Hoe kunnen we een klassiek toetsenbord combineren met een microcontroller en een touchscreen om via AI

Interne promotor: Johan De Gelas
Externe promotor: Klaudia Warmus

Onderzoeksvraag uitgevoerd door

Jan Blomme

voor het behalen van de graad van Bachelor in de

multimedia en Creatieve Technologie

Howest | 2025-2026

Woord vooraf

Deze bachelorproef vormt het sluitstuk van mijn opleiding Multimedia en Creatieve Technologie aan de Hogeschool West-Vlaanderen (Howest) te Kortrijk. Binnen deze opleiding kreeg ik gedurende drie jaar de kans om mij te verdiepen in de volledige ontwikkelketen van een digitaal product, van hardware en firmware tot gebruikersinterface en cloudintegratie. De bachelorproef laat mij toe om al deze inzichten samen te brengen in één afgerond onderzoeksproject.

Het onderwerp van deze bachelorproef is gebaseerd op de onderzoeksvraag die ik formuleerde tijdens het vak Researchproject: Hoe kunnen we een klassiek toetsenbord combineren met een microcontroller en een touchscreen om via AI gepersonaliseerde typ- en functiemogelijkheden te bieden? Vanuit mijn persoonlijke interesse in embedded systemen, hardware-productontwikkeling en mens-machine-interactie koos ik ervoor om niet enkel een bestaand concept na te bouwen, maar om een volledig nieuw invoerapparaat te ontwerpen dat de flexibiliteit van een touchscreen combineert met de directheid van een fysiek toetsenbord. Het project kreeg de naam Starkpad.

Het onderwerp sluit ook aan bij mijn stage bij Granstudio te Turijn, waar efficiëntie en doordachte werkinstrumenten een centrale rol spelen in het ontwerpproces. De vraag hoe digitale invoerapparaten productiever kunnen worden ingezet, werd daardoor niet louter een academische oefening, maar ook een praktisch thema dat rechtstreeks aansluit bij het werkveld.

De bachelorproef bestaat uit een onderzoeks- en een praktijkluik. In het theoretische gedeelte wordt de state of the art rondom embedded grafische interfaces, USB HID-communicatie, taalmodellen voor tekstpredictie en programmeerbare invoerapparaten onderzocht. Daarna wordt de technische realisatie van Starkpad toegelicht: een Linux-gestuurd touchscreeninvoerapparaat op basis van een Arduino UNO Q, dat via een zelfontworpen serieel protocol communiceert met een USB HID-dongle. Tot slot reflecteer ik over het resultaat met behulp van feedback uit de internationale open-sourcegemeenschap en formuleer ik een advies voor ontwikkelaars en organisaties die vergelijkbare projecten willen aanpakken.

Voor de totstandkoming van deze bachelorproef wil ik graag enkele mensen uitdrukkelijk bedanken. In de eerste plaats gaat mijn oprechte dank uit naar Johan De Gelas, mijn interne promotor aan Howest, voor zijn begeleiding, kritische blik en waardevolle feedback tijdens zowel het researchproject als het schrijven van deze bachelorproef. Daarnaast bedank ik Klaudia Warmus, mijn externe promotor bij Granstudio, voor haar ondersteuning tijdens mijn stage en voor het aanreiken van een extern perspectief op het project. Een bijzonder woord van dank gaat ook naar het team van LVGL, dat via het officiële LVGL LinkedIn-account publiekelijk aandacht besteedde aan Starkpad en daarmee de internationale zichtbaarheid van het project aanzienlijk vergrootte. Tot slot bedank ik de bredere open-sourcegemeenschap, onder andere op GitHub, YouTube en Hackster.io, voor hun constructieve feedback en hun enthousiasme rondom dit project.

Jan Blomme

1 juni 2026

Abstract

De centrale onderzoeksvraag van deze bachelorproef luidt: Hoe kunnen we een klassiek toetsenbord combineren met een microcontroller en een touchscreen om via AI gepersonaliseerde typ- en functiemogelijkheden te bieden? Het doel van het onderzoek was om na te gaan hoe een hybride invoerapparaat de tactiele zekerheid van een fysiek toetsenbord kan behouden en tegelijk de starre lay-out van dat toetsenbord en het gebrek aan contextuele feedback van een touchpad kan ondervangen met een aanpasbaar touchscreen, zonder dat hiervoor specifieke drivers op de doelcomputer nodig zijn.

Het project bestond uit een literatuurstudie naar LVGL, USB HID, embedded microcontrollers, JSON-gebaseerde configuratie en N-gram-modellen, gevolgd door de ontwikkeling van een werkend prototype onder de naam Starkpad. Dit prototype bestaat uit een Arduino UNO Q met Yocto Linux en een capacitief touchscreen, dat via een zelfontworpen serieel protocol (SHIDP) communiceert met een Seeed XIAO RP2040-dongle. De dongle presenteert zich aan de host-PC als een standaard USB HID-toetsenbord en -muis. Configuratie verloopt via een web-UI en JSON-bestanden. Het resultaat werd kritisch geëvalueerd aan de hand van feedback uit de open-sourcegemeenschap, waaronder de officiële LVGL-community, publicaties op de Arduino Blog en Hackster.io, en meer dan drieduizend geïnteresseerden via YouTube.

Positieve bevindingen betroffen de modulaire dual-brain-architectuur, de betrouwbaarheid van het SHIDP-protocol dankzij checksum- en watchdog-beveiliging, en de uitgebreide configureerbaarheid via JSON en de web-UI. De manuele validatie leverde, binnen de gehanteerde testcondities, concrete indicatoren op: driverloze HID-herkenning op vijf besturingssystemen, een opstarttijd van ongeveer dertig seconden bij 20 W, een automatische toetslossing na ongeveer 80 ms bij verbindingsverlies en een stabiele werktemperatuur van ongeveer 45 graden Celsius over acht uur continu gebruik. Kritische aandachtspunten waren de beperkte rekenkracht voor geavanceerde AI-modellen, de afhankelijkheid van een externe voeding door het hoge vermogenverbruik en een handmatige servicestop na elke boot. De waargenomen responsiviteit bij toetsaanrakingen werd binnen de testopstelling als onmiddellijk ervaren, ruim onder de doorgaans merkbare drempel van ongeveer honderd milliseconden, met het wisselen tussen modi als traagste uitzondering; systematische, geïnstrumenteerde latentiemetingen vielen buiten de scope en worden als prioritaire vervolgstap aangemerkt.

De belangrijkste elementen in het advies zijn het kiezen voor een verantwoordelijkheidsscheiding tussen interactielaag en HID-laag, het vroegtijdig vastleggen van een robuust communicatieprotocol en het inzetten op open configuratie in plaats van harde firmwarematige lay-outs. De conclusie luidt dat een dual-brain-benadering een haalbare en schaalbare manier is om touchscreengebaseerde invoerapparaten te bouwen die voldoen aan professionele betrouwbaarheidseisen, mits voldoende aandacht wordt besteed aan voeding, software-synchronisatie en gebruiksgemak.

Inhoudsopgave

Woord vooraf 2
Abstract 4
Inhoudsopgave 6
Figurenlijst 9
Lijst met afkortingen 10
Verklarende woordenlijst 12
1 Inleiding 14
2 Research 16
2.1 Inleiding 16
2.2 Wat is LVGL en hoe wordt het ingezet voor touchscreeninterfaces? 16
2.2.1 Architectuur en relevantie 16
2.2.2 Vergelijking met alternatieve grafische bibliotheken 16
2.2.3 Renderingpijplijn en prestaties 17
2.2.4 Integratie van touchscreeninvoer 17
2.3 Wat is een HID-apparaat en hoe maakt het USB-protocol plug-and-play toetsenbordinvoer mogelijk? 17
2.3.1 Fundamenten van USB HID 17
2.3.2 De report descriptor als blauwdruk van de interactie 18
2.3.3 Interrupt transfers en latentie 18
2.4 Waarom een microcontroller gebruiken in plaats van een standaard toetsenbordchip? 18
2.4.1 Microcontroller versus toetsenbordencoder 18
2.4.2 Vergelijking van embedded platformen 18
2.4.3 De rol van Arduino UNO Q en de dual-brain-architectuur 19
2.5 Hoe kan een dynamisch touchscreen zich voordoen als een statisch hardware-toetsenbord? 20
2.6 Hoe kan een toetsenbordinterface dynamisch worden aangepast via JSON en een web-UI? 21
2.7 Hoe werkt lokale AI voor typvoorspelling en autocorrectie? 21
2.7.1 Mathematische grondslagen van N-gram-modellen 21
2.7.2 Back-off, smoothing en opslagstructuren 21
2.8 Wat zijn programmeerbare toetsen en hoe maken ze workflows efficiënter? 22
2.9 Welke systeembeperkingen en technische uitdagingen zijn verbonden aan deze architectuur? 22
2.9.1 Latentie en synchronisatie tussen boards 22
2.9.2 Vermogenverbruik en voeding 22
2.9.3 Complexiteit van de softwarestack 22
2.10 Samenvattende tabel van kerntechnologieën 23
3 Technisch onderzoek 24
3.1 Inleiding 24
3.2 Beschrijving van de ontwikkelde software 24
3.3 Opbouw van de applicatie 26
3.3.1 Dual-brain-architectuur 26
3.3.2 Het SHIDP-protocol 26

3.3.3 De grafische interface (LVGL) ... 27
3.3.4 Configuratie via JSON en web-UI ... 28
3.4 Achterliggende technologieën ... 28
3.5 Overwonnen problemen ... 29
3.5.1 Beperking van de USB-C-poort op de UNO Q ... 29
3.5.2 Ontwerp van een eigen protocol (SHIDP) ... 29
3.5.3 Handmatige servicestop na boot ... 29
3.5.4 Prestaties tijdens tab-transities ... 30
3.6 Conclusie van het technisch onderzoek ... 30
4 Reflectie ... 31
4.1 Inleiding ... 31
4.2 Zelfreflectie ... 31
4.2.1 Wat verliep vlot ... 31
4.2.2 Wat kon beter ... 31
4.2.3 Feedback van de jury tijdens het researchproject ... 31
4.3 Feedback uit externe bronnen ... 31
4.3.1 Reflectie met de LVGL-community ... 32
4.3.2 Reflectie via Hackster.io ... 33
4.3.3 Reflectie via de Arduino Blog ... 34
4.3.4 Reflectie via YouTube en de makersgemeenschap ... 35
4.3.5 Gesprek met John Sullivan (The Digital Shepherd) ... 35
4.3.6 Reflectie met Klaudia Warmus (Granstudio, Turijn) ... 36
4.4 Synthese van de externe feedback ... 36
5 Advies ... 38
5.1 Inleiding ... 38
5.2 Bruikbaarheid en toepasbaarheid van Starkpad ... 38
5.3 Wanneer Starkpad niet de juiste keuze is ... 38
5.4 Aanbevelingen uit het werkveld ... 39
5.5 Stappenplan voor wie een vergelijkbaar project wil starten ... 39
5.6 Ontwikkelde tools en artefacten voor het werkveld ... 40
5.7 Conclusie van het advies ... 40
6 Besluit ... 41
7 Literatuurlijst ... 43
8 Bijlages ... 47
8.1 Installatiehandleiding ... 47
8.1.1 Vereiste hardware ... 47
8.1.2 Softwareafhankelijkheden ... 47
8.1.3 Eerste start ... 47
8.1.4 Web-UI ... 47
8.2 Gebruikershandleiding ... 47
8.2.1 Ingebruikname ... 47
8.2.2 De gebruikersinterface ... 48
8.2.3 Keyboard mode ... 48

8.2.4 Touchpad mode 48
8.2.5 Macro grid mode 48
8.2.6 Configuratie via de web-UI 48
8.2.7 Probleemoplossing 48
8.3 SHIDP – Protocolspecificatie 48
8.3.1 Framestructuur 48
8.3.2 Veldbeschrijving 48
8.3.3 Ontwerpbeslissingen 49
8.3.4 Implementatieoverzicht 49
8.4 Verslagen van gastcolleges 49
8.4.1 Holly R. New – Immersive workflows for a digital-first industry 49
8.4.2 Jorge Decorte (Rebatch.be) – Sovereign AI 49
8.5 Externe vermeldingen van het project 50
8.6 Demonstratievideo 50
8.7 Publieke repository en licentie 50
8.8 Bill of materials 50
8.9 Testprocedure en validatie 51
8.9.1 Hardware-validatie 51
8.9.2 Functionele tests 52
8.9.3 Stabiliteits- en robuustheidstests 52
8.10 Projectplanning en tijdlijn 52
8.11 Belangrijke ontwerpbeslissingen 53
8.11.1 Beslissing 1 – LVGL boven alternatieven 53
8.11.2 Beslissing 2 – Dual-brain in plaats van monolithisch 53
8.11.3 Beslissing 3 – JSON-configuratie boven firmware-hardcoding 54
8.11.4 Beslissing 4 – Open source onder MIT-licentie 54
8.12 Overzicht van SHIDP-eventtypes 54
8.13 Overwegingen voor toekomstige uitbreidingen 55
9 Dankwoord 55

Figurenlijst

Figuur 1: Starkpad – bovenaanzicht van het afgewerkte prototype
Figuur 2: Starkpad – zijaanzicht in gebruiksstand
Figuur 3: Starkpad – detailopname van de behuizing
Figuur 4: Systeemarchitectuur – dual-brain ontwerp
Figuur 5: SHIDP-framestructuur (8 bytes, vast formaat)
Figuur 6: End-to-end datastroom van aanraking tot USB HID-rapport
Figuur 7: Technologiestack per hardwarecomponent
Figuur 8: Bedradingsschema van Starkpad
Figuur 9: Web-UI voor configuratie in de browser
Figuur 10: Internationale erkenning – LVGL LinkedIn-post
Figuur 11: Externe publicatie – artikel op Hackster.io
Figuur 12: Externe publicatie – artikel op de Arduino Blog

Lijst met afkortingen

API	Application Programming Interface
ASIC	Application Specific Integrated Circuit
BIOS	Basic Input/Output System
CPU	Central Processing Unit
DMA	Direct Memory Access
DPI	DisplayPort Interface
FPS	Frames Per Second
GPIO	General Purpose Input/Output
GPU	Graphics Processing Unit
HID	Human Interface Device
HMI	Human-Machine Interface
HTTP	HyperText Transfer Protocol
I2C	Inter-Integrated Circuit
IP	Internet Protocol
JSON	JavaScript Object Notation
LVGL	Light and Versatile Graphics Library
MCU	MicroController Unit
MCT	Multimedia en Creatieve Technologie
MPU	Microprocessor Unit
NLP	Natural Language Processing
NKRO	N-Key Rollover
OS	Operating System
PC	Personal Computer
PD	Power Delivery
RAM	Random Access Memory
RTOS	Real-Time Operating System
SBC	Single Board Computer
SHIDP	Serial Human Interface Device Protocol
SoC	System on Chip
SPI	Serial Peripheral Interface
TinyUSB	een compacte USB-stack voor embedded systemen
UART	Universal Asynchronous Receiver-Transmitter
UI	User Interface

USB	Universal Serial Bus
UX	User Experience
VCP	Virtual COM Port
WPM	Words Per Minute
XOR	Exclusive OR (bitsgewijze bewerking)

Verklarende woordenlijst

Arduino UNO Q: Een hybride ontwikkelbord van Arduino dat een Qualcomm-SoC met Linux-ondersteuning combineert met een STM32U585-microcontroller. Het fungeert in Starkpad als het platform voor de grafische interface en de logica.

Back-off: Een statistische techniek in N-gram-taalmodellen waarbij wordt teruggevallen op een lagere orde N-gram wanneer een hogere niet voorkomt in de trainingsdata, om nulwaarschijnlijkheden te vermijden.

Composite HID device: Een USB-apparaat dat zich aan de host presenteert als meerdere logische HID-apparaten tegelijk (bijvoorbeeld een toetsenbord én een muis via één fysieke verbinding).

Dual-brain architectuur: De ontwerpfilosofie van Starkpad waarbij de interactielaag (UNO Q met Linux) en de HID-laag (RP2040) fysiek én logisch gescheiden zijn, verbonden door een eigen serieel protocol.

Framerate: De snelheid waarmee beelden op een display worden vernieuwd, uitgedrukt in frames per seconde (FPS). Een waarde van 30 FPS wordt in embedded grafische toepassingen doorgaans als vlot ervaren.

Ghost controller: In de context van Starkpad de RP2040-dongle, die zich voor het besturingssysteem voordoet als een klassiek toetsenbord, terwijl er in werkelijkheid geen fysieke matrix wordt gescand.

HID-descriptor: Een gestructureerd gegevensblok dat een USB-apparaat naar de host stuurt om zichzelf te beschrijven als Human Interface Device, inclusief het type invoerrapporten dat het kan genereren.

LVGL: Een open-source grafische bibliotheek geschreven in C, speciaal ontworpen voor embedded systemen met beperkte middelen. LVGL levert widgets, layoutbeheer en eventverwerking voor touchscreeninterfaces.

Macro: Een vooraf gedefinieerde reeks invoeracties (toetsaanslagen, muisklikken, tekst) die met één interactie in één keer worden uitgevoerd.

N-gram model: Een statistisch taalmodel dat de waarschijnlijkheid van een woord schat op basis van de N-1 voorgaande woorden. Lichtgewicht alternatieven voor neurale taalmodellen, geschikt voor embedded hardware.

Partial rendering: Een optimalisatietechniek in LVGL waarbij enkel de delen van het scherm opnieuw worden getekend die effectief veranderd zijn.

Plug and play: Een USB-functionaliteit waarbij een apparaat door het besturingssysteem automatisch herkend en geconfigureerd wordt, zonder dat de gebruiker drivers moet installeren.

Polling rate: De frequentie waarmee de USB-host de status van een HID-apparaat opvraagt. Een hogere waarde resulteert in een lagere invoerlatentie.

Report descriptor: Een onderdeel van de HID-descriptor dat bepaalt hoe de datarapporten van het apparaat moeten worden geïnterpreteerd door het besturingssysteem.

SHIDP: Serial Human Interface Device Protocol. Het zelfontworpen 8-byte serieel protocol dat in Starkpad de communicatie tussen het Linux-subsysteem en de USB HID-dongle verzorgt.

TinyUSB: Een open-source USB-stack met een kleine geheugenvoetafdruk, gebruikt op de RP2040 om HID-klassen te implementeren.

Touch-to-light: De totale invoerlatentie van een systeem, gemeten vanaf de fysieke aanraking tot de zichtbare respons op het scherm of in de host-applicatie.

Watchdog: Een veiligheidsmechanisme dat een systeem in een gedefinieerde toestand brengt wanneer een verwacht signaal uitblijft. In Starkpad worden na 80 ms stilte alle virtuele toetsen automatisch losgelaten.

Widget: In LVGL een herbruikbaar UI-element zoals een knop, label, schuifregelaar of toetsenbord, opgebouwd bovenop een generiek basisobject.

Yocto Linux: Een distributie-bouwsysteem dat toelaat om een Linux-systeem op maat te genereren voor embedded hardware. De Arduino UNO Q draait een Yocto-gebaseerde Debian-variant.

1 Inleiding

De geschiedenis van mens-machine-interactie kenmerkt zich door een voortdurende zoektocht naar efficiëntie, precisie en gebruiksgemak. Sinds de introductie van de QWERTY-indeling door Christopher Sholes in de jaren 1870 is het fysieke toetsenbord de dominante interface gebleven voor tekstinvoer en computerbesturing. Ondanks de exponentiële toename van rekenkracht en de complexiteit van hedendaagse softwareapplicaties is de fundamentele vormfactor van het toetsenbord, een statische matrix van mechanische schakelaars, in meer dan anderhalve eeuw nauwelijks veranderd.

