

Mikko Kempainen

**GRAAFISEN ETÄKÄYTTÖLIITTYMÄN KEHITTÄMINEN RASPBERRY  
PI:LLE**

Insinööriyö  
Kajaanin ammattikorkeakoulu  
Tekniikan ala  
Tietotekniikan koulutusohjelma  
Kevät 2014



Koulutusala Tekniikan ala	Koulutusohjelma Tietotekniikan koulutusohjelma
Tekijä(t) Mikko Kempainen	
Työn nimi Graafisen etäkäyttöliittymän kehittäminen Raspberry Pi:lle	
Vaihtoehtoiset ammattiopinnot	Ohjaaja(t) Asko Kinnunen Toimeksiantaja KajaPro Oy/Mika Keränen
Aika Kevät 2014	Sivumäärä ja liitteet 46
<p>Tämän insinööriyön toimeksiantaja oli KajaPro Oy, ja työ tuli yrityksen omaan käyttöön. Työn tarkoituksena oli kehittää KajaPro Oy:n graafinen etäkäyttöliittymä Raspberry Pi -sulautetulle laitteelle. Ohjelmistotuotteen kehittämisen vaatimuksena oli, että ohjelmakoodi voidaan kääntää Windows 7 -käyttöjärjestelmästä Raspbian wheezy -käyttöjärjestelmään, jota käytetään Raspberry Pi -laitteen käyttöjärjestelmänä. Lisäksi tällä laitteella piti testata socket-tietoliikenneyhteys. Graafiseen etäkäyttöliittymään piti myös suunnitella ja toteuttaa joystick-komponentti, jota voidaan hyödyntää servomoottoreiden ohjauksessa.</p> <p>Insinööriyön toteuttamiseen käytettiin KajaPro Oy:n graafista etäkäyttöliittymää, jonka resursseja hyödyntäen joystick-komponentti suunniteltiin ja toteutettiin. Komponenttia, jonka avulla ohjattiin servomoottoreita, testattiin Android-käyttöjärjestelmän sisältävässä mobiililaitteessa. Ohjelmistokehitysympäristönä käytettiin Microsoft Visual Studio 2010 -ohjelmaa. Insinööriyö alkoi joystick-komponentin suunnittelulla ja komponentin toiminnan ohjelmoinnilla C++-ohjelmointikielellä. Ristikääntäjää varten täytyi tehdä tutkimustyötä siitä, kuinka ohjelmakoodi saadaan käännettyä toiselta laitteistoarkkitehtuurilta toiselle. Kääntäjä, joka sisältää GNU-projektin työkaluja, asennettiin Windows 7 -ympäristöön.</p> <p>Raspberry Pi tarvitsee EGL-rajapintaa grafiikan esittämiseen ja SDL-rajapintaa näppäimistön ja hiiren toiminnallisuuteen. Kääntäjä tarvitsee useita kirjastoja, jotka pitävät sisällään EGL-, OpenGL ES 2- ja SDL-koodien menet. Nämä kirjastot piti konfiguroida ja kääntää Raspberry Pi:llä ja siirtää Windows 7:lle. Kirjastojen täytyy olla linkitettyinä linkkerin avulla, jotta kääntäjä löytää ohjelmakoodin tarvitsemat toiminnallisuudet. KajaPro Oy:n graafinen etäkäyttöliittymä voidaan kääntää Raspberry Pi:lle, kun kehitysympäristö on saatu kokoon. Ristikääntäjä, joka kääntää ohjelmistotuotteen koodin suoritettavaksi tiedostoksi, toteutettiin tätä varten. Suoritettava tiedosto voidaan käynnistää vain Raspbian wheezy -käyttöjärjestelmällä.</p> <p>Tämän insinööriyön lopputulos oli onnistunut. Ristikääntäjä kääntää ohjelmakoodin haluttuun muotoon ja KajaPro Oy:n ohjelmistotuote käynnistyy Raspberry Pi:llä. Socket-tietoliikenneyhteysttä testattiin kahden eri Raspberry Pi:n avulla ja serveriltä lähetetty graafinen käyttöliittymä saapui perille molemmille laitteille. Tämä insinööriyö on hyödyllinen niille, jotka tarvitsevat tietoa, kuinka lähdekoodi voidaan kääntää Windows 7 -ympäristöstä Raspbian wheezy -ympäristöön.</p>	
Kieli	Suomi
Asiasanat	Raspberry, ristikääntäjä, GNU, SDL, EGL, joystick
Säilytyspaikka	<input checked="" type="checkbox"/> Verkkokirjasto Theseus <input type="checkbox"/> Kajaanin ammattikorkeakoulun kirjasto

