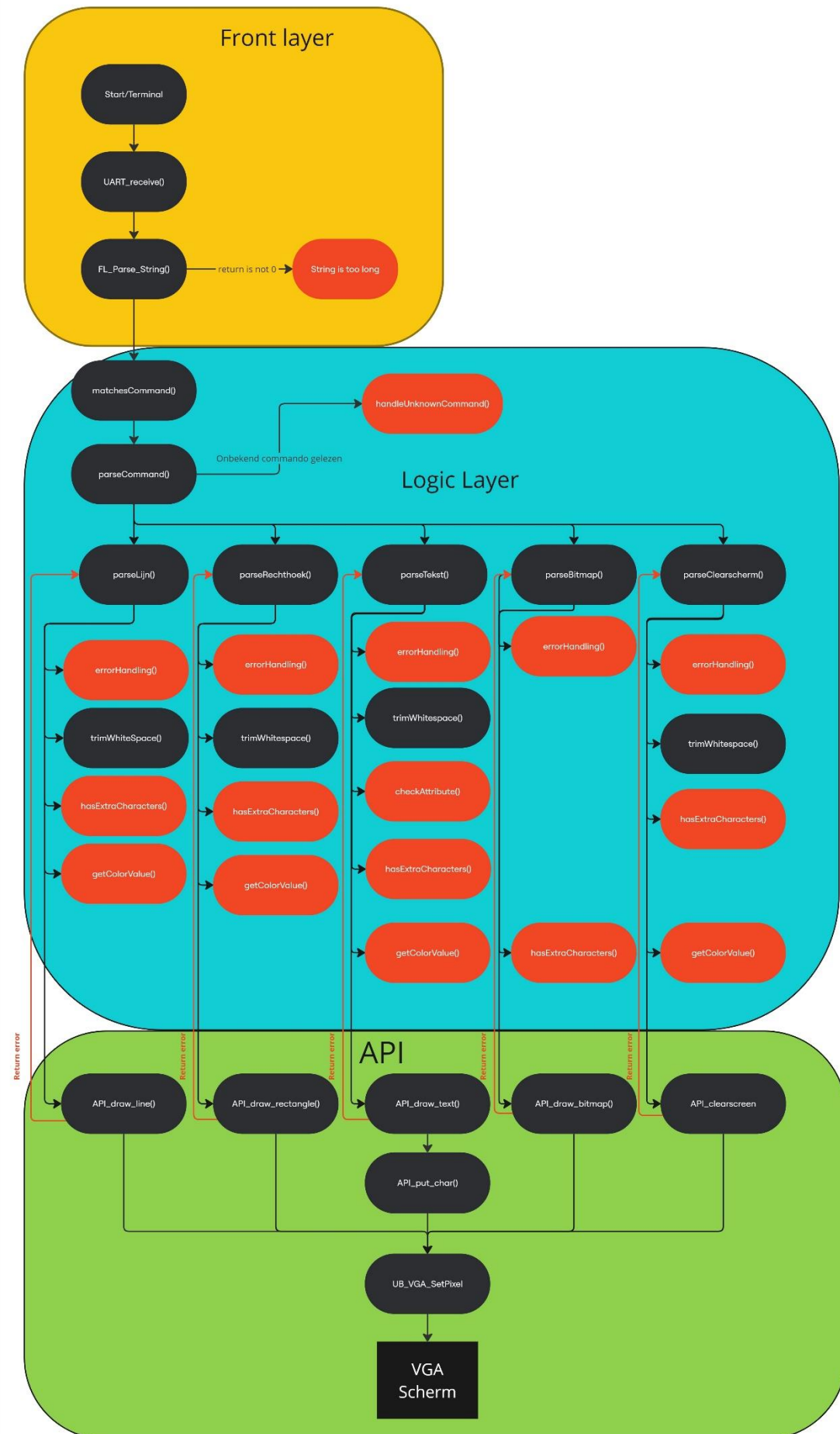
1 Graphical design

Het onderstaande low-level design/graphical design beschrijft de software opbouw verdeeld over drie lagen: de Front Layer, de Logic Layer, en de API Layer. De communicatie en de verbindingen tussen de verschillende functies zijn in dit plaatje weergegeven.