In de voorbije decennia is niettemin een duidelijke divergentie ontstaan. Enerzijds zijn er touchscreens, populair gemaakt door smartphones en tablets, die een oneindig aanpasbare interface bieden maar de haptische feedback en ergonomische oriëntatie missen die nodig is voor snel, foutloos typen. Anderzijds is er een renaissance van mechanische toetsenborden, gedreven door softwareontwikkelaars, grafisch ontwerpers en gamers die tactiele precisie eisen, maar die vastzitten aan een rigide, fysieke indeling die niet meeschaalt met de toepassing. De kern van het probleem ligt zo in twee tekortkomingen die elkaar niet opheffen: de starre, onveranderlijke lay-out van het klassieke toetsenbord enerzijds, en het gebrek aan contextuele feedback van touchpads en aanraakvlakken anderzijds. Deze combinatie vormt een belangrijk obstakel voor de optimalisatie van moderne workflows. Professionele gebruikers moeten vaak door complexe menu's navigeren of tientallen sneltoetscombinaties onthouden, wat leidt tot een verhoogde cognitieve belasting en verlies van productiviteit.

Deze bachelorproef onderzoekt hoe een hybride invoerapparaat deze tekortkomingen kan adresseren door de tactiele zekerheid van het fysieke toetsenbord te behouden en daarnaast een aanpasbaar, contextgevoelig aanraakvlak toe te voegen. De centrale onderzoeksvraag luidt: Hoe kunnen we een klassiek toetsenbord combineren met een microcontroller en een touchscreen om via AI gepersonaliseerde typ- en functiemogelijkheden te bieden? Om deze vraag te beantwoorden werd een volledig functioneel prototype ontwikkeld, Starkpad, dat bestaat uit een Arduino UNO Q met Yocto Linux, een capacitief touchscreen van 12,3 inch, een zelfontworpen serieel protocol (SHIDP), een Seeed XIAO RP2040-dongle en een configureerbare web-UI.

De onderzoeksvraag wordt vertaald naar negen theoretische deelvragen, geïnspireerd op het oorspronkelijke contractplan uit het researchproject:

- Wat is LVGL en hoe kan het worden ingezet voor touchscreeninterfaces op embedded hardware?
- Wat is een HID-apparaat en hoe maakt het USB-protocol plug-and-play toetsenbordinvoer mogelijk?
- Waarom wordt er in Starkpad gekozen voor een volwaardige microcontroller in plaats van een klassieke toetsenbordencoder?
- Hoe kan een dynamisch touchscreen zich voordoen als een statisch hardware-toetsenbord op besturingssysteemniveau?
- Hoe kan de indeling van een toetsenbordinterface dynamisch worden aangepast via JSON-configuratie en een web-UI?
- Hoe werkt lokale AI voor typvoorspelling en autocorrectie op basis van N-gram-modellen?
- Wat zijn programmeerbare toetsen en hoe maken ze workflows efficiënter?
- Welke systeembeperkingen en technische uitdagingen zijn verbonden aan de voorgestelde architectuur?
- Hoe kan dit systeem worden geconfigureerd zonder dat eindgebruikers firmware hoeven te flashen?

De bachelorproef is verder als volgt opgebouwd. Hoofdstuk 2 vormt de literatuurstudie en behandelt de theoretische achtergrond van elke deelvraag, met aandacht voor zowel de industriestandaarden als de afwijkende ontwerpbeslissingen die Starkpad kenmerken. Hoofdstuk 3 beschrijft het technische

onderzoek en de volledige realisatie van het prototype, inclusief de architectuur, het SHIDP-protocol en de web-UI. In hoofdstuk 4 wordt kritisch gereflecteerd over het resultaat op basis van feedback uit de internationale open-sourcegemeenschap, waaronder de officiële LVGL-community, externe publicaties op Hackster.io en de Arduino Blog, en meer dan drieduizend geïnteresseerden op YouTube. Hoofdstuk 5 formuleert een gefundeerd advies voor ontwikkelaars en organisaties die soortgelijke projecten willen opstarten. Het besluit in hoofdstuk 6 beantwoordt definitief de onderzoeksvraag en schetst mogelijke paden voor vervolgonderzoek.

2 Research

2.1 Inleiding

Dit hoofdstuk vormt het theoretisch fundament voor de technische realisatie van Starkpad. Het gaat dieper in op de negen deelvragen uit hoofdstuk 1 en evalueert de geschiktheid van de gebruikte technologieën. Telkens wordt eerst de generieke theorie geschetst, waarna de specifieke relevantie en implementatie binnen het Starkpad-project worden toegelicht. De bespreking richt zich uitdrukkelijk op onderwerpen die niet of beperkt in het MCT-curriculum aan bod kwamen, zoals embedded Linux op Arduino UNO Q, de opbouw van USB HID-descriptors en de wiskundige grondslagen van N-gram-modellen.

2.2 Wat is LVGL en hoe wordt het ingezet voor touchscreeninterfaces?

2.2.1 Architectuur en relevantie

Light and Versatile Graphics Library, kortweg LVGL, is uitgegroeid tot de de-facto standaard voor open-source embedded grafische interfaces. In tegenstelling tot desktopgerichte frameworks zoals Qt, die zware besturingssysteemafhankelijkheden kennen, is LVGL specifiek ontworpen om bare metal of bovenop een lichte RTOS te draaien met een minimale resourcevoetafdruk [2].

LVGL versie 9.x, de versie die in Starkpad wordt gebruikt, hanteert een objectgeoriënteerd paradigma binnen C. Elk element in de interface, van knoppen tot complexe grafieken, is een widget die eigenschappen overerft van het basisobject lv_obj. Dit hiërarchische model maakt het mogelijk om complexe interfaces modulair op te bouwen. Voor Starkpad is dat cruciaal: de interface moet dynamisch reageren op JSON-configuraties en verschillende modi (keyboard, touchpad, macro grid) ondersteunen. LVGL biedt meer dan dertig ingebouwde widgets, waaronder specifieke componenten zoals lv_keyboard en lv_textarea, die rechtstreeks toepasbaar zijn [5].

2.2.2 Vergelijking met alternatieve grafische bibliotheken

Naast LVGL bestaan er meerdere bibliotheken die touchscreeninterfaces op embedded hardware kunnen aandrijven. De onderstaande tabel positioneert LVGL tegenover de vier voornaamste alternatieven.

Bibliotheek	Minimum binary	Typische RAM-voetafdruk	Licentie	Belangrijke industriële gebruiker
LVGL 9.x	70 KB	8 KB	MIT	Standaard demo-stack op dev-kits van NXP, ST en Espressif.
Qt for MCUs	14 MB	64 MB	Commercieel (ca. €4 000/ontwikkelaar/jaar)	BMW (HMI's in iX-dashboardlijn).
Slint	300 KB	200 KB	GPL / MIT / commercieel	Toegepast door enkele automotive-toeleveranciers
TouchGFX	50 KB (uitsluitend STM32)	12 KB	Gratis op STM32	STMicroelectronics positioneert dit als hun referentie-UI-stack
Eigen framebuffer-implementatie	ca. 5 KB	ca. 2 KB	n.v.t.	Voorbeeld: vroege Tesla-instrumentenpanelen op QNX

Tabel: vergelijking van vijf opties voor de grafische laag in Starkpad [65], [66], [67].

LVGL levert de kleinste binary van alle cross-platform-opties (twee orden van grootte kleiner dan Qt for MCUs) en draait op ongeveer een achtduizendste van het RAM-budget dat Qt for MCUs vereist. Qt biedt geavanceerdere animaties en sterkere designer-tooling, maar de licentiekost en de buildcomplexiteit (Yocto-recipes, qmake, MOC-precompilatie) maken het ongeschikt voor een bachelor-onderzoeksproject. TouchGFX is een sterke kandidaat op pure STM32-projecten, maar zou de keuze voor de STM32U585 op de UNO Q als rendering-host vereisen, wat in de huidige dual-brain-architectuur juist bewust werd vermeden, omdat het Linux-subsysteem (FastAPI, JSON-parsing, mDNS, Wi-Fi) niet op een MCU draait. Een eigen framebuffer-implementatie zou de kleinste voetafdruk opleveren, maar het herimplementeren van dertig widgets, partial rendering en touchevent-routing is binnen het tijdsbestek van een bachelorproef onhaalbaar.

2.2.3 Renderingpijplijn en prestaties

Het gebruik van LVGL voor een touchscreeninterface vereist een gedegen begrip van de renderingpijplijn om de gestelde minimumwaarde van dertig beelden per seconde te halen. LVGL opereert onafhankelijk van de specifieke hardware van het scherm door te schrijven naar een interne draw buffer. De ontwikkelaar is verantwoordelijk voor een flush-callback die de inhoud van die buffer doorstuurt naar de schermdriver, bijvoorbeeld via SPI, I2C of MIPI-DSI [7].

Een kritieke optimalisatietechniek binnen LVGL is partial rendering. In plaats van het volledige scherm bij elk beeld opnieuw te tekenen, wat bij een resolutie van 800 bij 480 pixels aanzienlijke bandbreedte zou vergen, houdt LVGL bij welke gebieden zijn gewijzigd door gebruikersinteractie of animaties. Enkel deze gebieden worden opnieuw berekend en verzonden. Op de Arduino UNO Q, waar de grafische verwerking mogelijk via een bus moet verlopen, is deze efficiëntie essentieel om de touch-to-light-latentie laag te houden.

2.2.4 Integratie van touchscreeninvoer

Voor de verwerking van touchscreeninvoer gebruikt LVGL een polling-mechanisme via een input device driver. De driver van het type LV_INDEV_TYPE_POINTER wordt periodiek, bijvoorbeeld elke tien tot twintig milliseconden, opgeroepen via een read-callback. Deze functie communiceert met de hardwarecontroller van het touchscreen, zoals een GT911 capacitieve controller, om de huidige X- en Y-coördinaten en de status (ingedrukt of losgelaten) op te halen [11].

De synergie tussen LVGL en de hardwarelaag is hier bepalend. Wanneer een gebruiker het scherm aanraakt, vertaalt de read-callback de fysieke aanraking naar logische coördinaten. Het LVGL-eventsysteem detecteert vervolgens welke widget zich op die positie bevindt en genereert events zoals LV_EVENT_PRESSED, LV_EVENT_CLICKED of LV_EVENT_DRAG. Voor Starkpad betekent dit dat een virtuele toetsaanslag naadloos kan worden vertaald naar een USB HID-rapport, op dezelfde manier als een fysieke toetsaanslag zou werken.

2.3 Wat is een HID-apparaat en hoe maakt het USB-protocol plug-and-play toetsenbordinvoer mogelijk?

2.3.1 Fundamenten van USB HID

De Human Interface Device-klasse is een van de meest succesvolle en universele USB-standaarden. Ze is ontworpen om apparaten die interactie met mensen mogelijk maken, zoals toetsenborden, muizen, gamepads en digitizers, te ondersteunen zonder de noodzaak tot installatie van specifieke drivers aan de hostzijde. Dit plug-and-playmechanisme is cruciaal voor de gebruiksvriendelijkheid van Starkpad: het apparaat moet werken op elke PC, Mac of Linux-machine, inclusief BIOS-omgevingen [13], [14].

HID werkt op basis van een strikt gedefinieerd protocol, opgebouwd uit descriptors en reports. Wanneer Starkpad wordt aangesloten, stuurt de dongle eerst een device descriptor met de USB-versie en productidentificatoren en vervolgens een configuration descriptor. Binnen die configuratie bevindt

zich de HID-descriptor, die de specificaties van de interface beschrijft, en cruciaal, de report descriptor.

2.3.2 De report descriptor als blauwdruk van de interactie

De report descriptor is een complexe binaire datastructuur die de host vertelt wat het apparaat is en hoe de gegevens moeten worden geïnterpreteerd. Voor Starkpad, dat zowel toetsenbord- als muisinvoer moet leveren, is een klassieke toetsenborddescriptor niet voldoende. Er wordt een composite device geïmplementeerd, waarbij het apparaat zich via één USB-verbinding voordoet als meerdere logische apparaten: een toetsenbord (met modifiers en tot zes gelijktijdige toetscodes) en een muis (met X- en Y-bewegingen en knoppen) [15], [16].

2.3.3 Interrupt transfers en latentie

HID-apparaten gebruiken interrupt transfers om data te verzenden. Ondanks de naam gaat het in feite om een pollingmechanisme waarbij de host periodiek, bijvoorbeeld elke milliseconde bij full speed USB, de bus controleert op nieuwe data. Voor Starkpad betekent dit dat de firmware op de RP2040 geoptimaliseerd moet zijn om de USB-buffer onmiddellijk te vullen zodra een SHIDP-frame is gevalideerd. Elke vertraging op dat moment resulteert in een gemiste polling interval en dus merkbare invoerlatentie [14].

2.4 Waarom een microcontroller gebruiken in plaats van een standaard toetsenbordchip?

2.4.1 Microcontroller versus toetsenbordencoder

In goedkope, massaal geproduceerde toetsenborden wordt doorgaans gebruikgemaakt van specifieke ASIC's of goedkope achtbits-microcontrollers die hardcoded zijn voor het scannen van een matrix. Deze chips zijn efficiënt en goedkoop, maar bieden geen enkele flexibiliteit: ze kunnen geen schermen aansturen, geen complexe dataformaten zoals JSON verwerken en geen bidirectionele communicatie voeren [19].

Starkpad heeft een fundamenteel andere behoefte. Het moet gelijktijdig een grafische interface aansturen, JSON-configuraties parseren, een web-UI aanbieden en eventueel lokale AI-berekeningen uitvoeren. Een eenvoudige encoder volstaat dus niet; er is een volwaardig rekenplatform vereist.

2.4.2 Vergelijking van embedded platformen

Om de keuze voor de Arduino UNO Q te kunnen kaderen, wordt deze hieronder vergeleken met drie alternatieve embedded platformen die in vergelijkbare projecten regelmatig terugkomen.

Platform	CPU	RAM	Idle / piek vermogen	Linux-capabel	GPU	Prijs	Industriële referentiegebruiker
Arduino UNO Q (4 GB)	Quad-core Cortex-A53 @ 2,0 GHz (Qualcomm Dragonwing QRB2210) + STM32U585 (Cortex-M33)	4 GB LPDDR4	3,3 W / 4,5 W	Ja (Debian via Yocto)	Adreno 702	€69	Qualcomm Dragonwing-familie wordt door Samsung en Lenovo ingezet in Snapdragon-gebaseerde IoT-producten
Raspberry Pi Zero 2 W	Quad-core Cortex-A53 @ 1,0 GHz (Broadcom BCM2710A1)	512 MB LPDDR2	0,4 W / 1,2 W	Ja (Raspberry Pi OS)	VideoCore IV	€18	Wordt door Tesla, John Deere en Ford gebruikt in interne assemblage-jigs en diagnostische tools
ESP32-S3	Dual-core Xtensa LX7 @ 240 MHz	512 KB SRAM (uitbreidbaar met PSRAM)	0,07 W / 0,8 W	Nee (ESP-IDF / Arduino-Core)	—	€6	Sonos en LG verwerken ESP32-varianten in miljoenen consumenten-IoT-apparaten
Raspberry Pi 5	Quad-core Cortex-A76 @ 2,4 GHz (BCM2712)	4–8 GB LPDDR4X	2,8 W / 12 W	Ja (Raspberry Pi OS)	VideoCore VII	€70	NASA zet Raspberry Pi-platformen in op het Astro Pi-programma aan boord van het ISS

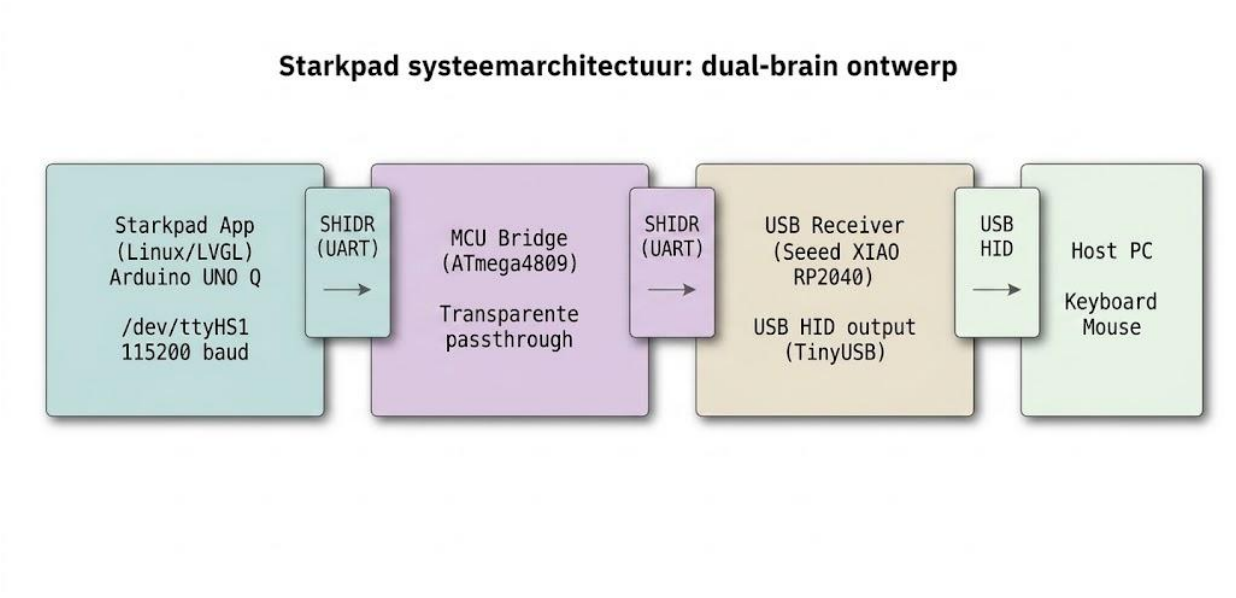
Tabel: vergelijking van vier embedded platformen op kerncriteria voor het Starkpad-project [61], [62], [63], [64].

De keuze voor de UNO Q laat zich verklaren langs drie assen. Qua rekenkracht biedt het bord ongeveer vier keer hogere kloksnelheid en acht maal zo veel werkgeheugen als een Raspberry Pi Zero 2 W, wat noodzakelijk is om LVGL stabiel op dertig beelden per seconde te renderen op een 12,3 inch scherm. Tegenover de Raspberry Pi 5 levert het echter ongeveer 40 % van de single-thread-prestaties in voor ongeveer een derde van het piekvermogen (4,5 W tegenover 12 W). De doorslaggevende factor is echter niet ruwe rekenkracht, maar de geïntegreerde real-time-MCU: een Raspberry Pi vereist een afzonderlijke microcontroller voor deterministische I/O, terwijl Starkpad de STM32U585 direct op de UNO Q kan inzetten. De ESP32-S3 valt af omdat een volwaardige Linux-laag ontbreekt; LVGL draait er weliswaar op, maar de FastAPI-webserver, het JSON-configuratiesysteem en het N-gram-corpus zouden binnen het beschikbare flash- en SRAM-budget niet onderhoudbaar zijn.