School Engineering	Degree Programme Information Technology
Author(s) Mikko Kempainen	
Title Developing The Remote Graphical User Interface for Raspberry Pi	
Optional Professional Studies	Instructor(s) Asko Kinnunen
	Commissioned by KajaPro Oy/Mika Keränen
Date Spring 2014	Total Number of Pages and Appendices 46
<p>This Bachelor's thesis was commissioned by KajaPro Oy and was made for the sole use of the company. The purpose of work was to develop the company's remote graphical user interface to Raspberry Pi embedded device. The development of the software demanded that the program code could be compiled from Windows 7 to Raspbian wheezy, which is in this case the operating system of the embedded device. A socket data communication had to be tested in the embedded system. Also, a joystick component had to be designed and made programmatically to the software of the company.</p> <p>The thesis started by programming the joystick component with programming language C++. The application that was used in writing the code was Microsoft Visual Studio 2010. The socket data communication was tested with Hercules utility in Windows 7. Research work had to be done in order to know how to compile code from different software architecture to another. The compiler which uses GNU is Not Unix (GNU) project toolchain was installed to the development environment. Raspberry Pi needs Interface between Khronos Group rendering (EGL) to show graphics and Simple DirectMedia Layer (SDL) to understand keyboard and mouse functionalities in the system. The compiler needs several libraries which hold the methods to the Open Graphics Layer for Embedded Systems 2 (OpenGL ES 2) and SDL. These interfaces had to be configured and built in the embedded device and then moved to the target operating system. There they had to be included and linked against in order to the compiler to find the required functions.</p> <p>The result of this thesis was successful. After the development environment was established the source code of the company could be compiled to the operating system needed. The cross-compiler compiles the code to an executable file and the software of KajaPro Oy runs in Raspberry Pi. The socket data communication was tested with two of these embedded systems and the graphical user interface from server was received in both devices. This thesis is useful for those who need to know how to compile code from Windows 7 environment to Raspbian wheezy environment.</p>	
Language of Thesis	Finnish
Keywords	Raspberry, cross-compiler, GNU, SDL, EGL, joystick
Deposited at	<input checked="" type="checkbox"/> Electronic library Theseus <input type="checkbox"/> Library of Kajaani University of Applied Sciences

## ALKUSANAT

Haluan kiittää KajaPro Oy:tä tämän insinööriyön toimeksiannosta. Kiitokset työn ohjaajille, Mika Keräselle KajaPro Oy:stä ja Asko Kinnuselle Kajaanin ammattikorkeakoulusta, jotka ohjeistivat minua tämän dokumentin sisällön suunnittelussa ja tarkastamisessa. Kiitokset suomen kielen opettajalleni Eero Soiniselle Kajaanin ammattikorkeakoulusta tämän dokumentin kielellisestä ohjauksesta. Kiitokset myös Seija Heikkiselle englanninkielisen abstraktin kielellisestä ohjauksesta.

Kajaani, huhtikuu 2014

Mikko Kemppainen

## SISÄLLYS

1 JOHDANTO	1
2 VAATIMUSMÄÄRITTELY	3
3 RASPBERRY PI	5
3.1 Raspbian wheezy -käyttöjärjestelmä	8
3.2 Ominaisuudet	9
3.3 Ohjelmiston arkkitehtuuri	11
4 EGL-RAJAPINTA	13
5 SDL-RAJAPINTA	14
5.1 SDL-rajapinnan asennus Raspberrylle	14
5.2 Kirjastojen ja header-tiedostojen käyttöönotto ristikäntäjässä	15
6 OHJELMAKOODIN KÄÄNTÄMINEN	16
6.1 GCC-kääntäjä	16
6.1.1 GDB-työkalu	16
6.1.2 Ristikääntäjä	17
6.2 C- ja C++-koodien kääntäminen	18
6.2.1 Objektitiedostot, linkkeri ja tarvittavat kirjastot	19
6.2.2 Suoritettavan tiedoston luominen	19
7 KEHITYSYMPÄRISTÖN KOKOONPANO	20
7.1 Ristikääntäjän asennus GNU-työkalun avulla	21
7.2 SSH-yhteyden luominen	25
8 JOYSTICK-KOMPONENTTI	30
8.1 Komponentin suunnittelu	30
8.2 Komponentin toteutus	32
8.3 Komponentin testaus	33
9 SOCKET-YHTEYS	34
10 TESTAUS	37