- **Front Layer (Geel):** Deze layer ontvangt invoer via de UART-terminal en verwerkt deze via functies zoals `UART_receive()` en `FL_Parse_String()`. Wanneer de invoer niet voldoet aan de verwachte voorwaarden (bijvoorbeeld te lange strings), wordt een foutmelding via UART teruggestuurd.
- **Logic Layer (Blauw):** In deze layer wordt de ontvangen string verder geanalyseerd en verwerkt door `parseCommand()`. Specifieke parsers zoals `parseLine()`, `parseRectangle()` etc. verwerken de gegevens afhankelijk van de commando's. Hiermee wordt een functie in de API layer aangeroepen met de opgegeven waardes. Fouten zoals extra karakters of ongeldige tekens worden afgehandeld via functies zoals `errorHandling()`. Ook hier worden de fouten zichtbaar gemaakt in de terminal doormiddel van UART.
- **API Layer (Groen):** Deze layer ontvangt de opdrachten van de logic layer en zet dit om naar grafisch beeld. Functies zoals `API_draw_line()` en `API_draw_rectangle()` maken gebruik van low-level operaties zoals `UB_VGA_SetPixel` om de pixels op het scherm te laten zien. Ook checkt deze of alle waardes die hij ontvangt correct zijn en returned een error naar de Logic Layer als dit nodig is.

Note

De rode blokken in het diagram markeren foutafhandelingspaden. Als er een fout optreedt (bijvoorbeeld een onherkenbaar commando of een patroonfout), wordt een foutmelding via UART teruggestuurd naar de gebruiker. Dit zorgt voor robuuste foutdetectie en gebruikersfeedback.



2 Drie lagen model

In het onderstaande plaatje ziet U het drie lagen model dat in onze software is geïmplementeerd. Deze is ontworpen om de functionaliteit modulair en overzichtelijk te maken. Het model bestaat uit drie lagen: de Front layer, de Logic layer, en de API layer. Elk van deze lagen heeft zijn eigen functie binnen dit systeem. Hieronder wordt een overzicht gegeven van de lagen en hun functies:

2.1 Front Layer

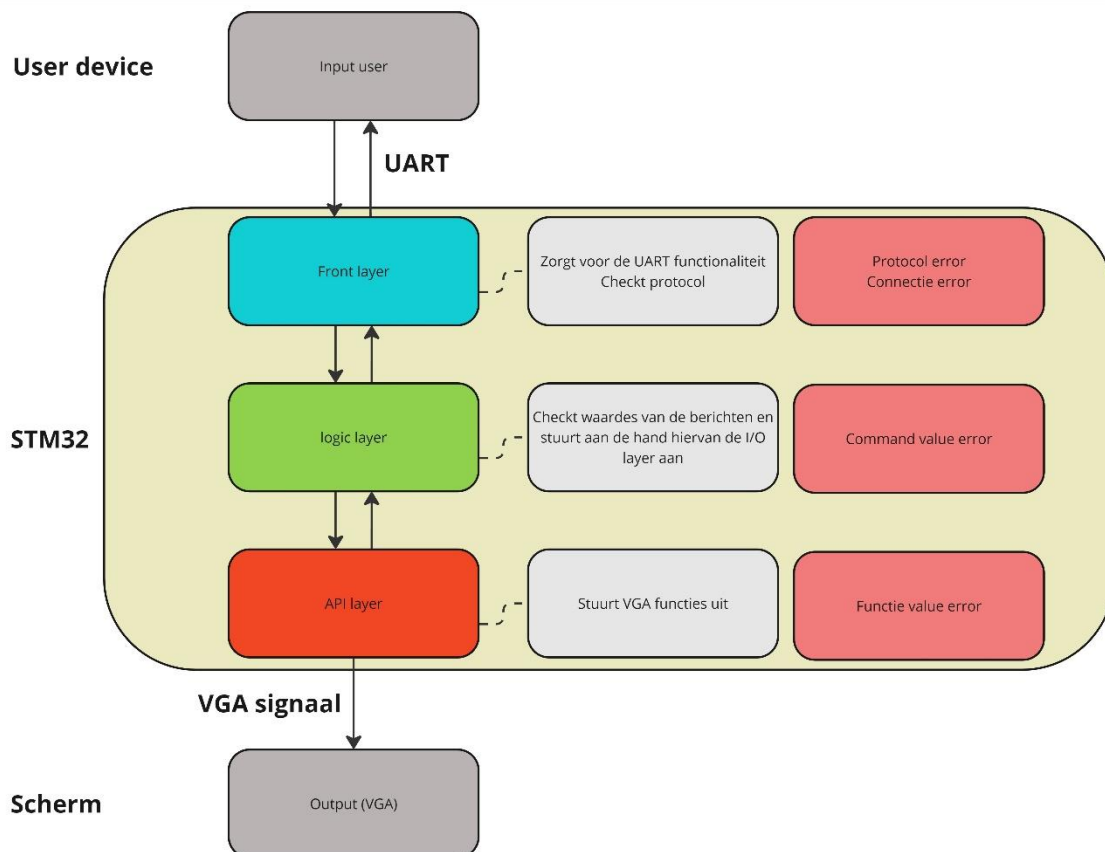
De front layer wordt gebruikt om de communicatie tussen de gebruiker en het systeem te verzorgen. Deze layer bestaat voornamelijk uit het verwerken van UART-communicatie. Dit houdt ook in het controleren van het protocol en het teruggeven van fouten aan de gebruiker.