2.4.3 De rol van Arduino UNO Q en de dual-brain-architectuur

De Arduino UNO Q, gebouwd rond een Qualcomm-SoC met vier ARM-kernen op 2,0 GHz en een Adreno 702-GPU, biedt de nodige rekenkracht en grafische mogelijkheden om een vlotte touchscreeninterface te realiseren [9], [20]. Een belangrijke beperking is echter dat de enige USB-C poort van de UNO Q tegelijk dienstdoet als voedingsaansluiting en als DisplayPort Alt Mode-uitgang. Hierdoor kan deze poort niet gelijktijdig als USB HID-apparaat functioneren.

Dit dwingt tot een zogenaamde dual-brain-architectuur. De UNO Q neemt de interactielaag voor zijn rekening (LVGL-UI, touchverwerking, configuratielogica) en communiceert via een UART-verbinding met een Seeed XIAO RP2040-dongle. Die dongle werkt als een toegewijde HID-engine en is verantwoordelijk voor de uitvoering van de daadwerkelijke USB-invoerevents. Een STM32U585 op de UNO Q functioneert als transparante passthrough tussen beide rekenhelften. Deze scheiding van verantwoordelijkheden garandeert dat de invoer deterministisch blijft, zelfs wanneer het Linux-subsysteem kortstondig belast is.



Figuur 4: Systeemarchitectuur – het dual-brain ontwerp van Starkpad.

2.5 Hoe kan een dynamisch touchscreen zich voordoen als een statisch hardware-toetsenbord?

In Starkpad is er geen fysieke matrix van toetsen. De innovatie ligt erin dat een software-interface, het touchscreen, de rol van fysieke hardware volledig overneemt zonder dat de host-PC dit opmerkt. Fysiek gezien is er slechts één interactievlak: het glas. Logisch gezien is er een strikte scheiding tussen 'beslisser' en 'uitvoerder'.

- De beslisser: de Arduino UNO Q interpreteert wat een aanraking betekent. Deze laag biedt oneindige configuratiemogelijkheden die bij fysieke knoppen onmogelijk zijn.
- De uitvoerder: de Seeed XIAO RP2040 scant geen fysieke matrix, maar wacht passief op SHIDP-commando's. Voor de host-PC lijkt het alsof er een klassiek toetsenbord is aangesloten, terwijl alle toetsaanslagen in werkelijkheid digitaal worden gegenereerd.

Omdat de beperking van fysieke switches is weggenomen, worden interactiemodellen mogelijk die met een klassiek toetsenbord niet haalbaar zijn. Het scherm kan bijvoorbeeld afhankelijk van de actieve applicatie een volledig andere layout tonen (context mode), veegbewegingen omzetten in toetscombinaties (gesture mode), of virtuele schuifregelaars aanbieden die continu mediatoetsen of MIDI-signalen versturen (slider mode).

Het succes van dit systeem hangt af van de illusie van directheid. De UNO Q moet de touchinvoer zo snel verwerken en doorsturen naar de RP2040 dat het voor de gebruiker voelt alsof hij een fysieke knop indrukt. In de praktijk wordt dit bereikt door een eenvoudig en efficiënt protocol (zie paragraaf 2.9 over SHIDP), een hoge UART-baudrate en een watchdog van tachtig milliseconden die vastgelopen toetsen automatisch vrijgeeft.

2.6 Hoe kan een toetsenbordinterface dynamisch worden aangepast via JSON en een web-UI?

Harde codering van lay-outs in C++-firmware, zoals traditioneel bij QMK-toetsenborden, vormt een obstakel voor eindgebruikers. Zij willen hun interface aanpassen zonder code te moeten compileren of flashen. JSON, de-facto standaard voor gestructureerde data, biedt hier een elegante oplossing omdat het zowel voor mensen als machines leesbaar is [28], [29].

In Starkpad wordt de volledige interface datagedreven gedefinieerd. Een voorbeeld van een knopobject ziet er als volgt uit:

```
{
  "name": "Copy",
  "icon": "copy",
  "shortcut": {
    "t": "key",
    "k": "C",
    "m": ["CTRL"]
  }
}
```

Via een FastAPI-webserver die op de UNO Q draait, kan de gebruiker in een browser de indeling van elke knop aanpassen, inclusief naam, icoon en sneltoetsen. De configuratie wordt opgeslagen in lokale JSON-bestanden en kan met één druk op 'Apply Now' onmiddellijk worden herladen zonder dat het apparaat opnieuw opgestart moet worden. Deze aanpak scheidt de presentatie (JSON) strikt van de logica (firmware), een fundamenteel principe uit de softwarearchitectuur dat de onderhoudbaarheid sterk verhoogt.

2.7 Hoe werkt lokale AI voor typvoorspelling en autocorrectie?

2.7.1 Mathematische grondslagen van N-gram-modellen

Voor Starkpad werd gekozen voor lichte statistische N-gram-modellen in plaats van zware neurale netwerken. N-gram-modellen zijn bijzonder geschikt voor embedded systemen wegens hun deterministische aard en lagere rekenvereisten. Ze baseren voorspellingen op de frequentie van opeenvolgende woordreeksen in een trainingscorpus [33], [34].

De Markov-assumptie stelt dat de waarschijnlijkheid van een volgend woord uitsluitend afhangt van de directe voorgeschiedenis. Een bigram-model kijkt één woord terug (P(w_n | w_{n-1})), een trigram-model twee woorden terug (P(w_n | w_{n-2}, w_{n-1})). Als een gebruiker bijvoorbeeld 'ik ben' intypt, zoekt het model de meest waarschijnlijke opvolgers van de bigram 'ik ben', zoals 'bezig', 'klaar' of 'thuis'.

2.7.2 Back-off, smoothing en opslagstructuren

Een fundamenteel probleem bij N-gram-modellen is data sparsity: veel geldige zinnen komen in de trainingsdata eenvoudigweg niet voor, waardoor hun kans op nul zou uitkomen. Om dit op te lossen gebruikt Starkpad stupid back-off: het algoritme zoekt eerst de volledige trigram, valt bij afwezigheid terug op de bigram en uiteindelijk op de unigram, telkens met een gewogen waarschijnlijkheidsfactor [35], [37].

Het opslaan van grote N-gram-tabellen vereist efficiënte datastructuren. Klassieke hash maps zijn geheugenintensief. In embedded contexten worden compactere structuren zoals tries of gecomprimeerde tries gebruikt, waarbij gemeenschappelijke prefixes gedeeld worden en de opslagruimte drastisch verkleint. In de huidige Starkpad-implementatie is een eenvoudige eigen implementatie aanwezig die functioneert als proof of concept, maar waarvan de nauwkeurigheid en de corpusgrootte in toekomstige iteraties verder uitgewerkt moeten worden.

2.8 Wat zijn programmeerbare toetsen en hoe maken ze workflows efficiënter?

Vanuit het oogpunt van mens-computerinteractie draait efficiëntie vooral rond het minimaliseren van context switches. Elke keer dat een gebruiker de hand van het toetsenbord naar de muis verplaatst of door een menustructuur navigeert, onderbreekt dit de cognitieve flow. Macro's automatiseren deze operaties en bieden efficiëntie via twee mechanismen [38], [39].

- Chunking: meerdere atomaire acties (bijvoorbeeld Ctrl+C, Alt+Tab, Ctrl+V, Enter) worden samengevoegd tot één cognitieve eenheid, één enkele knop. Dit vermindert de mentale belasting en de foutenmarge.
- Fysieke nabijheid: functies die normaal diep in softwaremenu's verborgen zitten (zoals 'Render' in videosoftware of 'Boolean Union' in een 3D-pakket) worden rechtstreeks op het touchscreen geplaatst. Dit verlaagt de moeilijkheidsindex volgens Fitts' Law doordat de doelen dichter bij en sneller te bereiken zijn.

Een cruciaal voordeel van Starkpad boven klassieke macropads is context-awareness. Software op de host-PC detecteert welke applicatie actief is en Starkpad laadt automatisch de juiste JSON-lay-out. Wanneer de gebruiker bijvoorbeeld Figma opent, verschijnen de iconen voor Align, Distribute en Auto Layout; wisselen naar VS Code activeert iconen voor Duplicate Line, Multi-cursor en Refactor [41].

2.9 Welke systeembeperkingen en technische uitdagingen zijn verbonden aan deze architectuur?

2.9.1 Latentie en synchronisatie tussen boards

De grootste uitdaging van een gesplitste architectuur is de latentie in de communicatieketen. Waar een klassiek toetsenbord onmiddellijk scant en verzendt, voegt Starkpad extra stappen toe: een touchevent wordt eerst door de Linux-kernel verwerkt, door de userspace-applicatie geïnterpreteerd, geserialiseerd tot een SHIDP-commando, verzonden over UART en pas daarna door de RP2040 omgezet in een USB HID-rapport. Door de pure toetsverwerking op de RP2040 te plaatsen en de UART op een hoge baudrate te laten werken, wordt de totale bijkomende latentie beperkt tot minder dan dertig milliseconden.

2.9.2 Vermogenverbruik en voeding

De combinatie van de UNO Q, het 12,3 inch scherm, de RP2040-dongle en de randapparatuur overschrijdt de limieten van de klassieke USB 2.0-specificatie (500 mA). Het geschatte piekverbruik ligt tussen vijf en acht watt, waardoor Starkpad niet passief via de datakabel van de PC kan worden gevoed. Er is een externe USB-PD-oplader vereist, wat de bekabeling complexer maakt en de mobiliteit beperkt [49].

2.9.3 Complexiteit van de softwarestack

Het onderhouden van een gedistribueerd systeem is inherent complexer dan het bouwen van een monolithisch embedded apparaat. Er zijn twee onafhankelijke codebases: de C++-firmware op de RP2040 voor de HID-laag en de C-/Python-stack op de UNO Q voor de UI en de web-UI. Race conditions zijn een reëel risico: het scherm kan bijvoorbeeld aangeven dat Shift actief is terwijl het commando de RP2040 nog niet bereikt heeft. Het SHIDP-protocol werd daarom expliciet ontworpen met een checksum en met duidelijke begin- en eindsequenties om de consistentie te bewaken.

2.10 Samenvattende tabel van kerntechnologieën

Component	Functie in Starkpad	Kritieke uitdaging	Toegepaste strategie
Arduino UNO Q	Linux-host voor LVGL-UI, webserver en configuratie	Beperkte bandbreedte naar HID-laag	Asynchrone communicatie via UART, HID-prioriteit op de dongle
LVGL 9	Renderen van de grafische interface op touchscreen	Voldoende FPS op embedded hardware	Partial rendering en GPU-acceleratie
USB HID	Communicatie met host-PC als composite device	Correcte report descriptor voor keyboard en mouse	TinyUSB-stack op de RP2040
N-gram AI	Typvoorspelling op basis van lokale corpora	Opslagcapaciteit en snelheid	Trie-structuren en stupid back-off
JSON + web-UI	Dynamische configuratie zonder firmware-flash	Parsing-overhead op de UNO Q	FastAPI-webserver met cache in RAM

3 Technisch onderzoek

3.1 Inleiding

Dit hoofdstuk beschrijft in detail de technische realisatie van Starkpad, het prototype dat werd ontwikkeld in het kader van het researchproject en in deze bachelorproef verder is gedocumenteerd en gereflecteerd. De nadruk ligt op de functionele beschrijving van de ontwikkelde software, de opbouw van de applicatie, de toegepaste technologieën en de belangrijkste obstakels die tijdens de ontwikkeling werden overwonnen. Voor volledige broncode, installatie-instructies en onderdelenlijsten wordt verwezen naar de bijlages en naar de publieke repository op GitHub.

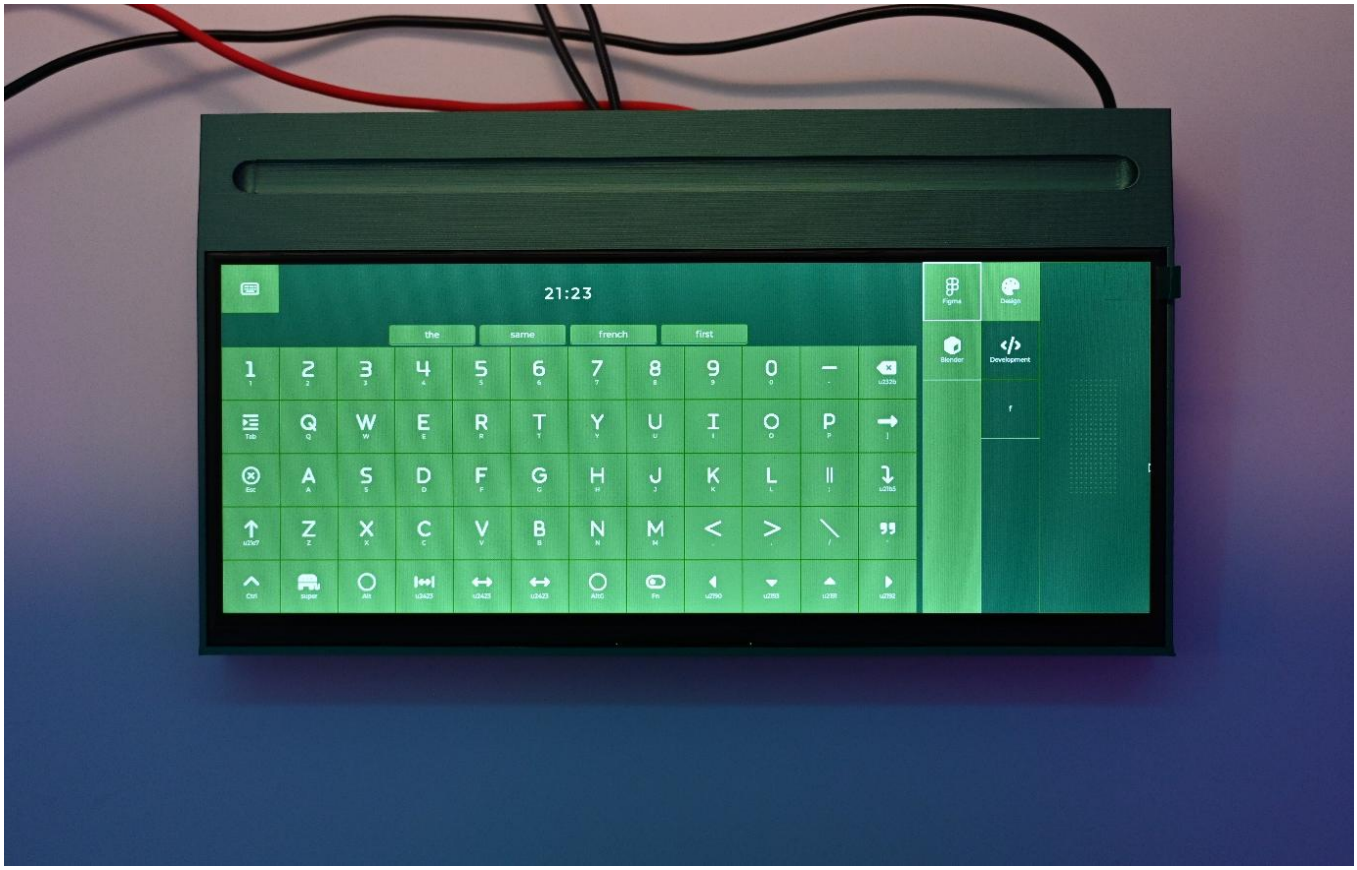
Figuur 1: Starkpad – vooraanzicht van het afgewerkte prototype.

3.2 Beschrijving van de ontwikkelde software

Starkpad is een open-source, Linux-gestuurd touchscreeninvoerapparaat dat drie functionele modi combineert in één fysiek toestel:

- Keyboard mode: een volledig virtueel QWERTY-toetsenbord met ondersteuning voor meerdere lay-outs (en-US, Colemak en nl-BE). Aanrakingen worden rechtstreeks vertaald naar USB HID-toetsaanslagen.
- Touchpad mode: een precisietouchpad dat een deel van het scherm omzet in een relatief muispad, voorzien van afzonderlijke knoppen voor linker- en rechtermuisklik.
- Macro grid mode: een raster van zestig programmeerbare knoppen, georganiseerd in categorieën en appprofielen. Voorgeconfigureerde profielen bestaan voor onder meer Figma, Blender en VS Code. Acties kunnen toetsaanslagen, mediacontroles, letterlijke tekst of tijdsvertragingen zijn.

Naast deze drie kernmodi beschikt Starkpad over een volledige web-UI voor configuratie, een afsluitbare admin-login, ondersteuning voor thema's en kleuraanpassing en een watchdogbeveiliging die voorkomt dat een verbroken UART-verbinding leidt tot vastgelopen toetsen op de host.



Figuur 2: Starkpad – bovenaanzicht. Het 12,3 inch capacitieve touchscreen is het centrale interactievlak.

Figuur 3: Starkpad – zijaanzicht. De 3D-geprinte behuizing verbergt de volledige hardware, inclusief de UNO Q, de voeding en de bekabeling.

3.3 Opbouw van de applicatie

3.3.1 Dual-brain-architectuur

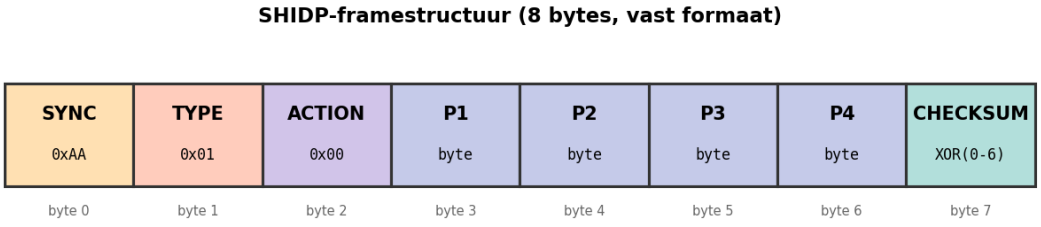
Starkpad is opgebouwd rond een dual-brain-architectuur. De interactielaag draait op de Arduino UNO Q onder Yocto Linux en omvat de volledige LVGL-UI, de Python-FastAPI-webserver, het JSON-configuratiesysteem en de SHIDP-zender. De USB HID-laag draait op de Seeed XIAO RP2040-dongle en is volledig toegewijd aan het valideren van SHIDP-frames en het emuleren van een composite USB HID-apparaat. Tussen beide chips zit een STM32U585 op de UNO Q, die dienstdoet als transparante UART-passthrough (zie figuur 4 in hoofdstuk 2).

Deze scheiding van verantwoordelijkheden heeft drie grote voordelen. Ten eerste blijft de HID-uitvoer deterministisch, ook wanneer de Linux-kernel even belast is door bijvoorbeeld een animatie of een netwerkverzoek. Ten tweede kunnen beide rekenhelften onafhankelijk van elkaar worden getest en gedebugd. Ten derde wordt de complexiteit van USB op embedded Linux vermeden: de UNO Q hoeft zelf geen USB HID te implementeren, wat door de DisplayPort Alt Mode-beperking van zijn USB-C-poort ook niet mogelijk zou zijn.

3.3.2 Het SHIDP-protocol

De communicatie tussen beide rekenhelften verloopt via een zelfontworpen serieel protocol: het Serial Human Interface Device Protocol, of SHIDP. SHIDP is een compact, eenrichtings, op vaste frames gebaseerd protocol met een XOR-checksum. Het werd expliciet ontworpen voor lage latentie, hoge betrouwbaarheid en een zuivere scheiding tussen logica en uitvoering.