11 TYÖN ANALYSOINTI	40
12 YHTEENVETO	42
LÄHTEET	43

## SYMBOL LIST

ABI	Application Binary Interface
API	Application Programming Interface
ARM	Advanced RISC Machines
EGL	Interface between Khronos Group rendering (OpenGL, OpenGL ES, OpenVG)
GCC	GNU Compiler Collection
GDB	GNU Project Debugger
GNU	“Gnu’s Not Unix!”
GPIO	General Purpose Input/Output
GPU	General Processing Unit
HDMI	High Definition Multimedia Interface
HTTP	Hypertext Transfer Protocol
I2C	Inter-Integrated Circuit
OpenGL ES 2	Open Graphics Library for Embedded Systems 2
SDL	Simple DirectMedia Layer
SoC	System on Chip
SPI	Serial Peripheral Interface
SSH	Secure Shell
TCP	Transmission Control Protocol
UART	Universal Asynchronous Receiver Transmitter
UDP	User Datagram Protocol

## 1 JOHDANTO

Raspberry Pi on Raspberry Pi Foundationin kehittämä hyvin pieni tietokone, jota ohjelmisto- ja elektroniikkakehittäjät voivat hyödyntää erilaisissa projekteissa. Vuosina 2006–2008 kyseisestä laitteesta oli kehitelty useita eri prototyyppejä, joista syntyi Raspberry Pi. Tähän sulautettuun laitteeseen voidaan asentaa käyttöjärjestelmä Raspbian wheezy, NOOBS (New Out of Box Software), NOOBS Lite, RaspBMC, Pidora, OpenELEC (Open Embedded Linux Entertainment Center), RISC (Reduced Instruction Set Computer) tai Arch [1]. Raspbian wheezy -käyttöjärjestelmä sisältää ohjelmakoodin kääntämiseen tarkoitetut GNU-projektin (GNU is Not Unix!) työkalut. [2.] [3.]

Tietokonesovellusten kehittäjä voi käyttää GNU-työkaluja Raspberry Pi:llä kääntämään C- ja C++-ohjelmointikielillä kirjoitettuja koodeja suoritettaviksi tiedostoiksi, jotka toimivat kyseisessä laitearkkitehtuurissa.

KajaPro Oy on vuonna 2004 perustettu tietotekniika-alan yritys. Yritys on erikoistunut mobiililaitteiden -, sulautettujen järjestelmien - sekä tietoliikenteen ohjelmistokehitykseen. Yritys tarjoaa myös valmennus- ja asiantuntijapalveluita muille yrityksille.

Suurien ohjelmistotuotteiden, kuten KajaPro Oy:n graafisen etäkäyttöliittymäsovelluksen, kääntäminen Raspberry Pi:llä on erittäin hidasta, joten tämä prosessi olisi nopeampaa suorittaa tehokkaammalla tietokoneella esimerkiksi Windows-käyttöjärjestelmän sisältävässä PC:ssä. Näin ollen sovelluksen versionhallinta on myös helpompaa, kun lähdekoodi voidaan säilyttää yhdessä paikassa. Tämä prosessi vaatii ristikäntäjän käyttöä Windowsilla. [2.] [3.]

Tämän insinööriyön tarkoituksena on toteuttaa ristikäntäjä tarvittavineen rajapintoineen Windows 7 -käyttöjärjestelmän sisältävälle tietokoneelle. Ristikääntäjää käytetään kääntämään sovellus usealle eri laitteistoarkkitehtuurille. Tässä työssä ristikäntäjää käytetään Windows 7 -käyttöjärjestelmällä kääntämään KajaPro Oy:n graafinen etäkäyttöliittymäsovellus toimimaan Raspbian wheezy -käyttöjärjestelmässä. Sovelluksen käyttöönotto Raspberry Pi:llä vaatii sovelluksen toiminnallisuuden ja grafiikan esittämisen kannalta tutkimustyötä SDL- (Simple DirectMedia Layer) ja EGL-rajapinnoista (Interface between Khronos Group rendering).

SDL-rajapinnan avulla voidaan toteuttaa tietokoneen hiiren ja näppäimistön toiminnallisuus [4]. Nämä ominaisuudet on toteutettu valmiiksi yrityksen sovelluksessa kyseisellä



rajapinnalla. EGL-rajapintaa käytetään tuottamaan OpenGL ES 2 (Open Graphics Library for Embedded Systems 2) -ohjelmointirajapinnalla toteutettua grafiikkaa [5]. Tämän rajapinnan implementointi pitää toteuttaa KajaPro Oy:n graafiseen etäkäyttöliittymään.

Ristikääntäjän lisäksi KajaPro Oy:n graafiseen etäkäyttöliittymään on tarkoitus toteuttaa joystick-komponentti yrityksen laiteriippumattoman ohjelmistokehityksen avulla. Komponentin toteutukseen käytetään ohjelmistokehityksen muiden komponenttien runkoa. Raspberry Pi:llä on tarkoitus myös testata yrityksen sovelluksen sisältämä socket-tietoliikenneyhteys hercules-testityökalun avulla.