2.2 Logic Layer

De Logic Layer is de layer die alles samenknoot. Deze layer kijkt wat er precies in de berichten staat die binnenkomen en bepaalt op basis van deze informatie welke functies moeten worden aangeroepen. Dit kan in de API Layer zijn maar ook weer in de Front Layer. Als de gegevens correct zijn, gaan ze naar de API Layer. Als de gegevens fout zijn, gaat er een foutmelding terug naar de Front Layer, die dit vervolgens communiceert naar de gebruiker. Hiermee zorgt de Logic Layer voor de juiste verwerking van de data.

2.3 API Layer

De API Layer zorgt voor de aansturing richting het scherm en past de VGA-buffer aan, waardoor er beeld op het scherm verschijnt. Dit wordt gedaan aan de hand van de instructies van de Logic Layer. Ook hier wordt nog gecontroleerd of er geen errors in de aangeroepen data zit zodat er nooit een fout commando aangeroepen wordt. Als hier wel een fout in zit, wordt er een foutcode naar de Logic Layer gecommuniceerd. Hiermee vormt de API Layer de laatste Layer voordat er beeld op het scherm komt.



3 Commando's

Hier onder zal aangegeven zijn welke commando's gebruikt kunnen worden en hoe deze exact functioneren.

3.1 Lijn

Commando: lijn, x1, y1, x2, y2, kleur, dikte

Met dit commando kan een lijn tussen twee punten op het scherm worden getekend.

- **x1, y1:** De coördinaten van het beginpunt van de lijn.
- **x2, y2:** De coördinaten van het eindpunt van de lijn.
- **Kleur:** De kleur van de lijn
- **Dikte:** De dikte van de lijn in pixels.

Voorbeeld: lijn, 10, 20, 100, 200, rood, 2

Dit tekent een rode lijn van (10, 20) naar (100, 200) met een dikte van 2 pixels.

3.2 Rechthoek

Commando: rechthoek, x_lup, y_lup, breedte, hoogte, kleur, gevuld (1,0)

Met dit commando kan een rechthoek op het scherm worden getekend.

- **x_lup, y_lup:** De coördinaten van de linkerbovenhoek van de rechthoek
- **Breedte:** De breedte van de rechthoek.
- **Hoogte:** De hoogte van de rechthoek.
- **Kleur:** De kleur van de rechthoek.
- **Gevuld:** Geef aan of de rechthoek gevuld moet zijn (1 = gevuld, 0 = niet gevuld).

Voorbeeld: rechthoek, 50, 50, 100, 200, groen, 1

Dit tekent een groene, gevulde rechthoek met de linkerbovenhoek op (50, 50), een breedte van 100 pixels en een hoogte van 200 pixels.

3.3 Tekst

Commando: tekst, x, y, kleur, tekst, fontnaam, fontgrootte, fontstijl

Met dit commando kun je tekst op het scherm tonen.

- **x, y:** De coördinaten van het beginpunt van de tekst (linksboven).
- **Kleur:** De kleur van de tekst.
- **Tekst:** De string die wordt weergegeven.
- **Fontnaam:** De naam van het lettertype (bijvoorbeeld Arial, Consolas, Comic Sans).
- **Fontgrootte:** De grootte van het lettertype (bijvoorbeeld 1 voor klein, 2 voor groot).
- **Fontstijl:** De stijl van het lettertype (normaal, vet, cursief).

Voorbeeld: tekst, 20, 50, blauw, "Hallo wereld", Arial, 2, vet

Dit toont de tekst "Hallo wereld" in blauwe kleur, Arial lettertype, grootte 2, en vetgedrukt op positie (20, 50).

3.4 Bitmap

Commando: bitmap, nr, x-lup, y-lup

Met dit commando kun je een bitmap-afbeelding op het scherm weergeven.

- **nr:** Het nummer van de bitmap (verwijst naar een van de vooraf vastgelegde bitmaps).
- **x-lup, y-lup:** De coördinaten van de linkerbovenhoek van de bitmap.

Voorbeeld: bitmap, 1, 100, 100

Dit plaatst bitmap nummer 1 (een boze smiley) met de linkerbovenhoek op (100, 100).

3.5 Clearschermb

Commando: clearschermb, kleur

Dit commando wist het hele scherm en vult het met een opgegeven kleur.

- **Kleur:** De kleur waarmee het scherm wordt gevuld.

Voorbeeld: clearschermb, wit

Dit maakt het scherm volledig wit.