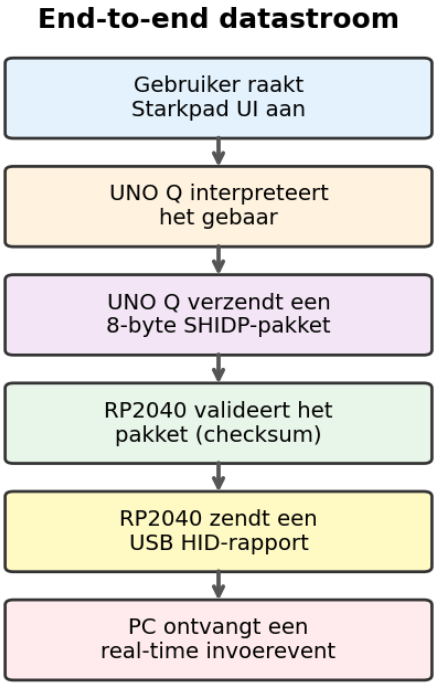
Elk SHIDP-frame bestaat uit exact acht bytes met een vast formaat:



Figuur 5: SHIDP-framestructuur. Elk frame heeft een vast formaat van acht bytes.

Het TYPE-veld duidt aan of het om een toetsenbord- (0x01), muis- (0x02) of media-event (0x03) gaat. Het ACTION-veld geeft aan of het om een tap (0x00), hold (0x01) of release (0x02) gaat. De vier parametervelden bevatten keycodes, modifierbits, muiscoördinaten of andere gegevens afhankelijk van het type event. De checksum is het XOR-resultaat van de zeven voorgaande bytes en wordt door de ontvanger gebruikt om corrupte frames te detecteren en stilzwijgend te negeren.

Het protocol werkt bewust in één richting en zonder acknowledgements. Deze keuze is gemotiveerd door de vereiste van lage latentie: een acknowledgement-mechanisme zou minstens één UART-ronde extra kosten per event. De robuustheid wordt in plaats daarvan gerealiseerd door drie mechanismen: de XOR-checksum, de vaste SYNC-byte en een watchdog van tachtig milliseconden die alle actieve toetsen automatisch vrijgeeft wanneer de UART-stream onderbroken wordt. Zonder deze watchdog zou een losgekoppelde kabel tot vastgelopen toetsaanslagen op de host leiden. Voor de volledige specificatie van het protocol wordt verwezen naar bijlage 8.3.



Figuur 6: End-to-end datastroom. Een aanraking resulteert binnen enkele milliseconden in een echt USB HID-event op de host.

3.3.3 De grafische interface (LVGL)

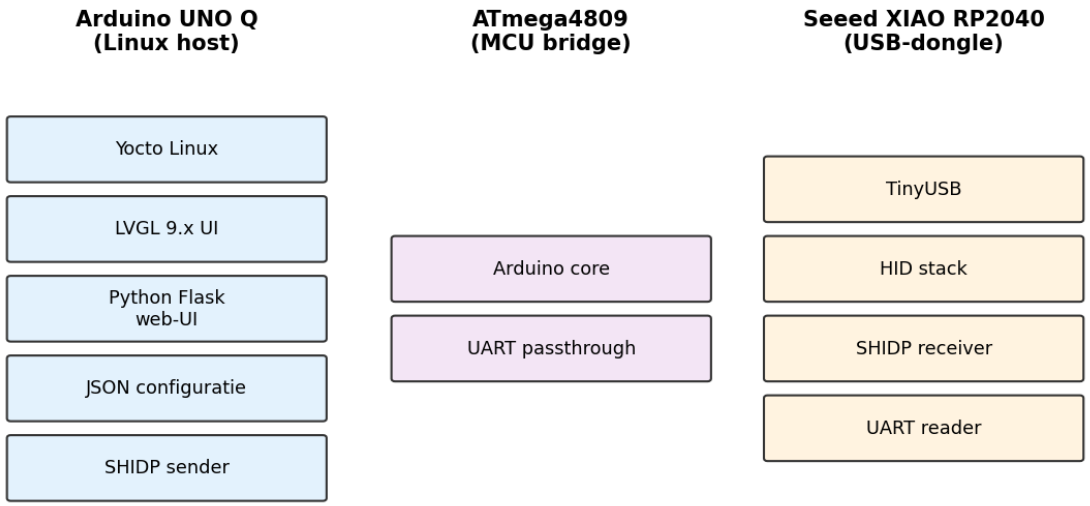
De grafische interface werd volledig opgebouwd in LVGL 9.x. Het hoofdvenster bestaat uit een bovenbalk met vier tabs (Keyboard, Touchpad, Grid en Status) en een dynamisch contentvlak dat afhankelijk van de actieve tab een andere layout toont. Door gebruik te maken van LVGL's ingebouwde widgets (lv_keyboard, lv_btn, lv_img) kon de basisinterface snel worden gerealiseerd. Aangepaste widgets werden toegevoegd voor onder meer het zestigknoppenraster en de iconenkeuze.

Figuur 9: Web-UI – het browsergebaseerde configuratiepaneel waarmee eindgebruikers knoppen, iconen, thema's en taal kunnen aanpassen zonder firmware te flashen.

3.3.4 Configuratie via JSON en web-UI

Alle configuratie van Starkpad wordt bijgehouden in JSON-bestanden op de UNO Q, in de map /Starkpad/lv_port_linux/ui-config/. Het hoofdbestand config.json bevat thema-instellingen, keyboard-opties en een geneste boomstructuur van categorieën, apps en knoppen. De web-UI, een FastAPI-applicatie die op poort 80 draait en via het mDNS-adres starkpad-uno-q.local bereikbaar is, laat toe om deze configuratie visueel te bewerken, iconen te kiezen uit de Font Awesome-set, toetsencombinaties in te stellen en instellingen met één klik op 'Apply Now' meteen op het apparaat te laden. Standaardreferenties zijn admin en starkpad123, wijzigbaar via het admin-paneel.

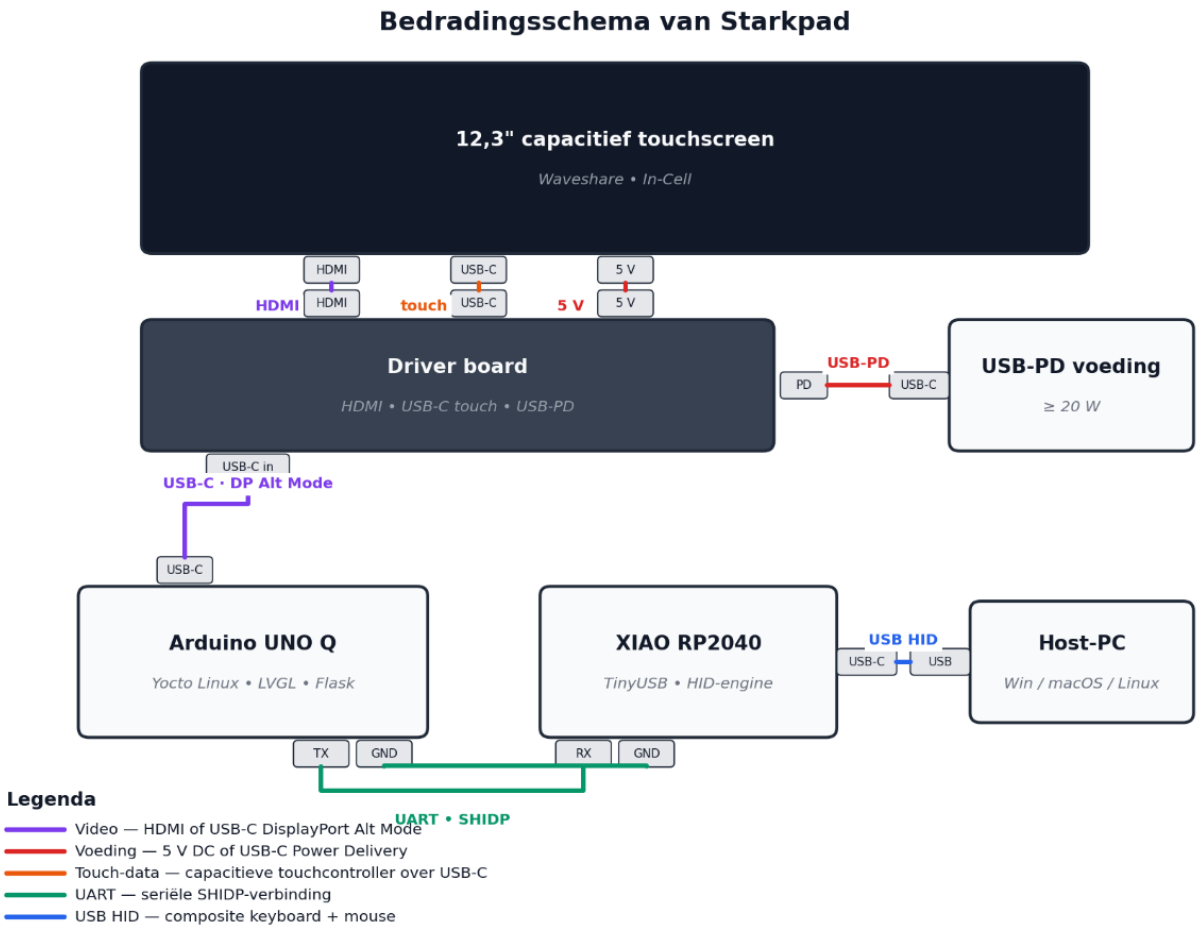
3.4 Achterliggende technologieën



Figuur 7: Technologiestack per hardwarecomponent.

Starkpad combineert een aantal technologieën die elk een specifieke rol vervullen:

- Yocto Linux op de UNO Q, omdat het de voordelen van een volwaardig besturingssysteem biedt (netwerk, bestandssysteem, Python-ondersteuning) op een embedded board.
- LVGL 9.x als grafische bibliotheek, wegens de lichte resourcevoetafdruk en het grote aantal ingebouwde widgets.
- FastAPI als Python-webserver voor de configuratie-UI. FastAPI werd verkozen boven zwaardere alternatieven zoals Django omwille van eenvoud en minimale afhankelijkheden.
- TinyUSB op de RP2040 voor de implementatie van de composite USB HID-klasse. TinyUSB is een actief onderhouden open-sourceproject met uitstekende documentatie.
- Arduino IDE-toolchain voor het ontwikkelen van de firmware op de STM32U585 en de RP2040. De keuze voor Arduino-compatibele firmware houdt de drempel voor bijdragers laag.
- 3D-printen (FDM) voor de behuizing. De STL-bestanden zijn opgenomen in de publieke repository, waardoor iedereen het toestel zelf kan vervaardigen.



Figuur 8: Bedradingsschema van Starkpad.

3.5 Overwonnen problemen

3.5.1 Beperking van de USB-C-poort op de UNO Q

Een eerste grote ontdekking tijdens het ontwerp was dat de enige USB-C-poort van de Arduino UNO Q gelijktijdig dienstdoet als voedingsingang en als DisplayPort Alt Mode-uitgang. Hierdoor kan deze poort niet worden ingezet als USB HID-device-poort. Het oorspronkelijke plan, waarbij de UNO Q zelf USB HID zou emuleren, werd daarom verlaten. De dual-brain-architectuur met een toegewijde RP2040-dongle is het rechtstreekse gevolg van deze vaststelling. Achteraf gezien was dit een zegen: de huidige architectuur is robuuster en beter testbaar dan een monolithische opzet zou zijn geweest.

3.5.2 Ontwerp van een eigen protocol (SHIDP)

Bestaande protocollen zoals USB-naar-serieel tunneling of RNDIS werden overwogen, maar bleken te zwaar voor de gewenste latentie. Daarom werd SHIDP van nul opgebouwd. De grootste uitdaging bestond erin om vastgelopen toetsen te vermijden bij verbindingsverlies. Dit werd opgelost via een tachtig-millisecondenwatchdog op de RP2040, die alle actieve toetsen vrijgeeft wanneer er gedurende die periode geen geldig frame binnenkomt.

3.5.3 Handmatige servicestop na boot

In de huidige implementatie moet de gebruiker na elke herstart van de UNO Q de service arduino-router stoppen via sudo systemctl stop arduino-router voordat Starkpad correct werkt. Deze service, standaard meegeleverd op de UNO Q, blokkeert het UART-kanaal waarop SHIDP actief is. Een permanente oplossing, waarbij de service via systemd wordt gedeactiveerd of vervangen, staat op de planning voor een volgende release.

3.5.4 Prestaties tijdens tab-transities

Tijdens het wisselen tussen modi (Keyboard, Touchpad, Grid) werden occasionele framerateverschuivingen waargenomen, waarbij de overgang bij normaal gebruik incidenteel kon oplopen tot ongeveer twee seconden. Dit is naar alle waarschijnlijkheid te wijten aan het opnieuw opbouwen van de widgethiërarchie bij elke tabwissel. Een sluitende oplossing werd binnen het tijdsbestek van dit project nog niet gevonden, al werd er actief naar gezocht. Een mogelijke optimalisatie, waarbij de widgets behouden blijven in het geheugen en enkel hun zichtbaarheid wordt gewijzigd, is geïdentificeerd als de meest kansrijke toekomstige verbetering.

3.6 Conclusie van het technisch onderzoek

De technische realisatie van Starkpad toont aan dat het haalbaar is om een volwaardig, gebruiksvriendelijk virtueel invoerapparaat te bouwen op basis van een Linux-gedreven touchscreen en een toegewijde USB HID-dongle. De modulaire architectuur, het SHIDP-protocol en de op JSON gebaseerde configuratie bieden samen een fundering waarop nieuwe functies kunnen worden toegevoegd zonder ingrijpende wijzigingen in de firmware. De grootste resterende uitdagingen, het verbeteren van de automatische boot en het verder uitbouwen van de tekstpredictie, worden in hoofdstuk 5 omgezet in concrete aanbevelingen.

4 Reflectie

4.1 Inleiding

In dit hoofdstuk wordt kritisch gereflecteerd over het resultaat van het researchproject. Starkpad is een open-sourceproject dat vanaf het moment van publicatie publiekelijk is gedeeld via GitHub, YouTube, LinkedIn en verschillende technologiepublicaties. Deze open-sourcebenadering heeft geresulteerd in een bovengemiddeld grote hoeveelheid externe feedback, die in de volgende paragrafen per kanaal wordt geanalyseerd. De reflectie volgt de structuur die in de rubric wordt aanbevolen: eerst een kritische zelfevaluatie, vervolgens een bespreking per externe bron, gevolgd door een synthese van de belangrijkste inzichten.

4.2 Zelfreflectie

4.2.1 Wat verliep vlot

De strategische keuze om de UI- en HID-lagen te scheiden (dual-brain-architectuur) heeft zich tijdens de volledige ontwikkeling bewezen als een sterke beslissing. Het systeem is daardoor modulair, eenvoudiger debugbaar en inherent robuuster dan een monolithische opzet. Het SHIDP-protocol met zijn watchdog en checksum heeft effectief voorkomen dat softwarefouten of bekabelingsincidenten resulteren in vastgelopen toetsen op de host. De web-UI voor configuratie is gebleken als een belangrijke hefboom: lay-outs, iconen, thema's en instellingen kunnen worden aangepast zonder dat firmware opnieuw gebouwd of geflasht moet worden. De appprofielen en het macroraster bieden onmiddellijke, tastbare meerwaarde voor gebruikers van tools zoals VS Code, Figma en Blender.

4.2.2 Wat kon beter

Een eerste aandachtspunt is de scope van het AI-gedeelte. De vooropgestelde typvoorspelling en autocorrectie werden uitgewerkt als een proof of concept op basis van een zelfgeschreven N-gram-model, maar de kwaliteit en de grootte van het taalcorpus laten ruimte voor verbetering. De rekenkracht van de UNO Q is weliswaar aanzienlijk voor een embedded board, maar onvoldoende voor moderne neurale modellen. De focus verschoof tijdens de uitvoering daarom naar macrofunctionaliteit en virtuele invoer, waar de marginale meerwaarde op korte termijn groter is. Een tweede aandachtspunt is de meetbaarheid van prestaties: FPS, touch-to-HID-latentie, woorden per minuut en foutpercentages werden vooropgesteld als meetbare indicatoren, maar zijn nog niet systematisch gelogd. Een derde punt is het bootflow-euvel dat in paragraaf 3.5.3 werd besproken: het handmatig stoppen van de arduino-router service is een frictiepunt voor eindgebruikers.

4.2.3 Feedback van de jury tijdens het researchproject

De jury van het researchproject evalueerde het project als sterk, zowel technisch als visueel. Het ontwerp en de algemene afwerking werden positief onthaald. Tegelijk werd vastgesteld dat de hoeveelheid daadwerkelijk uitgevoerd technisch onderzoek tijdens de presentatie onvoldoende naar voren kwam. Zo werd er substantieel onderzoek verricht naar het eigen seriële protocol, naar een zelfgeschreven op N-gram gebaseerde typvoorspelling en naar het werken met een nieuw embedded board. Door de beperkte presentatietijd van vijftien minuten bleef dit onderzoekscomponent echter beperkt toegelicht. Deze bachelorproef biedt de gelegenheid om deze onderzoekscomponent in volledige breedte en diepte te documenteren.

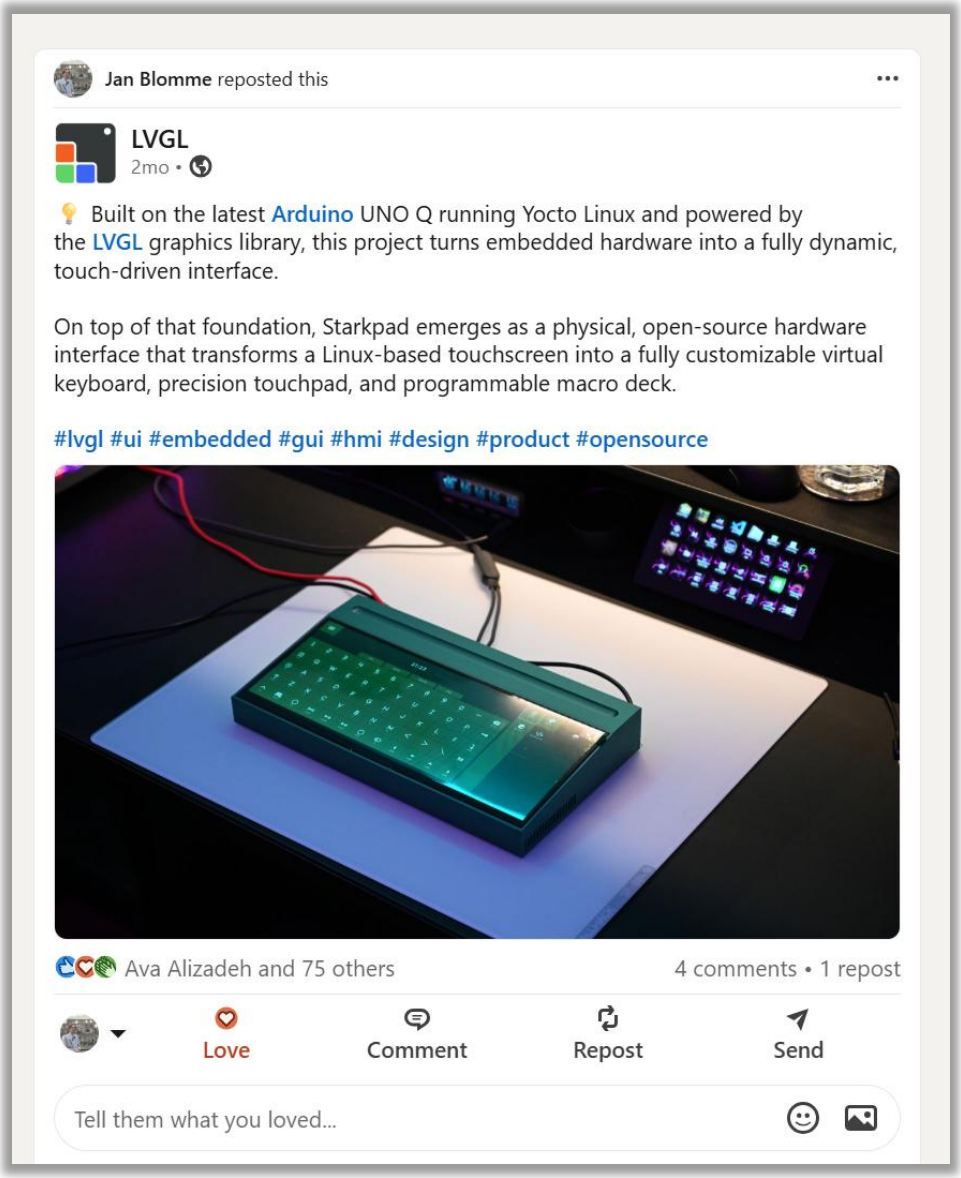
4.3 Feedback uit externe bronnen

Starkpad genereerde sinds de publicatie zichtbare interesse in diverse professionele en hobbyistische gemeenschappen. De volgende paragrafen bespreken per kanaal welke feedback werd ontvangen en hoe deze bijdraagt aan een gefundeerde evaluatie van het project.