Tämän työn vaiheisiin kuuluvat tutkimustyön tekeminen, suunnittelu, toteutus, testaus ja dokumentointi edellä mainittujen vaiheiden rinnalla.

## 2 VAATIMUSMÄÄRITTELY

Tämän insinööriyön tarkoituksena on kehittää KajaPro Oy:n graafiseen etäkäyttöliittymään tuki Raspberry Pi:lle käyttäen ristikäntäjää. Tämän lisäksi etäkäyttöliittymään on tarkoitus toteuttaa yleiskäyttöinen joystick-komponentti, jota voidaan käyttää esimerkiksi servomootoreiden ohjaukseen. Joystick-komponenttia pitää pystyä liikuttamaan sen x- ja y-akseleiden suhteen etäkäyttöliittymän määrittelemissä laitteissa.

Ristikääntäjän käyttöönotto ja kehitysympäristön kokoonpano on tarkoitettu KajaPro Oy:n uusien työntekijöiden ohjeistukseksi. Etäkäyttöliittymä täytyy pystyä ristikäntämään GCC:n (GNU Compiler Collection) avulla Windows 7 -käyttöjärjestelmällä vastaamaan Raspbian wheezy -käyttöjärjestelmän ohjelmistoarkkitehtuuria. Ristikääntäminen pitää voida suorittaa suoraan Windowsin komentoriviltä yksinkertaisella komennolla. Ohjelmistotuote tarvitsee EGL- ja SDL-rajapintojen tuen, sekä ristikäntäjä tarvitsee kirjastot kyseisistä rajapinnoista toimiakseen Raspberry Pi:llä.

Raspberry Pi:llä toimivasta graafisesta etäkäyttöliittymästä pitää pystyä muodostamaan socket-yhteys toiselle laitteelle, esimerkiksi internetselaimessa toimivaan HTTP-sivustoon (Hypertext Transfer Protocol) tai lähiverkossa toimivaan hercules-testityökaluun, joka voidaan ladata ilmaiseksi Windows 7:lle. Socket-tietoliikenneyhteydessä serverinä toimii mikä tietokone tahansa. Tämän työn testausympäristössä serverinä on käytetty Windows 7 -käyttöjärjestelmää ja asiakassovelluksena Raspberry Pi -laitetta. Asiakassovellus suorittaa graafisen etäkäyttöliittymäsovelluksen ja ottaa yhteyden serveriin. Yhteyden muodostuksessa asiakasohjelma lähettää serverille yhteyspyynnön komennolla <REQ>, johon serveri vastaa komennolla <RESP>. Kun yhteys on saatu muodostettua, laite tai socket-yhteyden päällä toimiva HTML-sivusto (HyperText Markup Language) lähettää käyttöliittymän Raspberry Pi:lle. Asiakasohjelma kuittaa serveriltä saadun komennon viestillä <OK>. Etäkäyttöliittymä tulostaa koko ajan tietoa kohdelaitteessa toimivalle komentoriville. Taulukossa 1 on esitetty insinööriyölle asetetut vaatimukset.

Taulukko 1. Vaatimusmäärittelyt

Vaatimus nro.	Vaatimus	Kuvaus
1	Windows 7	Ohjelmistotuotteen ristikäntämiseen vaaditaan Windows 7 -käyttöjärjestelmä.
2	Raspberry Pi	Kehitysympäristön kokoonpanoon tarvitaan Raspberry Pi.
3	Raspbian wheezy	Raspberry Pi vaatii Raspbian wheezy -käyttöjärjestelmän.
4	EGL-rajapinta	Etäkäyttöliittymän grafiikka esitetään Raspberry Pi:llä EGL-rajapinnan avulla.
5	EGL-kirjasto	Ristikääntäjän linkkeri vaatii EGL-kirjaston toteutuksen EGL-koodin metodien suorittamista varten.
6	SDL-kirjasto	Ristikääntäjän linkkeri vaatii SDL-kirjaston toteutuksen SDL-koodin metodien suorittamista varten.
7	bat-tiedostot	KajaPro Oy:n etäkäyttöliittymä ristikäännetään bat-tiedostojen avulla Windowsin komentoriviltä.
8	Socket-yhteys	Etäkäyttöliittymä voidaan lähettää socket-yhteyden avulla Windows-käyttöjärjestelmältä Raspberry Pi:lle.
9	x- ja y-akselit	Joystick-komponenttia pystyy liikuttamaan graafisen etäkäyttöliittymän tukemissa alustoissa korkeus- ja leveysuunnassa.

### 3 RASPBERRY PI

Vuonna 2006 Raspberry Pi:n kehittäjät Eben Upton, Rob Mullins, Jack Lang ja Alan Mycroft Cambridgen yliopistosta huolestuivat siitä, että tietojenkäsittelytieteen koulutusohjelmaan hakeneilla oppilailla ei ollut paljoakaan tietoa tietokoneen toiminnasta, niiden rakenteesta tai ohjelmoinnista. Oppilaat eivät olleet kiinnostuneita tietojenkäsittelytieteestä, vaan pitivät sitä suorastaan tylsänä ja latteana. Niinpä kehittäjät alkoivat suunnitella alustaa, joka ei olisi kallis ja jota nuoret pystyivät käyttämään ohjelmointiympäristönä. Vuosina 2006–2008 kehittäjät suunnittelivat useita eri prototyyppisiä laitteesta, joka tunnetaan nykypäivänä Raspberry Pi:nä. [2.] [6.]

Vuoteen 2008 mennessä puhelimet sisälsivät tarpeeksi tehokkaita prosessoreita, joiden avulla voitiin tuottaa hyvänlaatuista multimediaa. Tämän ominaisuuden avulla kehittäjät ajattelivat toteuttaa alustan nuorille, jotka ovat kiinnostuneita ohjelmoitavasta laitteesta. Näin ollen kehittäjät Eben Upton, Rob Mullins, Jack Lang ja Alan Mycroft ryhtyivät työskentelemään Pete Lomasin ja David Brabenin kanssa perustaen Raspberry Pi Foundationin marraskuussa 2008. Raspberry Pi Foundation on Yhdistyneessä kuningaskunnassa rekisteröity hyväntekeväisyysjärjestö, jonka tarkoituksena on edesauttaa lasten ja aikuisten koulutuksen kehittämistä tietojenkäsittelytieteissä, tietokoneiden ja niihin liittyvien koulutusaineiden kanssa [7]. [2.] [6.]

Raspberry Pi mallia B alettiin valmistaa massatuotantona vuonna 2011 lisenssivalmistuksena Premier Farnell/Element 14 - ja RS Electronics -yhtiöiden kautta, ja vuodessa kyseistä mallia oli myyty yli miljoona kappaletta. Kehitysmaat ovat kiinnostuneista Raspberry Pi:stä sen edullisen hinnan vuoksi erityisesti alueilla, joilla ei ole varaa tehokkaaseen PC-laitteistoon. Esimerkiksi sairaalat ja museot ovat ottaneet yhteyttä kehittäjiin osoittaen mielenkiintoa, kuinka Raspberry Pi:llä voidaan ohjata erilaisia näyttölaitteita. [2.]

Vaikeavammaisten lasten vanhemmat ovat keskustelleet kehittäjien kanssa seuranta- ja helppokäyttösovelluksista, ja yli miljoona ihmistä haluaisikin rakentaa robotin käyttäen Raspberry Pi:tä. Kehittäjät haluavat nähdä halpoja, ohjelmoitavia tietokoneita kaikkialla, ja he rohkaisevat aktiivisesti muita yrityksiä toteuttamaan samankaltaisia laitteita. [2.]

Tästä pienestä luottokortin kokoisesta tietokoneesta on olemassa myös malli A. Molemmat mallit sisältävät useita eri liitännämahdollisuuksia pienestä koostaan riippumatta. B-malli

tarjoaa muun muassa enemmän liitäntöjä ja muistia. Taulukossa 2 on esitetty A- ja B-mallien teknillisiä ominaisuuksia. [8.]

Taulukko 2. Raspberry Pi -mallien A ja B teknilliset ominaisuudet [8].

Ominaisuus	Malli A	Malli B
Chip	Broadcom BCM2835 SoC full HD multimedia applications processor	Broadcom BCM2835 SoC full HD multimedia applications processor
CPU	700 MHz Low Power ARM1176JZ-F Applications Processor	700 MHz Low Power ARM1176JZ-F Applications Processor
GPU	Dual Core VideoCore IV® Multimedia Co-Processor	Dual Core VideoCore IV® Multimedia Co-Processor
Memory	256 MB SDRAM	512 MB SDRAM
Ethernet	None	Onboard 10/100 Ethernet RJ45 jack
USB 2.0	Single USB Connector	Dual USB Connector
Video Output	HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC)	HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC)
Audio Output	3.5 mm jack, HDMI	3.5 mm jack, HDMI
Onboard Storage	SD, MMC, SDIO card slot	SD, MMC, SDIO card slot
Operating System	Linux	Linux
Dimensions	8.6 cm x 5.4 cm x 1.5 cm	8.6 cm x 5.4 cm x 1.7 cm

Raspberry Pi -laitteeseen voi liittää TV:n HDMI-kaapelin (High Definition Multimedia Interface) avulla, sekä näppäimistön ja hiiren USB-liitäntöjen (Universal Serial Bus) avulla, jotka mahdollistavat pienikokoisen kehitysympäristön kokoonpanon. Raspberry Pi käyttää Broadcom BCM2835 SoC (System on Chip) -piiriä ARM1176JZF-S (Advanced RISC Machines) 700 MHz:n prosessorilla. Laitteen käyttöjärjestelmä asennetaan ulkoiselle SD-kortille (Secure Digital). [8.]