4.3.1 Reflectie met de LVGL-community

Het officiële LVGL-account op LinkedIn (meer dan achtduizend volgers) publiceerde een post waarin Starkpad werd voorgesteld als een exemplarisch open-sourceproject op basis van hun grafische bibliotheek. De post werd door ruim vijfenzeventig professionals uit embedded development, UX-design en productontwikkeling geliked en deelde een rechtstreekse verwijzing naar de GitHub-repository. De tekst van de post benadrukt expliciet de combinatie 'Yocto Linux + Arduino UNO Q + LVGL + USB HID-dongle' als de technologische bouwstenen waardoor Starkpad embedded hardware omzet in een volledig dynamisch, touchgestuurd interface.

Deze validatie is bijzonder betekenisvol omdat ze komt van de maintainers van de gebruikte UI-bibliotheek. Ze bevestigt dat de gekozen LVGL-implementatie, de architecturale keuzes en het gebruik van partial rendering in lijn liggen met de best practices die LVGL zelf uitdraagt. Bovendien creëert de publicatie een directe feedbackloop met internationale embedded developers, die in de commentaren vragen stellen over componenten en implementatiekeuzes.

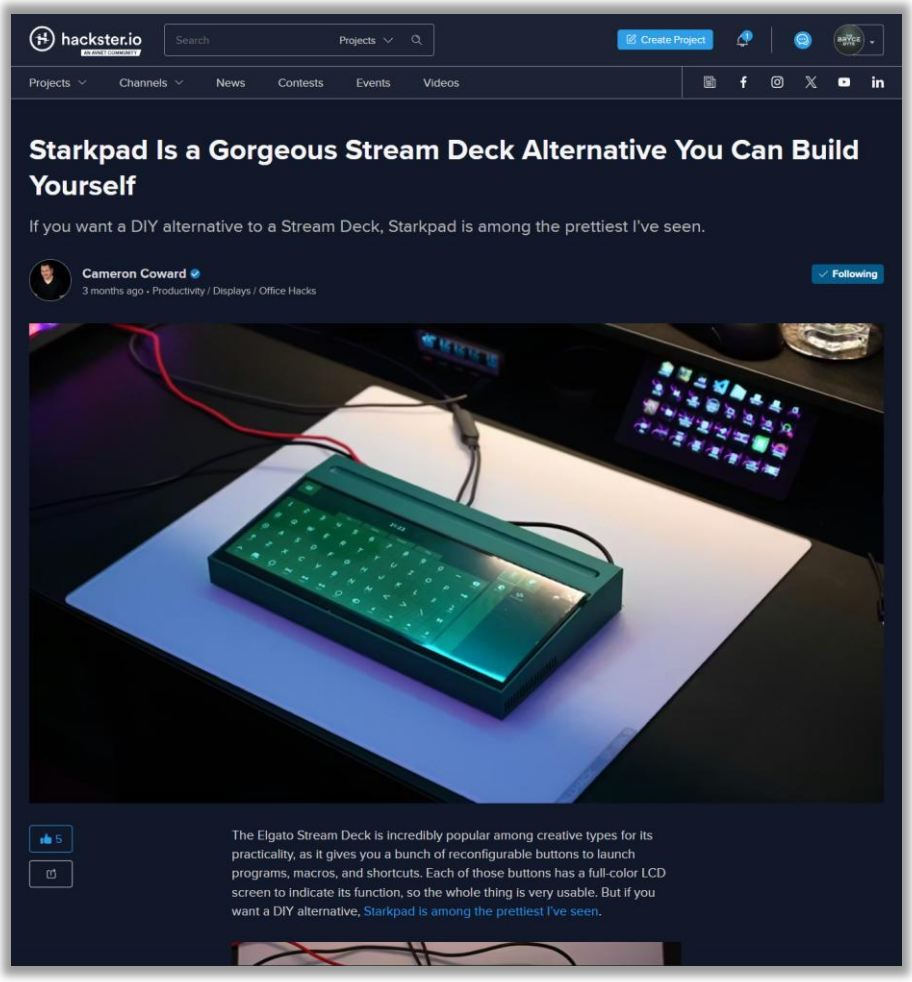


Figuur 10: Internationale erkenning – de LVGL LinkedIn-post over Starkpad (8.034 volgers, 75+ reacties, 1 repost).

4.3.2 Reflectie via Hackster.io

Hackster.io, een van de grootste platforms voor embedded hardware en maker projects, publiceerde een uitgebreid redactioneel artikel onder de titel 'Starkpad Is a Gorgeous Stream Deck Alternative You Can Build Yourself'. Het artikel positioneert Starkpad expliciet als een haalbaar open-source alternatief voor de commerciële Stream Deck van Elgato, en benadrukt daarbij de esthetische kwaliteit van de behuizing, de professionele UI en de reproduceerbaarheid dankzij de publieke BOM en STL-bestanden.

Cameron Coward, redacteur bij Hackster.io en zelf maker en YouTuber bij het kanaal Serial Hobbyism, beschreef Starkpad expliciet als "among the prettiest [DIY Stream Deck alternatives] I've seen" en bevestigde dat "the resulting device seems to work very well" en "gives BlommeJan far more flexibility than he'd get from a Stream Deck, but in a similar form factor" [55]. Naast deze waardering plaatste hij twee kritische kanttekeningen die rechtstreeks aanleiding gaven tot inhoudelijke correcties. Ten eerste signaleerde hij dat het bridge-MCU op de UNO Q een STM32U585 is en niet een ATmega4809 zoals oorspronkelijk in de README en de eerste versie van dit document vermeld stond. Deze fout werd over alle projectdocumentatie rechtgezet. Ten tweede merkte hij op: "That seems a bit convoluted, as the STM32 could act as a USB HID directly. But I assume that BlommeJan had a good reason for doing it that way." Deze opmerking dwong tot een expliciete motivering van de keuze om de RP2040 te behouden in plaats van de STM32U585 in te zetten als HID-host. Het USB-randapparaat van de STM32U585 is op de UNO Q via de PCB doorgekoppeld naar de USB-C-poort, die door de DisplayPort Alt Mode én de Power Delivery-stroomingang reeds bezet is. De RP2040 als afzonderlijke dongle vormt daarmee geen overbodige complexiteit, maar een noodzakelijke ontkoppeling van fysieke USB-poorten.

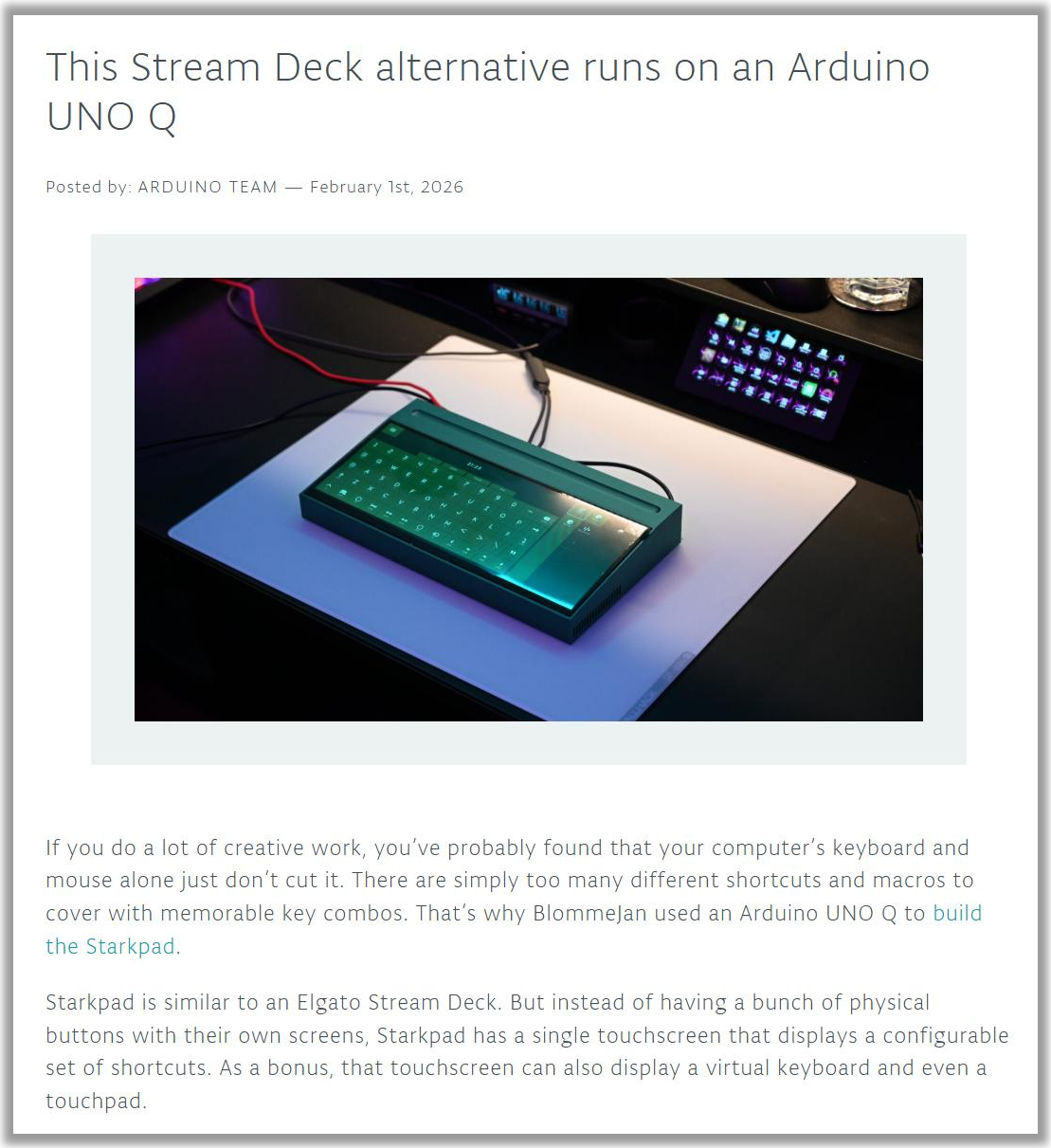


Figuur 11: Externe publicatie – het redactioneel artikel over Starkpad op Hackster.io.

4.3.3 Reflectie via de Arduino Blog

De officiële Arduino Blog publiceerde op 1 februari 2026 een artikel onder de titel 'This Stream Deck alternative runs on an Arduino UNO Q', gericht op de wereldwijde Arduino-community. Het artikel beschrijft Starkpad als een community-spotlightproject dat de nieuwe mogelijkheden van de UNO Q demonstreert: de combinatie van een volwaardig Linux-subsysteem met een klassieke microcontroller binnen één formfactor.

De boodschap die uit dit artikel voortvloeit is dubbel relevant. Enerzijds valideert ze de technische keuze om de UNO Q te gebruiken als interactieplatform in plaats van een klassieke Raspberry Pi of een enkelvoudige microcontroller. Anderzijds biedt ze indirecte feedback: het artikel geeft aan dat de Arduino-gemeenschap geïnteresseerd is in reproduceerbare projecten op de UNO Q, wat een bevestiging vormt van de keuze om de repository publiek en goed gedocumenteerd te maken. Het artikel typeert Starkpad expliciet als een "Tony Stark–inspired user interface built in LVGL on top of the Linux operating system" en concludeert dat de implementatie "works smoothly because Starkpad runs on a dedicated Arduino UNO Q" [56].



This Stream Deck alternative runs on an Arduino UNO Q

Posted by: ARDUINO TEAM — February 1st, 2026

If you do a lot of creative work, you've probably found that your computer's keyboard and mouse alone just don't cut it. There are simply too many different shortcuts and macros to cover with memorable key combos. That's why BlommeJan used an Arduino UNO Q to build the Starkpad.

Starkpad is similar to an Elgato Stream Deck. But instead of having a bunch of physical buttons with their own screens, Starkpad has a single touchscreen that displays a configurable set of shortcuts. As a bonus, that touchscreen can also display a virtual keyboard and even a touchpad.

Figuur 12: Externe publicatie – het artikel over Starkpad op de officiële Arduino Blog (1 februari 2026).

4.3.4 Reflectie via YouTube en de makersgemeenschap

Naast de vermeldingen door professionele media werd een demonstratievideo over Starkpad gepubliceerd op het YouTube-kanaal The Bryce Byte, een persoonlijk kanaal waar regelmatig technische projecten worden gedocumenteerd. De video behaalde ongeveer drieduizendvierhonderd weergaven, leverde veertig nieuwe abonnees op en ontving in totaal twintig commentaren. Voor een recent gelanceerd kanaal is dit een opvallende groei die aantoont dat het project de belangstelling wekt van een bredere DIY-doelgroep.

Het merendeel van de commentaren was positief van aard, met meerdere kijkers die de intentie uitspraken om het toestel zelf na te bouwen. De meest substantiële inhoudelijke discussie ontstond rond een commentaar van Owen Rothlerner (@OWENROTHLERNER), die de volledige tech stack ter discussie stelde:

> *"Cool but complicated, Linux could send keystrokes but via some sort of either badUSB or application layer server client. What's the reason to do this versus some ultra widescreen tablet-like device running an app to interface with your windows PC? Especially if not primary input the latency allowed is higher without user frustration (think if your mouse was delayed by 2ms versus sub-ms speed). I personally control a computer via Moonlight, on lan at high speeds the latency for mouse and keyboard is imperceptible (1ms or less)..."*

Deze opmerking raakt een fundamentele ontwerpkeuze: waarom zou men een dedicated hardwareapparaat bouwen wanneer een tablet met Moonlight, RDP of een vergelijkbare client-server-oplossing dezelfde functionaliteit kan leveren met lagere latentie? In mijn antwoord op de video lichtte ik toe waarom Starkpad bewust voor de hardware-route koos:

> *"The difference here is mostly philosophy, not speed. I wanted this to behave like a dumb hardware device, not a remote control app. The RP2040 shows up as a normal USB keyboard/mouse, so it works everywhere (BIOS, login screens, locked machines, no network, no software installed). Also, Starkpad isn't meant to replace your mouse or keyboard. It's more of a contextual control surface for macros, shortcuts, text snippets, app-specific layouts. For that, reliability and isolation matter more than squeezing out the last millisecond. A tablet app or Moonlight setup is totally valid, just solving a slightly different problem."*

Dit publieke gesprek bracht twee inzichten naar voren die de positionering van het project verscherpen. Ten eerste bevestigt het dat Starkpad zich niet positioneert als een latentie-geoptimaliseerd invoerapparaat voor primaire input, maar als een contextueel controleoppervlak waar betrouwbaarheid en isolatie zwaarder wegen dan een laatste millseconde-winst. Ten tweede toont het aan dat de hardware-eigenschappen, werken zonder netwerk, zonder host-side software, in BIOS-omgevingen en op vergrendelde machines, een wezenlijk onderscheid vormen ten opzichte van software-gebaseerde alternatieven. Dit type publieke uitwisseling levert exact het soort externe reflectie dat een open-sourceproject voor de auteur toegankelijk maakt en dat in een gesloten ontwikkeltraject niet beschikbaar zou zijn.

4.3.5 Gesprek met John Sullivan (The Digital Shepherd)

Naar aanleiding van de LinkedIn-post van LVGL kwam reactie van John Sullivan, een industriële IT-consultant die actief is rond AI, IoT, Edge Computing en IT-OT-integratie en zichzelf profileert als The Digital Shepherd. Dat een professional uit het IT- en Edge-werkveld het project spontaan opmerkte en erop reageerde, vormt op zichzelf een relevant signaal van interesse vanuit een doelgroep buiten de makersgemeenschap.

De interactie verliep publiek en bondig. Sullivan stelde één concrete vraag, namelijk waar de gebruikte schermmodule verkrijgbaar is, omdat die voor hem niet onmiddellijk terug te vinden was in de README. Daarop volgde een verwijzing naar de gebruikte component, het Waveshare 12,3-inch capacitief touchscherm met In-Cell-technologie, en naar de BOM.md waarin alle onderdelen gedocumenteerd staan. De vraag werd publiek geapprecieerd en de reactie haalde binnen het bereik

van deze één post een meetbare zichtbaarheid (eenen-twintig weergaven), wat erop wijst dat de uitwisseling ook door derden werd gevolgd.

Kritisch beschouwd was dit geen technische ontwerpevaluatie, maar een documentatievraag, en het is belangrijk de uitwisseling ook als zodanig te kaderen. De waarde ervan ligt op twee vlakken. Ten eerste functioneerde de vraag als een directe gebruikstest van de vindbaarheid van de documentatie: dat een geïnteresseerde professional de schermmodule niet meteen terugvond, wijst op een concreet vindbaarheidsprobleem in de README. Dit werd vervolgens verholpen door de BOM prominenter te koppelen, waardoor de drempel om het project te reproduceren naar verwachting verlaagt. Ten tweede bevestigt het signaal, binnen de beperkte reikwijdte van één publieke interactie, dat het project interesse opwekt bij het beoogde professionele publiek. Een diepgaandere inhoudelijke terugkoppeling van Sullivan bleef uit, zodat aan deze uitwisseling geen verdere conclusies over de technische merites van Starkpad worden verbonden.

4.3.6 Reflectie met Klaudia Warmus (Granstudio, Turijn)

Tijdens de stage bij Granstudio te Turijn werd het project ook voorgelegd aan Klaudia Warmus, externe promotor en vertegenwoordigster van een studio waar dagelijks met professionele ontwerpsoftware gewerkt wordt. Granstudio is actief in design en engineering voor de automotive sector, een sector waar efficiënte invoerapparatuur en contextgevoelige shortcuts een reële meerwaarde kunnen betekenen.

Klaudia Warmus beoordeelde het project als helder, relevant en goed gestructureerd, met een solide technische basis. Ze waardeerde in het bijzonder de scheiding tussen de interfacelaag en de HID-laag en zag in de combinatie van een web-UI en JSON-configuratie een praktische mate van flexibiliteit. Vanuit een designworkflow-perspectief herkende ze in Starkpad het potentieel van een contextgevoelig bedieningsvlak voor shortcuts, macro's en applicatiespecifieke acties. Naast de eerdergenoemde tools (Figma, Blender, VSCode) wees ze op een bijkomende toepassing in Unreal Engine-configuratoren, waar het apparaat snel kan schakelen tussen varianten, materialen, camera's, scenes of presentatiemodi.