Kuten normaalilla tietokoneella, myös Raspberry Pi:llä käyttäjä pystyy käsittelemään kuvaa ja tekstiä. Internet-yhteyden lisäksi käyttäjän on mahdollista pelata pelejä ja katsoa videoita tämän pienen sulautetun laitteen avulla. Raspberry Pi:tä pystyy kehittämään ja muokkaamaan

ohjelmallisesti erilaisiin käyttötarkoituksiin, kuten palvelintoimintaan tai X-Box Media Center -ohjelman käyttöön. [9.]

GPIO-pinnejä (General Purpose Input/Output) Raspberry Pi:ltä löytyy 26 kappaletta, joiden käyttötarkoitukset Raspberry Pi -mallin B versiossa 1 ja 2 on esitetty taulukossa 3. Niitä hyödyntämällä laitetta voidaan käyttää monenlaisissa elektroniikan projekteissa. Laite tarjoaa mahdollisuuden tiedonsiirrossa I2C- (Inter-Integrated Circuit), SPI- (Serial Peripheral Interface) tai UART-väylään (Universal Asynchronous Receiver Transmitter). GPIO-pinnien käyttöjännite on 3,3 V, eikä piirillä ole suojausta ylikuormitusta varten. Näiden lisäksi piikkirimastolla on käyttöjännitteet 3,3 V ja 5 V sekä neljä eri maapinniä. [10.]

Taulukko 3. Raspberry Pi GPIO-pinnien käyttötarkoitukset. [10.]

GPIO-pinnin numero	Käyttötarkoitus	GPIO-pinnin numero	Käyttötarkoitus
1	3,3 V jännite	14	Maa
2	5 V jännite	15	GPIO 22
3	Versio 1: GPIO 0 (SDA) Versio 2: GPIO 2 (SDA)	16	GPIO 23
4	5 V jännite	17	3,3 V jännite
5	Versio 1: GPIO 1 (SCL) Versio 2: GPIO 3 (SCL)	18	GPIO 24
6	Maa	19	GPIO 10 (MOSI)
7	GPIO 4 (GPCLK0)	20	Maa
8	GPIO 14 (TXD)	21	GPIO 9 (MISO)
9	Maa	22	GPIO 25
10	GPIO 15 (RXD)	23	GPIO 11 (SCLK)
11	GPIO 17	24	GPIO 8 (CE0)
12	GPIO 18 (PCM_CLK)	25	Maa
13	Versio 1: GPIO 21 Versio 2: GPIO 27	26	GPIO 7 (CE1)

### 3.1 Raspbian wheezy -käyttöjärjestelmä

Raspberry Pi -sulautetulle laitteelle voidaan asentaa ilmainen Raspbian wheezy, joka on Debian wheezy -käyttöjärjestelmään perustuva Raspberry Pi:lle sopivaksi optimoitu käyttöjärjestelmä. Raspbian wheezy -käyttöjärjestelmän toteutti pieni kehittäjäryhmä, joka on kiinnostunut Raspberry Pi -laitteesta, Raspberry Pi -säätöön koulutuksellisista tavoitteista sekä Debian-projektista. Käyttöjärjestelmän mukana tulee yli 35000 ohjelmistopakettia, jotka ovat valmiiksi esikäännettyinä vastaamaan Raspberry Pi:n vaatimuksia. Käyttäjä voi halutessaan asentaa valmiita paketteja ja hyödyntää niitä omissa projekteissaan. [3.]

Debian on vapaa tietokoneen käyttöjärjestelmä, joka sisältää useita työkaluja ja sovelluksia. Käyttöjärjestelmää kehittävät ja tukevat monet ohjelmoijat eri puolilla maailmaa. Debianin ytimenä toimii kernel joko Linux- tai FreeBSD-ydin. [11.]

Debian wheezyn versio 7.4 julkaistiin 8.2.2014, ja se tukee useita eri tietokonearkkitehtuuria, kuten:

- 64-bittinen PC (amd64)
- 32-bittinen PC (i386)
- EABI ARM
- PowerPC
- Hard Float ABI ARM
- SPARC
- kFreeBSD 64-bittinen PC (amd64)
- Intel Itanium IA-64
- MIPS (little endian)
- kFreeBSD 32-bittinen PC (i386)
- IBM S/390
- MIPS (big endian)
- IBM System z [12.]

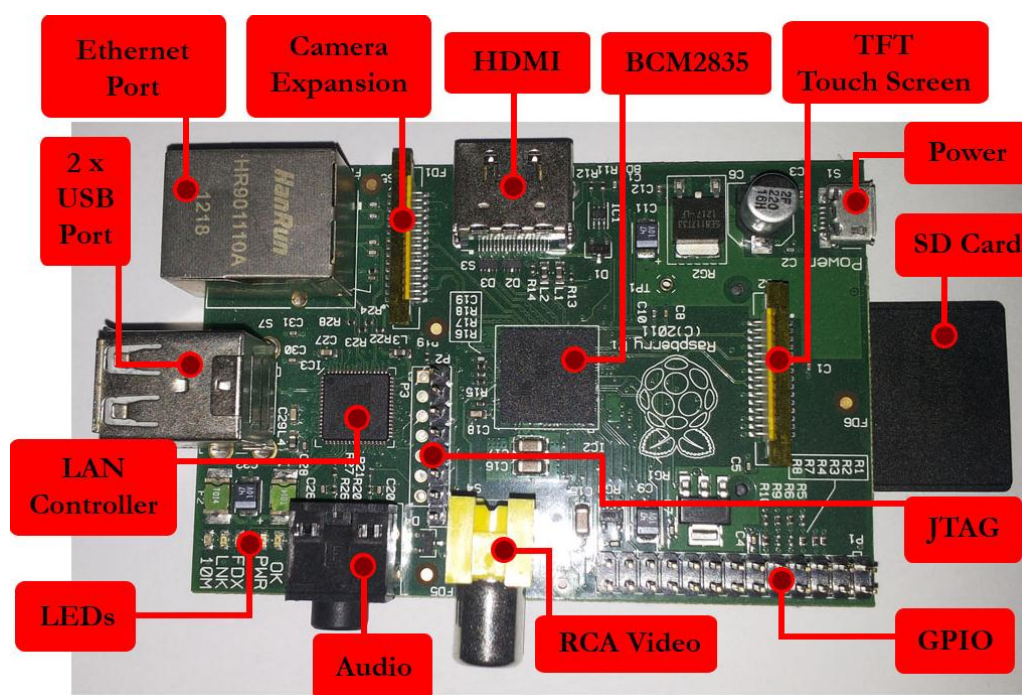
Pääasiassa Raspbian wheezy käyttää Raspberry Pi -laitteen liukulukuja ja hard float ABI:a (Application Binary Interface) mikroprosessoriarkkitehtuurilla ARMv6. Hard float ABI:lla tarkoitetaan tässä tapauksessa sääntöjä, joita käytetään, kun ollaan tekemisissä laitteen rekistereiden ja pino-tietorakenteen kanssa. Hard float ABI asettaa liukulukuparametrit liukulukurekistereihin, kun taas soft float ABI asettaa samat parametrit

kokonaislukurekistereihin. Näin ollen nämä kaksi ABI:a eivät ole yhteensopivia keskenään, koska ne eivät käytä samoja rekistereitä. Hard float ABI:a on mahdollista käyttää soft float ABI:n kanssa, mutta tällöin hard float ABI:n liukuluku pitää siirtää kokonaislukurekisteriin, jotta se ei aiheuta haittaa laitteen suorituskyvylle. [13.]

Raspbian wheezy -käyttöjärjestelmää voidaan käyttää pelkän komentorivin kautta ilman graafista käyttöliittymää. LXDE (Lightweight X11 Desktop Environment) on Raspbianissa käytettävä työpöytä, joka kuitenkin mahdollistaa samanlaiset hiiren ja näppäimistön toiminnot, niin kuin muissakin graafista käyttöliittymää käyttävissä käyttöjärjestelmissä. Raspbian sisältää myös monia perusohjelmia, kuten laskimen, tekstinkäsittelyohjelman Leafpad, komentorivit LXTerminal ja Root Terminal, kuvankäsittelyohjelman Image Viewer, sekä tiedostojen pakkaukseen ja purkamiseen tarkoitettua Xarchiver-ohjelman. [14, s. 45–46.]

### 3.2 Ominaisuudet

Raspberry Pi tarjoaa useita eri ominaisuuksia kehitysympäristön kokoonpanoon. Kuvassa 1 esitettyyn Raspberry Pi -laitteen malli B:hen voidaan kytkeä valmiiksi esimerkiksi Raspbian wheezy -käyttöjärjestelmän sisältävä SD-kortti, tietokoneen näppäimistö ja hiiri laitteen USB-portteihin sekä tietokoneen näyttö HDMI-liitäntään.