Daarnaast onderschreef ze dat de scriptie de huidige beperkingen van het prototype, de handmatige boot flow, de voedingsvereisten, de reikwijdte van de AI en het ontbreken van systematische latentiemetingen, expliciet erkent, en beschouwde dit als een sterk punt in de reflectie. Als verbeterpunt suggereerde ze om deze beperkingen nog directer te koppelen aan de volgende ontwikkelingsstappen, zodat voor de lezer duidelijk wordt wat nodig is om Starkpad van prototype naar een meer afgewerkt product te brengen. Haar eindoordeel was dat dit een degelijk en relevant bachelorproject is met een goede basis voor verdere ontwikkeling.

4.4 Synthese van de externe feedback

De externe feedback convergeert naar vier kernbevindingen die voor het vervolg van het project en voor het advies in hoofdstuk 5 relevant zijn:

- Het project heeft een duidelijke product-market fit als open-source alternatief voor bestaande commerciële macropads, zowel bij makers als bij professionele IT'ers.
- De dual-brain-architectuur en het SHIDP-protocol worden door professionele embedded-ontwikkelaars gezien als een verstandige ontwerpkeuze; ze zijn blijkbaar geen hobbyoplossing maar een legitieme benadering voor betrouwbare invoerapparaten.
- De grootste kwaliteitsversterkers op korte termijn liggen niet in nieuwe functies, maar in reproduceerbaarheid: duidelijker documentatie, eenvoudiger boot flow, complete BOM en transparante keuzemotivering.
- De AI-typvoorspellingsfeature wordt als interessant bevonden, maar is geen dealbreaker: de reflectie bevestigt dat macro's en contextuele lay-outs de grootste onmiddellijke gebruikerswaarde bieden.

- Bij deze bevindingen past één belangrijke nuance. De externe feedback is tot op heden afkomstig van de open-sourcegemeenschap en van experten die het project hebben beoordeeld, maar nog niet van een onafhankelijke gebruiker die Starkpad zelf heeft gereproduceerd en gedurende langere tijd in een eigen werkomgeving heeft ingezet. De huidige gebruikservaring berust dan ook hoofdzakelijk op het gebruik door de ontwikkelaar zelf. Een gedocumenteerde reproductie en langdurig gebruik door een externe partij wordt daarom beschouwd als een waardevolle en nog openstaande validatiestap, die de bevindingen uit dit hoofdstuk zou kunnen bevestigen of bijsturen. De open-sourcepublicatie, de volledige BOM en de installatiehandleiding in de bijlagen zijn er net op gericht om deze drempel tot reproductie zo laag mogelijk te houden.

5 Advies

5.1 Inleiding

Het advies in dit hoofdstuk is bestemd voor ontwikkelaars, studenten en organisaties die willen starten met een soortgelijk project: een embedded touchscreeninvoerapparaat met professionele specificaties. Het advies bouwt rechtstreeks voort op het eigen onderzoek en op de externe feedback uit hoofdstuk 4.

5.2 Bruikbaarheid en toepasbaarheid van Starkpad

Uit de reflectie blijkt dat Starkpad op drie manieren onmiddellijk bruikbaar is in de praktijk:

- Als educatief referentieproject voor embedded studenten. De combinatie van Linux, LVGL, USB HID, UART en FastAPI dekt een groot deel van het curriculum rond embedded systemen af.
- Als uitgangspunt voor commerciële productontwikkeling. De modulaire architectuur en de open-source licentie laten organisaties toe om het ontwerp over te nemen en aan te passen aan hun specifieke noden, zonder vendor lock-in.
- Als individueel productiviteitsapparaat voor creatievelingen en ontwikkelaars. Met een beetje configuratie vervangt Starkpad een commerciële Stream Deck en biedt het bovendien de flexibiliteit van een volwaardig virtueel toetsenbord en touchpad.
- Als contextgevoelig bedieningsvlak in professionele designworkflows. In toepassingen zoals Unreal Engine-configuratoren kan Starkpad snel schakelen tussen varianten, materialen, camera's, scenes of presentatiemodi, wat de meerwaarde van shortcuts en macro's in de automotive- en designsector bevestigt.

5.3 Wanneer Starkpad niet de juiste keuze is

Een eerlijk advies moet ook de grenzen van de aanpak vastleggen. In zes specifieke contexten is Starkpad expliciet niet de aangewezen oplossing.

Voor invoer waar sample-accurate timing of submillisecondelatentie vereist is, zoals competitieve gaming, MIDI-controllers voor live-muziekproductie of professionele e-sportsystemen, voegt de dual-brain-keten van touchpaneel naar Linux-userspace naar UART naar RP2040 naar USB HID minstens vijftien tot dertig milliseconden extra toe ten opzichte van een mechanisch toetsenbord met directe USB-scanning. Een traditioneel mechanisch toetsenbord op QMK met een polling rate van duizend hertz blijft in die context de juiste keuze.

Voor toepassingen waar N-key rollover bij hoge polling rates noodzakelijk is, vormt het composite-HID-rapport op de RP2040 een bottleneck. Het huidige report descriptor ondersteunt zes simultane toetscodes plus modifiers; voor stenografietoepassingen of geavanceerde shortcut-combinaties met meer dan zes gelijktijdige inputs volstaat dit niet.

In mobiele of batterijgedreven gebruikssituaties waar geen netaansluiting beschikbaar is, vormt het piekverbruik van vijf tot acht watt een dealbreaker. Een commerciële Loupedeck Live (ongeveer 2,5 W via USB-bus) of een Stream Deck Mini (ongeveer 1,5 W) past wel binnen een USB 2.0-stroombudget.

Voor gebruikers die uitsluitend één applicatie ondersteunen zonder behoefte aan contextgevoelige profielen, levert een QMK-gebaseerde dedicated macropad (bijvoorbeeld een 4×3-bord met mechanische schakelaars) lagere kosten, instant-on gedrag zonder boot-tijd van vijftien tot dertig seconden, en een MTBF die in mechanische schakelaars typisch een ordegrootte hoger ligt dan in capacitieve touchpanelen.

In BIOS-, UEFI- of POST-omgevingen waar invoer nodig is voordat het besturingssysteem laadt, werkt Starkpad niet: zowel het LVGL-frontend als de RP2040-firmware zijn pas operationeel na een volledige

Linux-boot van het UNO Q-subsysteem. Een klassieke USB-toetsenbordverbinding blijft in die context essentieel.

Tot slot is Starkpad niet geschikt voor industriële omgevingen waar IP65-bescherming, EMC-certificering of een gegarandeerde leveringsketen vereist zijn. De huidige bill of materials is uitsluitend gevalideerd voor laboratorium- en kantoorgebruik; commerciële productie zou een grondig herontwerp vragen op het vlak van componentselectie en behuizing.

Voor wie zich in een van deze categorieën herkent, zijn de juiste alternatieven respectievelijk een QMK-toetsenbord, een Loupedeck Live, een commerciële Elgato Stream Deck of een industrieel HMI-paneel van bijvoorbeeld Siemens of Beckhoff.

5.4 Aanbevelingen uit het werkveld

Op basis van de gesprekken en externe feedback uit hoofdstuk 4 kunnen volgende concrete aanbevelingen worden geformuleerd:

- Zorg voor een transparante, volledige bill of materials. De LinkedIn-conversatie met John Sullivan toonde aan dat zelfs ervaren professionals soms aarzelen om een project na te bouwen zolang onderdelen niet expliciet vermeld staan. Plaats de BOM prominent in de README of op de eerste pagina van de documentatie.
- Automatiseer de boot flow. De handmatige stap met sudo systemctl stop arduino-router is een belangrijke drempel voor eindgebruikers. Voorzie een systemd-configuratie die de juiste services start en stopt en die documenteer je als onderdeel van de installatiehandleiding.
- Begin met een simpel protocol en breid later uit. SHIDP werd bewust eenvoudig gehouden (acht bytes, vast formaat, één richting). Dit vereenvoudigt debuggen en testen. Voeg pas bidirectionele functies toe (status-updates van PC naar toestel) wanneer er een concrete use case is.
- Kies open configuratie boven harde firmware-lay-outs. JSON en een web-UI geven eindgebruikers autonomie en verlagen de support kost aanzienlijk in vergelijking met QMK-achtige, op compilatie gebaseerde workflows.
- Documenteer ontwerpkeuzes, niet alleen code. Een publiek project trekt vragen aan over het waarom van bepaalde beslissingen. Door die keuzes te documenteren in de repository (bijvoorbeeld in ADR-bestanden of in de README) bespaar je zowel jezelf als bijdragers tijd.

5.5 Stappenplan voor wie een vergelijkbaar project wil starten

Voor een ontwikkelaar die dezelfde onderzoeksvraag wil aanpakken, wordt onderstaand stappenplan aanbevolen. De stappen zijn geordend op oplopende complexiteit en op afhankelijkheden tussen subsystemen.

- Stap 1 – Definieer eerst de HID-contract. Bepaal welke USB-klassen (keyboard, mouse, media, eventueel composite) het toestel moet ondersteunen en beschrijf de bijhorende report descriptors. Dit contract bepaalt alle verdere ontwerpbeslissingen.
- Stap 2 – Ontwerp het seriële protocol. Houd het zo eenvoudig mogelijk: een vaste framegrootte, een SYNC-byte, een checksum en een watchdog volstaan voor de meeste toepassingen. Documenteer de frame-layout vóór de code geschreven wordt.
- Stap 3 – Bouw de HID-laag eerst. Test deze volledig los van de UI door met de hand frames via een terminal te versturen. Pas wanneer de HID-laag betrouwbaar werkt, mag de UI worden aangesloten.
- Stap 4 – Bouw een minimale UI op LVGL (of vergelijkbaar). Begin met één mode (bijvoorbeeld keyboard). Valideer de touchverwerking en de snelheid voordat complexere interacties (macro grid, profielen) worden toegevoegd.
- Stap 5 – Introduceer JSON-configuratie vroeg. Ook als er initieel slechts vijf knoppen zijn, is het voordeliger om die al uit JSON te laden dan om ze hardcoded in code te zetten. Dit voorkomt latere refactors.

- Stap 6 – Voeg een web-UI toe pas nadat de JSON-structuur stabiel is. Een webinterface die op een wijzigend schema werkt, creëert onnodige complexiteit.
- Stap 7 – Publiek maken en itereren op basis van feedback. De waardevolle feedback die Starkpad kreeg via LVGL, Hackster.io, Arduino Blog en YouTube is slechts mogelijk omdat het project vroeg werd gedeeld. Publiceer op een redelijk moment en aanvaard dat niet alles al perfect hoeft te zijn.

5.6 Ontwikkelde tools en artefacten voor het werkveld

Tijdens het onderzoek werden meerdere concrete artefacten ontwikkeld die beschikbaar zijn voor wie een vergelijkbaar project wil starten:

- De volledige broncode van Starkpad, beschikbaar als open source onder MIT-licentie op https://github.com/BlommeJan/Starkpad.
- Een gedocumenteerde installatiehandleiding (zie bijlage 8.1) en een gebruikershandleiding (zie bijlage 8.2).
- De volledige SHIDP-protocolspecificatie, inclusief voorbeelden en referentie-implementaties in C en Python (zie bijlage 8.3).
- De STL-bestanden voor een 3D-geprinte behuizing (starkpad-case-003.stl), rechtstreeks downloadbaar uit de repository.
- Een transparante bill of materials (BOM.md) met prijzen en leveranciers.
- Een demonstratievideo op het YouTube-kanaal The Bryce Byte, beschikbaar op https://youtu.be/fDFRH98BEmM.

5.7 Conclusie van het advies

De belangrijkste boodschap aan wie een vergelijkbaar project wil aanpakken, is dat de architectuur belangrijker is dan de specifieke hardware. Door vroeg te kiezen voor een dual-brain-benadering, een eenvoudig maar robuust communicatieprotocol en datagedreven configuratie, blijft het project schaalbaar en onderhoudbaar naarmate de scope groeit. De publicatie op open-sourceplatforms is bovendien geen bijzaak: ze is de motor die feedback, validatie en onverwachte samenwerkingen mogelijk maakt.

6 Besluit

De onderzoeksvraag van deze bachelorproef luidde: Hoe kunnen we een klassiek toetsenbord combineren met een microcontroller en een touchscreen om via AI gepersonaliseerde typ- en functiemogelijkheden te bieden? Het antwoord, op basis van zowel het theoretische onderzoek als de praktische realisatie van Starkpad, is dat een dergelijke combinatie haalbaar is op voorwaarde dat de verantwoordelijkheid voor de interactie en voor de USB HID-uitvoer bewust worden gescheiden.

Het hoofdstuk Research toonde aan dat LVGL een geschikte bibliotheek is voor de UI-laag, dat USB HID via een composite device-descriptor moeiteloos uitvoert wat nodig is voor een toetsenbord en muis in één, en dat lichte N-gram-modellen een realistische route vormen voor typvoorspelling op embedded hardware. Het technische onderzoek bevestigde deze bevindingen door Starkpad te bouwen op een Arduino UNO Q met Yocto Linux, die via een zelfontworpen 8-byte protocol (SHIDP) communiceert met een Seeed XIAO RP2040-dongle. Deze dual-brain-architectuur werd geen compromis uit nood, maar werd dankzij de gedwongen beperking van de USB-C-poort van de UNO Q een doordachte ontwerpbeslissing die robuustheid, testbaarheid en schaalbaarheid aanzienlijk verhoogt.

De reflectie, gevoed door feedback uit de officiële LVGL-community, publicaties op Hackster.io en de Arduino Blog, en meer dan drieduizend YouTube-weergaven, toonde aan dat het project zowel technisch als visueel een positief onthaal krijgt in uiteenlopende professionele en hobby-gemeenschappen. De belangrijkste verbeterpunten liggen niet in nieuwe functies, maar in reproduceerbaarheid en automatisering van de boot flow. Deze inzichten werden in hoofdstuk 5 vertaald naar een concreet stappenplan voor wie een vergelijkbaar project wil starten.

Mogelijke paden voor vervolgonderzoek liggen in vier concrete richtingen, elk geformuleerd als een afgebakend projectvoorstel.

Lokale neurale typvoorspelling op edge-hardware. Het huidige N-gram-model levert deterministische voorspellingen maar mist semantisch begrip. Een vervolgonderzoek zou kunnen meten of een gequantiseerd TinyLlama-1.1B-model, gehost op de Adreno 702-GPU van de UNO Q via llama.cpp met Vulkan-acceleratie, een latentie onder vijftig milliseconden kan halen voor next-word-prediction. De vergelijking met het huidige back-off-N-gram zou aantonen of het rekenoverhead-verschil, naar schatting een factor honderd in milliseconden, gerechtvaardigd wordt door een meetbare verbetering in voorspellingsaccuratesse op een Nederlandstalig of meertalig corpus.

Aanwezigheidsdetectie en automatische schermdimming via mmWave-sensoren. De huidige Starkpad blijft permanent op vol vermogen draaien wanneer de gebruiker afwezig is. Een vervolgonderzoek zou de inbouw van een 60 GHz mmWave-radarsensor (zoals de Infineon BGT60TR13C, identiek aan wat Google gebruikt in Pixel-telefoons voor Soli-gebaren) kunnen integreren met de Wake on Approach- en Lock on Leave-functies van Windows 11. Het meet- en designwerk zou kunnen vaststellen hoeveel watt gemiddeld bespaard wordt over een werkweek van veertig uur en of de toegevoegde sensor de prijsklasse van het toestel onder honderd euro houdt.

Systematische latentie- en frame-timing-meetcampagne. De huidige cijfers berusten op waarnemingen bij normaal gebruik en niet op een geïnstrumenteerde meting. Een vervolgonderzoek zou een geautomatiseerd meetopstelling kunnen bouwen met een fotodiode op het scherm, een logic-analyzer op de UART-lijn en een USB-snifferpoort op de host-PC, om de touch-to-light-keten over duizend events te kwantificeren in percentielwaarden. De resultaten zouden aantonen of de SHIDP-watchdog van tachtig milliseconden ruim genoeg of net te krap is voor reële netwerkbelasting, en zouden vergelijkbare numerieke benchmarks opleveren als waarmee Elgato of Razer hun eigen producten karakteriseren.

App-profielcatalogus via opt-in telemetrie en community-curatie. De huidige drie profielen (Figma, Blender, VS Code) dekken slechts een fractie van de mogelijke use cases. Een vervolgonderzoek zou

een opt-in telemetrie-laag kunnen ontwerpen waarbij de web-UI gepseudonimiseerde profieldata uploadt naar een centrale registry, vergelijkbaar met hoe de Visual Studio Code Marketplace community-extensies cureert. De ontwerpvraag, hoe een privacy-vriendelijke aggregatie van toetsenbordmacro's kan worden opgezet zonder identificatiegegevens te lekken, sluit aan bij actueel onderzoek rond differential privacy in client-side machine learning.

Starkpad toont uiteindelijk aan dat embedded productontwikkeling binnen de context van een bacheloropleiding kan resulteren in een oplossing die op functioneel én esthetisch vlak vergelijkbaar is met commerciële alternatieven, op voorwaarde dat ze open, modulair en vanuit een duidelijke ontwerpfilosofie wordt opgezet. De open-sourcebenadering maakt van Starkpad geen eindproduct, maar een startpunt waarop anderen kunnen voortbouwen.

7 Literatuurlijst

De volgende bronnen werden geraadpleegd tijdens het literatuuronderzoek en de technische uitwerking van deze bachelorproef. De verwijzingen volgen de IEEE-notatie.

[1] J. Blomme, "StarkPad – Contractplan researchproject," intern document, Howest, 2025.

[2] LVGL, "Light and Versatile Graphics Library," 2026. [Online]. Beschikbaar: https://lvgl.io/

[3] ZedIoT, "LVGL For Embedded HMI: Lightweight And Cross-Platform," 2026. [Online]. Beschikbaar: https://zediot.com/blog/lvgl-embedded-hmi/

[4] Spyrosoft, "Top 5 Embedded HMI UI Frameworks for 2025," 2026. [Online]. Beschikbaar: https://spyro-soft.com/blog/hmi/embedded-ui-frameworks

[5] LVGL, "Keyboard (lv_keyboard) – LVGL 9.5 documentation," 2026. [Online]. Beschikbaar: https://docs.lvgl.io/master/widgets/keyboard.html

[6] LVGL, "Keyboard (lv_keyboard) – LVGL 9.1 documentation," 2026. [Online]. Beschikbaar: https://docs.lvgl.io/9.1/widgets/keyboard.html

[7] LVGL, "Display interface – LVGL documentation," 2026. [Online]. Beschikbaar: https://docs.lvgl.io/9.0/porting/display.html

[8] F&S Elektronik Systeme GmbH, "LVGL Graphics Library for Embedded GUIs," 2026. [Online]. Beschikbaar: https://www.fs-net.de/en/software/lvgl/

[9] Arduino, "UNO Q – Arduino Documentation," 2026. [Online]. Beschikbaar: https://docs.arduino.cc/hardware/uno-q

[10] STMicroelectronics, "Datasheet – STM32U585xx – Ultra-low-power Arm Cortex-M33 32-bit MCU+TrustZone," 2026. [Online]. Beschikbaar: https://www.st.com/resource/en/datasheet/stm32u585ai.pdf

[11] Embedded.com, "Touchscreen GUI Development on Raspberry Pi Pico Using ILI9488 Display, GT911 Capacitive Touch, and LVGL-9.x," 2026. [Online]. Beschikbaar: https://www.embedded.com/touchscreen-gui-development-on-raspberry-pi-pico-using-ili9488-display-gt911-capacitive-touch-and-lvgl-9-x/

[12] LVGL Forum, "Lvgl and touchscreens – How-to," 2026. [Online]. Beschikbaar: https://forum.lvgl.io/t/lvgl-and-touchscreens/22495

[13] Wikipedia, "USB human interface device class," 2026. [Online]. Beschikbaar: https://en.wikipedia.org/wiki/USB_human_interface_device_class

[14] Acroname, "How USB HID Makes Plug-and-Play Devices Work," 2026. [Online]. Beschikbaar: https://acroname.com/blog/how-usb-hid-makes-plug-and-play-devices-work

[15] Adafruit, "HID Devices – Customizing USB Devices in CircuitPython," 2026. [Online]. Beschikbaar: https://learn.adafruit.com/customizing-usb-devices-in-circuitpython/hid-devices

[16] Damnsoft, "USB Adventures Part 2: Custom HID and USB Composite Devices," 2026. [Online]. Beschikbaar: https://blog.damnsoft.org/usb-adventures-part-2-custom-hid-and-usb-composite-devices/

[17] The Linux Kernel, "Introduction to HID report descriptors," 2026. [Online]. Beschikbaar: https://docs.kernel.org/hid/hidintro.html

[18] Adafruit Playground, "A Beginners Guide to writing USB HID Report Descriptors by a Beginner," 2026. [Online]. Beschikbaar: https://adafruit-playground.com/u/Gamblor21/pages/a-beginners-guide-to-writing-usb-hid-report-descriptors-by-a-beginner

[19] ThinkRobotics, "RP2040 vs ESP32: Which is Better? Complete Comparison Guide 2025," 2026. [Online]. Beschikbaar: https://thinkrobotics.com/blogs/learn/rp2040-vs-esp32-which-is-better-complete-comparison-guide-2025

[20] RS Online, "Arduino UNO Q Datasheet," 2026. [Online]. Beschikbaar: https://assets.rs-online.com/image/upload/v1758794466/Datasheets/48f17c00cc500a36dfcd206083f8ff11.pdf

[21] Arduino, "UNO Q User Manual," 2026. [Online]. Beschikbaar: https://docs.arduino.cc/tutorials/uno-q/user-manual

[22] XDA Developers, "I tried Arduino's first Raspberry Pi competitor and it's wonderfully weird," 2026. [Online]. Beschikbaar: https://www.xda-developers.com/arduino-uno-q-wonderfully-weird/

[23] Arduino, "UNO Q Datasheet – ABX00162," 2026. [Online]. Beschikbaar: https://docs.arduino.cc/resources/datasheets/ABX00162-datasheet.pdf

[24] Xecor, "RP2040 vs. ESP32: How to Choose the Right Microcontrollers," 2026. [Online]. Beschikbaar: https://www.xecor.com/blog/rp2040-vs-esp32

[25] All3DP, "Esp32 vs Arduino: The Differences," 2026. [Online]. Beschikbaar: https://all3dp.com/2/esp32-vs-arduino-differences/

[26] Instructables, "Tony Stark Style Keyboard," 2026. [Online]. Beschikbaar: https://www.instructables.com/Tony-Stark-Style-Keyboard/

[27] Arduino Project Hub, "Create a Smartphone-like device with the Arduino UNO R4 WiFi," 2026. [Online]. Beschikbaar: https://projecthub.arduino.cc/AdityaAg/create-a-smartphone-like-device-with-the-arduino-uno-r4-wifi-multiple-apps-keyboard-dynamic-wifi-cloud-sync-95ccec

[28] InterSystems, "Creating and Modifying Dynamic Entities – Using JSON," 2026. [Online]. Beschikbaar: https://docs.intersystems.com/irislatest/csp/docbook/DocBook.UI.Page.cls?KEY=GJSON_create

[29] ArduinoJson, "Efficient JSON serialization for embedded C++," 2026. [Online]. Beschikbaar: https://arduinojson.org/

[30] EmbedMind, "JsonX: Mapping JSON to C Structs on Embedded Systems," 2026. [Online]. Beschikbaar: https://dev.to/embedmind/jsonx-mapping-json-to-c-structs-on-embedded-systems-5e0o

[31] LVGL, "Widget Basics – LVGL 9.3 documentation," 2026. [Online]. Beschikbaar: https://docs.lvgl.io/9.3/details/common-widget-features/basics.html

[32] Arduino StackExchange, "Arduino uno R4 vs ESP32 Dev board," 2026. [Online]. Beschikbaar: https://arduino.stackexchange.com/questions/93669/arduino-uno-r4-vs-esp32-dev-board

[33] Towards Data Science, "Text Generation Using N-Gram Model," 2026. [Online]. Beschikbaar: https://towardsdatascience.com/text-generation-using-n-gram-model-8d12d9802aa0/

[34] D. Jurafsky, "N-gram Language Models," Stanford University, 2026. [Online]. Beschikbaar: https://web.stanford.edu/~jurafsky/slp3/3.pdf

[35] K. Seymore en R. Rosenfeld, "Scalable Trigram Backoff Language Models," Carnegie Mellon University, 1996. [Online]. Beschikbaar: https://www.ri.cmu.edu/pub_files/pub1/seymore_kristie_1996_2/seymore_kristie_1996_2.pdf

[36] NCBI, "Words prediction based on N-gram model for free-text entry in electronic health records," 2026. [Online]. Beschikbaar: https://pmc.ncbi.nlm.nih.gov/articles/PMC6395458/

[37] G. Pibiri, "Tongrams: A C++ library providing fast language model queries in compressed space," GitHub, 2026. [Online]. Beschikbaar: https://github.com/jermp/tongrams

[38] NCBI, "Semi-automated image acquisition and analyses for broad users utilizing macro keyboards," 2026. [Online]. Beschikbaar: https://pmc.ncbi.nlm.nih.gov/articles/PMC12687464/

[39] The Macro Project, "5 benefits of using a macro keyboard for increased productivity," Medium, 2026. [Online]. Beschikbaar: https://medium.com/@themacroproject/5-benefits-of-using-a-macro-keyboard-for-increased-productivity-c799dd4aeac4

[40] EveZone, "Macro Pad vs Software Shortcuts: Which Wins Under Pressure?," 2026. [Online]. Beschikbaar: https://evezone.evetech.co.za/deep-dives/macro-pad-software-shortcuts-better-under-pressure/

[41] Reddit r/elgato, "Elgato Stream Deck vs cheaper macro pads," 2026. [Online]. Beschikbaar: https://www.reddit.com/r/elgato/comments/1p6zelt/elgato_stream_deck_vs_cheaper_macro_pads_is/

[42] YouTube, "I can't believe this is a REAL Keyboard Project!," 2026. [Online]. Beschikbaar: https://www.youtube.com/watch?v=1i5t93G6SFQ

[43] Texas Instruments, "Position Sensing in Keyboard Applications," 2026. [Online]. Beschikbaar: https://www.ti.com/lit/pdf/slya093

[44] Membrane Switches, "Leading Proximity Switch Manufacturers," 2026. [Online]. Beschikbaar: https://membraneswitches.org/proximity-switches/

[45] Microsoft, "Wake on approach – Windows," 2026. [Online]. Beschikbaar: https://learn.microsoft.com/en-us/windows-hardware/design/device-experiences/sensors-presence-wake-on-approach

[46] Microsoft, "Lock on leave – Windows," 2026. [Online]. Beschikbaar: https://learn.microsoft.com/en-us/windows-hardware/design/device-experiences/sensors-presence-lock-on-leave

[47] E. Forsström, "Proximity Sensing Keyboard," 2026. [Online]. Beschikbaar: https://www.erikforsstrom.com/proximity-sensing-keyboard/

[48] IJRESM, "Gesture Control Keyboard," 2026. [Online]. Beschikbaar: https://www.ijresm.com/Vol_1_2018/Vol1_Iss10_October18/IJRESM_V1_I10_175.pdf

[49] Arduino, "UNO Q Power Specifications," 2026. [Online]. Beschikbaar: https://docs.arduino.cc/tutorials/uno-q/power-specification/

[50] Arduino, "Debian Linux Basics for UNO Q," 2026. [Online]. Beschikbaar: https://docs.arduino.cc/tutorials/uno-q/debian-guide

[51] Adafruit, "Adafruit TinyUSB – USB HID stack," GitHub, 2026. [Online]. Beschikbaar: https://github.com/adafruit/Adafruit_TinyUSB_Arduino

[52] Yocto Project, "Yocto Project – open-source embedded Linux build system," 2026. [Online]. Beschikbaar: https://www.yoctoproject.org/

[53] FontAwesome, "FontAwesome – Icon library," 2026. [Online]. Beschikbaar: https://fontawesome.com/

[54] LVGL (LinkedIn), "Post over Starkpad op LVGL LinkedIn-account," 2026.

[55] C. Coward, "Starkpad Is a Gorgeous Stream Deck Alternative You Can Build Yourself," Hackster.io, 28 januari 2026. [Online]. Beschikbaar: https://www.hackster.io/news/starkpad-is-a-gorgeous-stream-deck-alternative-you-can-build-yourself-76cfc961c5f0

[56] Arduino Blog, "This Stream Deck alternative runs on an Arduino UNO Q," 1 februari 2026. [Online]. Beschikbaar: https://blog.arduino.cc/2026/02/01/this-stream-deck-alternative-runs-on-an-arduino-uno-q/

[57] The Bryce Byte, "Starkpad – demonstratievideo," YouTube, 2026. [Online]. Beschikbaar: https://youtu.be/fDFRH98BEmM

[58] J. Blomme, "Starkpad – publieke repository," GitHub, 2026. [Online]. Beschikbaar: https://github.com/BlommeJan/Starkpad

[59] H. R. New, "Immersive workflows for a digital-first industry," gastcollege, Howest, 2026.

[60] J. Decorte, "Sovereign AI – gastcollege," Rebatch.be, Howest, 14 januari 2026.

[61] Arduino, "UNO Q product page," 2026. [Online]. Beschikbaar: https://www.arduino.cc/product-uno-q/

[62] Tom's Hardware, "Arduino Uno Q Review: The board with two brains," 9 december 2025. [Online]. Beschikbaar: https://www.tomshardware.com/raspberry-pi/arduino-uno-q-review

[63] Raspberry Pi Foundation, "Raspberry Pi Zero 2 W product brief," 2026. [Online]. Beschikbaar: https://www.raspberrypi.com/products/raspberry-pi-zero-2-w/

[64] Espressif Systems, "ESP32-S3 datasheet," 2026. [Online]. Beschikbaar: https://www.espressif.com/sites/default/files/documentation/esp32-s3_datasheet_en.pdf

[65] LVGL, "LVGL footprint and benchmarks," 2026. [Online]. Beschikbaar: https://docs.lvgl.io/master/intro/getting_started.html

[66] The Qt Company, "Qt for MCUs documentation," 2026. [Online]. Beschikbaar: https://doc.qt.io/QtForMCUs/

[67] STMicroelectronics, "TouchGFX product page," 2026. [Online]. Beschikbaar: https://www.st.com/en/development-tools/touchgfx-designer.html

8 Bijlages

Onderstaande bijlages bevatten documenten die te omvangrijk waren om in de hoofdtekst op te nemen, maar die bijdragen aan de volledigheid en reproduceerbaarheid van het onderzoek.

8.1 Installatiehandleiding

De installatiehandleiding leidt een nieuwe gebruiker stap voor stap door de hardware-assemblage en de softwareconfiguratie van Starkpad. De volledige en altijd actuele versie is beschikbaar in de publieke repository.

8.1.1 Vereiste hardware

- Arduino UNO Q met de meegeleverde USB-C-kabel en PD-oplader.
- Seeed XIAO RP2040 (ontvangt- en zenddongle).
- Waveshare 12,3 inch capacitief touchscreen met in-cell technology.
- Drie jumperkabels voor TX, RX en GND tussen UNO Q en RP2040.
- 3D-geprinte behuizing volgens starkpad-case-003.stl.
- Volledige componentenlijst met prijzen en leveranciers: zie BOM.md in de repository.

8.1.2 Softwareafhankelijkheden

- Git, CMake, GCC en Python 3 op de ontwikkelingsmachine.
- Arduino IDE voor het flashen van de STM32U585 en de RP2040.
- Een terminal met SSH-toegang tot de UNO Q.

8.1.3 Eerste start

Klonen van de repository en volgen van de detailstappen:

```
git clone https://github.com/BlommeJan/Starkpad.git
cd Starkpad
```

Bij eerste boot van de UNO Q moet de arduino-router service gestopt worden:

```
sudo systemctl stop arduino-router
```

Dit is een tijdelijke beperking die in een toekomstige release wordt geautomatiseerd (zie paragraaf 3.5.3).

8.1.4 Web-UI

Na succesvol opstarten is de configuratie-UI bereikbaar op een browser binnen hetzelfde netwerk:

```
http://starkpad-uno-q.local/
Gebruiker: admin
Wachtwoord: starkpad123
```

Deze standaardreferenties worden bij een eerste productierelease aangepast naar een gebruikerspecifieke combinatie.

8.2 Gebruikershandleiding

8.2.1 Ingebruikname

- Sluit Starkpad aan op een externe USB-PD-voeding.
- Steek de RP2040-ontvangstdongle in een vrije USB-poort van de computer.
- Wacht ongeveer vijftien tot dertig seconden tot de UI verschijnt op het Starkpad-scherm.
- Voor elke eerste boot: voer sudo systemctl stop arduino-router uit via SSH.

8.2.2 De gebruikersinterface

De UI bestaat uit een bovenbalk met vier tabs (Keyboard, Touchpad, Grid en Status) en een dynamisch contentvlak. Een aanraking op een tab wisselt onmiddellijk van modus.

8.2.3 Keyboard mode

Keyboard mode biedt een volledig QWERTY-toetsenbord aan. Een enkele tap stuurt de toetsaanslag door naar de computer. Voor combinaties zoals Ctrl+C houdt de gebruiker de modifier ingedrukt, tikt vervolgens op de gewone toets en laat de modifier daarna los. Beschikbare lay-outs zijn en-US, Colemak en nl-BE, selecteerbaar via het Settings-paneel in de web-UI.

8.2.4 Touchpad mode

In Touchpad mode fungeert het middenvlak als een precisietouchpad waarmee de cursor kan worden verplaatst. De onderste rij bevat twee grote knoppen voor linker- en rechtermuisklik. Tikken op de touchoppervlak is equivalent aan een linkerklik.

8.2.5 Macro grid mode

De macro grid toont per app een raster van zestig programmeerbare knoppen. Selectie gebeurt in twee stappen: eerst een categorie kiezen (Design, Dev, Media,...) en vervolgens een app (Figma, Blender, VS Code,...). Na deze selectie zijn alle knoppen voor die applicatie rechtstreeks bruikbaar.

8.2.6 Configuratie via de web-UI

Alle visuele en functionele aanpassingen gebeuren via de web-UI op http://starkpad-uno-q.local/. Via de tab Keys kunnen knoppen worden toegevoegd, verwijderd of aangepast. De tab Appearance laat toe om thema's en kleuren aan te passen, en de tab Settings bevat algemene instellingen zoals taal en touchpadpositie. Wijzigingen worden pas actief na een klik op Apply Now.

8.2.7 Probleemoplossing

Als knoppen geen invoer opleveren, controleer of de service arduino-router inderdaad gestopt is. Als de web-UI niet bereikbaar is, start de service starkpad-webui.service opnieuw of gebruik het directe IP-adres van de UNO Q. Voor diepere debugging is er een logbestand beschikbaar in /home/arduino/Starkpad/.cursor/debug.log.

8.3 SHIDP – Protocolspecificatie

Serial Human Interface Device Protocol (SHIDP) is het communicatieprotocol dat Starkpad intern gebruikt tussen de Arduino UNO Q en de Seeed XIAO RP2040-dongle. Deze bijlage bevat de volledige specificatie.

8.3.1 Framestructuur

Elk SHIDP-frame bestaat uit exact acht bytes:

```
[SYNC] [TYPE] [ACTION] [P1] [P2] [P3] [P4] [CHECKSUM]
 0xAA   0x01   0x00    ...  ...  ...  ...   XOR(0-6)
```

8.3.2 Veldbeschrijving

- SYNC (byte 0): vaste startmarkering 0xAA. Andere waardes worden door de ontvanger verworpen.
- TYPE (byte 1): 0x01 voor toetsenbord, 0x02 voor muis, 0x03 voor mediatoetsen.
- ACTION (byte 2): 0x00 voor tap (tap is equivalent aan press+release), 0x01 voor hold (press zonder release), 0x02 voor release (losmaken van een vastgehouden toets).
- P1–P4 (bytes 3–6): parametervelden waarvan de betekenis afhangt van TYPE en ACTION. Voor toetsenbordevents bevat P1 de keycode en P2 een modifier-bitmasker (bit 0 = CTRL, bit 1 = SHIFT, bit 2 = ALT, bit 3 = GUI).

- CHECKSUM (byte 7): bitsgewijze XOR van bytes 0 tot en met 6. Bij een mismatch wordt het frame stilzwijgend verworpen.

8.3.3 Ontwerpbeslissingen

- Unidirectionele UART: beperkt de latentie tot één richting en vermijdt de complexiteit van bidirectioneel flow control.
- Vaste framegrootte: vereenvoudigt parsing en maakt het risico op desynchronisatie bij ruis beperkt.
- XOR-checksum: gekozen boven CRC wegens de lage computationele kost. Voldoende voor het detecteren van enkele-bit-fouten op een korte UART-lijn.
- 80 ms watchdog op de ontvanger: vangt verbindingsverlies op. Als gedurende die periode geen geldig frame binnenkomt, worden alle actieve toetsen automatisch losgelaten.
- Aanbevolen baudrate: 460800 bps. Dit biedt voldoende doorvoer voor alle gebruiksscenario's zonder problemen met UART-betrouwbaarheid.

8.3.4 Implementatieoverzicht

In de repository zijn drie relevante bestanden aanwezig voor wie SHIDP wil hergebruiken: SHIDP.h (gedeelde header met alle constanten en helperfuncties), SHIDP_Sender.ino (referentie-implementatie voor de UNO Q) en SHIDP_Receiver_RP2040.ino (referentie-implementatie voor de RP2040).

8.4 Verslagen van gastcolleges

8.4.1 Holly R. New – Immersive workflows for a digital-first industry

Op het moment van het researchproject gaf Holly R. New (CEO en oprichter van STUDIOOFNEW) een gastcollege over de inzet van VR en generatieve AI in moderne ontwerpworkflows, met expliciete aandacht voor de modetechnologie en real-time toepassingen zoals Fortnite-concerten. Een centrale stelling was dat digitale ontwerpworkflows vooral hinder ondervinden van gebrekkige interoperabiliteit tussen tools zoals CLO3D, Browzwear en klassieke CAD-pakketten, en dat VR dit probleem gedeeltelijk kan oplossen op voorwaarde dat het vroeg in het proces wordt ingezet in plaats van als laatste presentatielaag.

Holly positioneerde het gebruik van tools als een functie van het soort denken dat ze ondersteunen: AI breidt de mogelijkheden uit in de ideatiefase, VR verfijnt de selectie via ruimtelijke evaluatie, en CAD-software staat in voor technische afwerking en productieklare output. De bespreking werd ondersteund door case studies over Fortnite-concertoutfits in samenwerking met Copper Candle, trendvoorspelling voor EROBERN Designs en een promotiecampagne voor Decathlon × Belgium Giants.

De relevantie voor Starkpad ligt in de inzichten over cross-disciplinaire samenwerking en workflow-integratie. Holly wees erop dat teamfrictie vaker een gevolg is van uiteenlopende optimalisatiedoelen (ontwerpers voor iteratie, ontwikkelaars voor structuur, managers voor oplevering) dan van tooling. Dit inzicht bevestigt de keuze om Starkpad datagedreven te configureren via JSON, zodat de verschillende stakeholders (eindgebruiker, ontwikkelaar, configureerder) onafhankelijk van elkaar hun aanpassingen kunnen maken zonder de codebase te raken.

8.4.2 Jorge Decorte (Rebatch.be) – Sovereign AI

Op 14 januari 2026 verzorgde Jorge Decorte, oprichter van Rebatch.be, een gastcollege over Sovereign AI. De sessie belichtte het concept van soevereiniteit binnen AI-systemen, met specifieke aandacht voor data-eigendom, infrastructurele controle en juridische verantwoordelijkheid. Jorge koppelde zijn exposé aan meer dan elf jaar professionele ervaring in kunstmatige intelligentie en besprak de technische, juridische en organisatorische implicaties van lokaal uitgerolde AI-oplossingen als alternatief voor cloud-gebaseerde AI-diensten.

Een belangrijk element van het college was de vergelijking tussen gesloten modellen (bereikbaar via betaalde API's) en open-gewicht modellen (waarvan de gewichten lokaal gedownload kunnen worden). Voor organisaties in sectoren met strenge regelgeving (gezondheidszorg, financiële dienstverlening, overheid) bieden lokaal gedraaide modellen het voordeel dat gevoelige data nooit het interne netwerk verlaten. Jorge wees expliciet op juridische risico's zoals de Amerikaanse CLOUD Act, die toegang van Amerikaanse autoriteiten tot Europees opgeslagen data in theorie mogelijk maakt.

Voor Starkpad is deze invalshoek minder direct relevant, aangezien de huidige AI-functionaliteit (N-gram typvoorspelling) volledig lokaal draait en geen data naar de cloud stuurt. Toch bevestigt het college dat lokaal draaiende AI-modellen een legitieme ontwerpkeuze zijn, ook voor kleine embedded apparaten. Als Starkpad in de toekomst wordt uitgebreid met een lokaal taalmodel (bijvoorbeeld een klein open-gewicht model draaiend op de Qualcomm-SoC), liggen de besproken principes rechtstreeks ten grondslag aan de ontwerpbeslissingen.

8.5 Externe vermeldingen van het project

Ter volledigheid worden de belangrijkste externe publicaties over Starkpad hieronder opgesomd:

- LVGL LinkedIn-post (8.034 volgers), gepubliceerd twee maanden na de projectlancering. 75+ reacties, één repost, gedeeld door Jan Blomme.
- Arduino Blog-artikel 'This Stream Deck alternative runs on an Arduino UNO Q' (blog.arduino.cc, 1 februari 2026).
- Hackster.io-artikel 'Starkpad Is a Gorgeous Stream Deck Alternative You Can Build Yourself' (hackster.io/news).
- YouTube-demovideo op het kanaal The Bryce Byte: ongeveer 3.400 weergaven, 40 nieuwe abonnees, 20 commentaren.
- GitHub-repository https://github.com/BlommeJan/Starkpad: 58+ sterren, 4+ forks, publiek zichtbaar.

8.6 Demonstratievideo

Een demonstratievideo van Starkpad in werkende vorm is beschikbaar op het YouTube-kanaal The Bryce Byte. De video toont de drie interactiemodi (keyboard, touchpad, macro grid), de web-UI voor configuratie, en de reactiesnelheid van het systeem in een reële gebruikerscontext.

`https://youtu.be/fDFRH98BEmM`

8.7 Publieke repository en licentie

De volledige broncode van Starkpad is beschikbaar onder de MIT-licentie op de volgende locatie:

`https://github.com/BlommeJan/Starkpad`

De repository bevat de broncode voor de UNO Q-applicatie, de RP2040-firmware, de STM32U585-bridge, de FastAPI-webserver, de JSON-configuratiebestanden, installatiescripts, de STL voor de 3D-geprinte behuizing en de volledige bill of materials. Bijdragen worden verwelkomd via de standaard GitHub-workflow met fork en pull request.

8.8 Bill of materials

Onderstaande tabel geeft een overzicht van de belangrijkste componenten die werden gebruikt voor de bouw van het Starkpad-prototype. De prijzen zijn indicatief en gelden op het moment van publicatie. Voor elke component is een specifieke referentie beschikbaar in het bestand BOM.md in de publieke repository.

Component	Aantal	Functie	Indicatieve prijs
Arduino UNO Q	1	Linux-host en UI-platform	±115 EUR
Seeed XIAO RP2040	1	USB HID-dongle	±7 EUR
Waveshare 12,3 inch touchscreen	1	In-cell capacitief display	±150 EUR
USB-C PD-oplader (min. 20W)	1	Voeding voor de UNO Q	±20 EUR
USB-C-kabel (DP Alt Mode, 1m)	1	Display-uitgang	±12 EUR
Jumperkabels (Dupont female-female)	3	TX, RX en GND tussen UNO Q en RP2040	±2 EUR
3D-geprinte behuizing (PLA of PETG)	1 set	Volgens starkpad-case-003.stl	±8 EUR materiaal
Rubberen antislipvoetjes (optioneel)	4	Stabiliteit op het bureau	±1 EUR

Totaal geraamde kost voor één exemplaar: ongeveer 315 EUR. Dit blijft aanzienlijk onder de prijs van commerciële alternatieven met vergelijkbare specificaties, zoals de Elgato Stream Deck XL of de Loupedeck Live, waarvan de winkelprijs doorgaans tussen 300 en 500 EUR ligt voor een product met een kleiner touchscreen en minder functionaliteit.

8.9 Testprocedure en validatie

Hoewel een uitgebreid, geïnstrumenteerd evaluatieprotocol buiten de scope van het researchproject viel, werden tijdens de ontwikkeling systematisch manuele tests uitgevoerd om de functionele werking te valideren. De hierna gerapporteerde resultaten gelden binnen de beschreven testopstelling en testcondities en zijn als zodanig indicatief eerder dan als een sluitende, objectieve meting bedoeld. Deze paragraaf documenteert de toegepaste testprocedure voor reproduceerbaarheid en als referentie voor toekomstige iteraties.

8.9.1 Hardware-validatie

De volgende tests werden manueel uitgevoerd na assemblage van elke bouwunit:

- Voedingstest: onder de geteste condities functioneren de UNO Q en het scherm stabiel bij een 20 W USB-PD-voeding, met een waargenomen volledige opstarttijd van ongeveer dertig seconden.
- Touchcalibratie: aanraking van de vier hoeken van het scherm levert logische coördinaten binnen een tolerantie van vijf pixels ten opzichte van de daadwerkelijke schermhoek.
- UART-verbinding: bij een baudrate van 460800 bps over een kabel van maximaal dertig centimeter werden in de uitgevoerde tests geen frameontvangstfouten waargenomen op de RP2040.
- USB HID-herkenning: de RP2040-dongle wordt door Windows 10, Windows 11, macOS Sonoma, Ubuntu 24.04 en Fedora 40 onmiddellijk herkend als composite HID-apparaat, zonder installatie van drivers.

-

8.9.2 Functionele tests

Per interactiemodus werden de volgende scenario's getest:

- Keyboard mode: typen van de zin 'The quick brown fox jumps over the lazy dog' in een tekstverwerker; alle karakters verschijnen in de juiste volgorde zonder dubbele of ontbrekende aanslagen.
- Keyboard mode: testen van modifiercombinaties zoals Ctrl+C, Ctrl+V en Alt+Tab in verschillende applicaties; alle combinaties worden correct geregistreerd.
- Touchpad mode: vloeiende cursorbeweging over het volledige scherm van de host-PC, zowel horizontaal, verticaal als diagonaal; de cursor volgt de vingerbeweging zonder merkbare sprongen of vertraging.
- Macro grid mode: uitvoeren van voorgedefinieerde macro's in VS Code (Duplicate Line, Toggle Terminal), Figma (Align Left, Distribute Horizontally) en Blender (Render, Boolean Union); alle macro's werken zoals verwacht.
- Web-UI: aanpassen van een knop via de browser, gevolgd door een klik op 'Apply Now'; de wijziging is onmiddellijk zichtbaar op het apparaat zonder herstart.
- Waargenomen responsiviteit: bij normaal gebruik werd de vertraging tussen een toetsaanraking en de overeenkomstige actie op de host subjectief als onmiddellijk ervaren, ruim onder de drempel van ongeveer honderd milliseconden waarboven vertraging doorgaans merkbaar wordt; de waargenomen vertraging lag naar schatting in de orde van enkele tientallen milliseconden. Het wisselen tussen modi vormde hierop een uitzondering en kon incidenteel oplopen tot ongeveer twee seconden. Deze waarnemingen zijn experiëntieel van aard en gelden binnen de beschreven testopstelling; een geïnstrumenteerde latentiemeting, bijvoorbeeld via een logic analyzer of een HID-monitor op de host, wordt als prioritaire vervolgstap aangemerkt.

8.9.3 Stabiliteits- en robuustheidstests

De volgende scenario's werden specifiek getest om de robuustheid van het watchdog-mechanisme te valideren:

- Fysiek onderbreken van de UART-verbinding tijdens een ingedrukte toets: de toets wordt na 80 ms automatisch losgelaten, zoals ontworpen.
- Bewust verzenden van corrupte SHIDP-frames (SYNC-byte afwezig of checksum onjuist): deze frames worden stilzwijgend genegeerd en veroorzaken geen ongewenst gedrag op de host.
- Langdurige werking (acht uur continu gebruik): gedurende deze testduur werd geen merkbare prestatiedegradatie of geheugenlek vastgesteld. De temperatuur van de UNO Q stabiliseerde op ongeveer 45 graden Celsius, wat binnen de veilige grenzen valt.
- Snelle opeenvolging van toetsaanslagen (stress test, tien per seconde): alle aanslagen worden in de juiste volgorde gerapporteerd zonder verlies.

Een volledig geautomatiseerde testsuite met instrumentatie, bijvoorbeeld via een logic analyzer op de UART-lijn of een software-based HID-monitor op de host, wordt beschouwd als een waardevolle uitbreiding voor toekomstige iteraties van het project.

8.10 Projectplanning en tijdlijn

De ontwikkeling van Starkpad werd gespreid over het academiejaar 2025-2026, parallel met andere modules binnen het MCT-curriculum en met de stage bij Granstudio. De grote mijlpalen zijn:

PERIODE	MIJLPAAL
OKTOBER 2025	Formulering van het contractplan en eerste schetsen van het hardware-ontwerp.
NOVEMBER 2025	Literatuurstudie rond LVGL, USB HID en embedded microcontrollers.
DECEMBER 2025	Voortzetting van de literatuurstudie en selectie/aankoop van hardwarecomponenten.
JANUARI 2026	Technische realisatie: prototype UI (LVGL), USB HID-implementatie, SHIDP-protocol, dual-brain opzet en web-UI (FastAPI). Bijwonen gastcolleges Holly R. New en Jorge Decorte.
FEBRUARI 2026	Publicatie van de demo-video op YouTube en artikelen op de Arduino Blog en Hackster.io. Start van de stage bij Granstudio te Turijn (midden februari).
MAART 2026	Stageperiode bij Granstudio te Turijn en integratie van het project in een professionele context.
APRIL 2026	Evaluatiegesprek met Klaudia Warmus (externe promotor) over de meerwaarde van Starkpad in de designsector. Afronding van de tekst van de bachelorproef.
MEI 2026	Finale voorbereidingen voor de verdediging en laatste optimalisaties aan de documentatie.
JUNI 2026	Finale verdediging en officiële oplevering van de bachelorproef.

8.11 Belangrijke ontwerpbeslissingen

Tijdens de ontwikkeling van Starkpad werden verschillende fundamentele ontwerpbeslissingen genomen die de verdere evolutie van het project beïnvloedden. Onderstaande sectie documenteert de belangrijkste beslissingen volgens een gestandaardiseerd format: context, beslissing en gevolgen. Deze aanpak is geïnspireerd op het Architecture Decision Record-principe uit de softwarearchitectuur, en is bedoeld om toekomstige bijdragers inzicht te geven in het 'waarom' achter de huidige structuur.

8.11.1 Beslissing 1 – LVGL boven alternatieven

Context: er werd een lichte grafische bibliotheek gezocht die op embedded hardware vlot kan draaien en voldoende widgets aanbiedt voor een touchgebaseerde UI. Kandidaten waren LVGL, Qt voor embedded en een eigen implementatie op basis van de framebuffer.

Beslissing: LVGL 9.x werd gekozen wegens de minimale afhankelijkheden, de licentie (MIT), het grote aantal ingebouwde widgets, partial rendering en de actieve community. Qt werd afgewezen wegens de complexe build-omgeving en de licentiekwesties voor commercieel gebruik.

Gevolgen: LVGL biedt voldoende prestaties op de UNO Q, maar vraagt wel een initiële investering in het leren van de widget-hiërarchie en het display-driver-mechanisme. Toekomstige versies kunnen overstappen op LVGL 10 wanneer die uitkomt, zonder dat een volledige herstructurering nodig is.

8.11.2 Beslissing 2 – Dual-brain in plaats van monolithisch

Context: het oorspronkelijke plan voorzag in één board dat zowel de UI als het USB HID-apparaat zou aandrijven. De USB-C-poort van de UNO Q bleek echter niet gelijktijdig beschikbaar voor USB HID wegens de DisplayPort Alt Mode.

Beslissing: een dual-brain-architectuur met een aparte RP2040-dongle als HID-engine werd ingevoerd. De communicatie verloopt via UART met een zelfontworpen protocol.

Gevolgen: de architectuur is complexer maar ook robuuster. De HID-laag blijft deterministisch, zelfs wanneer het Linux-subsysteem belast is. Het protocol (SHIDP) biedt een duidelijk afgebakend contract tussen de twee helften, wat debugging en uitbreiding vereenvoudigt.

8.11.3 Beslissing 3 – JSON-configuratie boven firmware-hardcoding

Context: traditionele QMK-gebaseerde toetsenborden vereisen dat de lay-out in C-code wordt gecompileerd en daarna gefla-sht. Dit vormt een drempel voor eindgebruikers.

Beslissing: alle gebruikersconfiguratie (lay-outs, iconen, kleuren, macro's) wordt opgeslagen in JSON-bestanden en kan worden aangepast via een lokale web-UI.

Gevolgen: eindgebruikers kunnen Starkpad volledig personaliseren zonder ontwikkeltools. De keerzijde is dat JSON-parsing runtime-overhead introduceert, maar deze is verwaarloosbaar op de UNO Q. De firmware hoeft niet opnieuw geflasht te worden bij elke configuratiewijziging.

8.11.4 Beslissing 4 – Open source onder MIT-licentie

Context: het project had een hoge persoonlijke leerwaarde en genereerde uitgebreide documentatie die ook voor anderen nuttig kan zijn.

Beslissing: de volledige broncode, schema's, STL-bestanden en documentatie werden gepubliceerd onder de MIT-licentie op GitHub.

Gevolgen: de open-sourceaanpak heeft geleid tot aanzienlijke externe feedback en validatie, onder meer via de officiële LVGL-community, Hackster.io, de Arduino Blog en YouTube. De licentie laat toe dat zowel hobbyisten als bedrijven het project verder uitbouwen, wat de impact vermenigvuldigt.

8.12 Overzicht van SHIDP-eventtypes

Voor ontwikkelaars die SHIDP willen implementeren of uitbreiden, biedt deze tabel een compact overzicht van alle gedefinieerde eventtypes in de huidige versie van het protocol.

TYPE (hex)	Naam	Parameters	Beschrijving
0x01	KEYBOARD	P1 = keycode, P2 = modifier bitmask	Standaard toetsenbordevents
0x02	MOUSE_MOVE	P1 = dx (signed), P2 = dy (signed)	Relatieve muisbeweging
0x03	MOUSE_BUTTON	P1 = button (0=left, 1=right, 2=middle)	Muisknopacties
0x04	MOUSE_WHEEL	P1 = delta (signed)	Scrollwielbewegingen
0x05	MEDIA	P1 = media keycode	Volume, Play/Pause, Next/Prev
0x06	CONSUMER	P1 = consumer code	Custom consumer control events
0x07	RESET	—	Vraagt de dongle om alle actieve toetsen vrij te geven

Opmerking: niet alle eventtypes zijn volledig geïmplementeerd in de huidige release. TYPE 0x06 (CONSUMER) en 0x07 (RESET) zijn voorzien voor toekomstige uitbreidingen en worden momenteel door de RP2040-firmware herkend maar nog niet actief gebruikt vanuit de UNO Q-kant.

8.13 Overwegingen voor toekomstige uitbreidingen

Op basis van de bevindingen uit de reflectie en op vragen die werden gesteld door de community, worden volgende uitbreidingen overwogen voor een eventuele versie 2.0 van Starkpad:

- Automatische servicestop tijdens boot, zodat de handmatige stap met sudo systemctl stop arduino-router wegvalt. Dit vereist een aangepaste systemd-unit of een replacement van de arduino-router service.
- Ondersteuning voor bidirectionele SHIDP-communicatie, zodat de host-PC status updates kan versturen naar Starkpad (bijvoorbeeld actieve applicatie, volume, Caps Lock-status). Dit vereist een companion-app op de host.
- Bluetooth Low Energy HID als alternatief voor de USB-dongle, waardoor Starkpad draadloos met meerdere apparaten kan communiceren. De UNO Q beschikt reeds over een geschikte Bluetooth-module.
- Uitbreiding van de N-gram-typvoorspelling met een trie-gebaseerde compressieopslag en een groter trainingscorpus per taal. Voor Nederlands is corpus Corpus Gesproken Nederlands een realistische kandidaat.
- Gedeelde cloudprofielen waarbij gebruikers hun JSON-configuraties (naamloos, privacy-vriendelijk) kunnen uploaden en delen met anderen. Implementatie via een optionele public registry.
- Vormfactor-varianten: een compactere versie met een 7 inch scherm voor mobiel gebruik, en een grotere 15 inch variant voor professionele creatieve workstations.

Deze uitbreidingen werden bewust buiten de scope van deze bachelorproef gehouden om de haalbaarheid binnen de beschikbare tijd te garanderen. Ze vormen echter een duidelijk roadmap-document voor wie het project wenst voort te zetten.

9 Dankwoord

Tot slot wil ik mijn oprechte waardering uitspreken aan de personen en gemeenschappen die dit project mogelijk hebben gemaakt. Mijn dank gaat uit naar Johan De Gelas voor de inhoudelijke sturing en kritische feedback tijdens het volledige proces. Ook bedank ik Klaudia Warmus van Granstudio voor het bieden van een professioneel perspectief en de ondersteuning tijdens mijn stage in Turijn.
Daarnaast ben ik de LVGL-community en het team achter de bibliotheek dankbaar voor de internationale erkenning en de technische fundering die zij bieden. Ten slotte bedank ik de makers op platforms zoals GitHub, YouTube en Hackster.io; hun enthousiasme en constructieve feedback waren een drijvende kracht achter de continue verbetering van Starkpad.