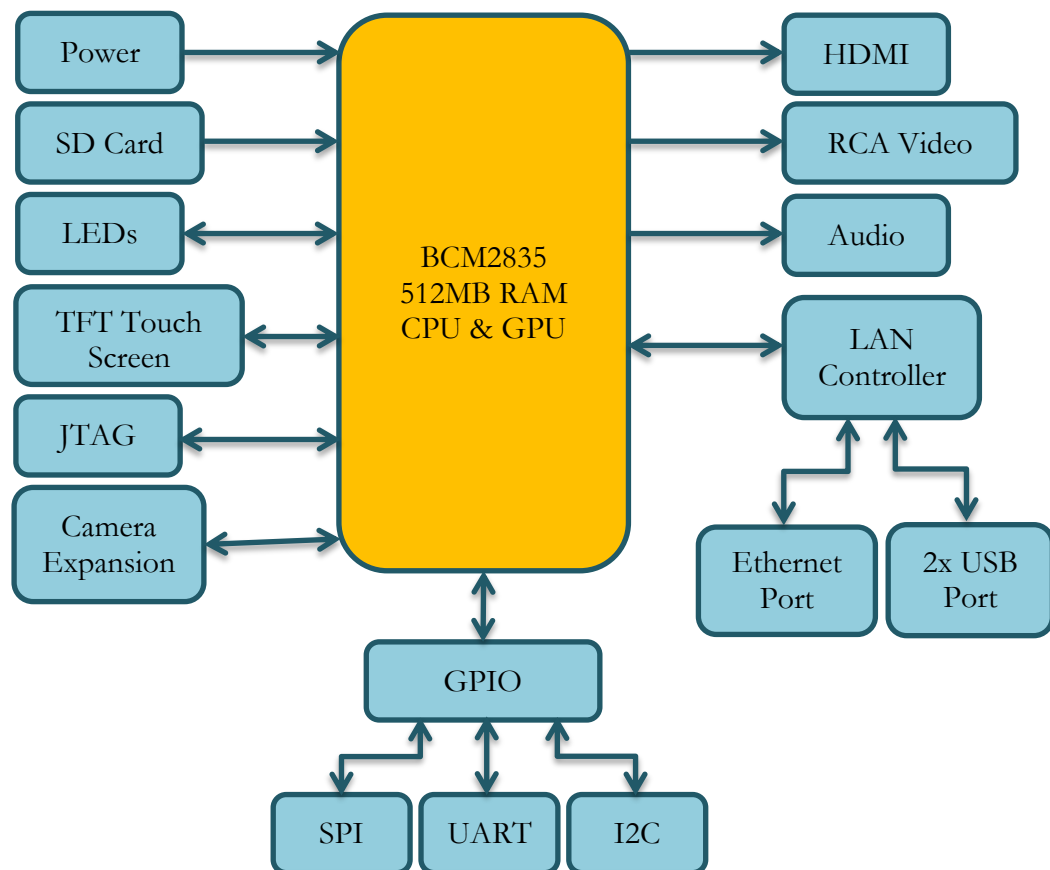


Kuva 1. Raspberry Pi, malli B:n versio 1 [15].



Kuvassa 2 on esitetty lohkokaavio Raspberry Pi -laitteen malli B:n ominaisuuksista. Raspberry Pi voidaan kytkeä micro USB -kaapelilla tietokoneeseen, josta se saa virran. Laite voidaan myös kytkeä suoraan verkkojännitteeseen lisäämällä micro USB -kaapelin päähän AC-adapteri, joka muuntaa jännitteeksi viisi voltia. Raspberry Pi käyttää 5 V:n jännitettä, malli B 700–1000 mA:n virralla ja malli A 500–1000 mA:n virralla [16]. Raspberry Pi:n käyttöjärjestelmä asennetaan ulkoiselle SD-kortille, joka kytketään Raspberry Pi -laitteeseen.

Audio-liittimeen voi kytkeä erilliset kaiuttimet tai kuulokkeet. RCA Video -liittimeen kytketään analoginen TV tai näyttö jos TV:tä tai näyttöä ei ole kytketty HDMI-liitäntään. GPIO-pinnien avulla Raspberry Pi:tä voidaan käyttää erilaisissa elektroniikan projekteissa ja ne mahdollistavat SPI-, UART- ja I2C-tietoliikenneväylien käytön. Lisäksi laitteeseen voidaan kytkeä erillinen TFT-kosketusnäyttö (Thin Film Transistor) eli TFT Touch Screen, sekä kamera eli Camera Expansion CSI-liitännällä (Camera Serial Interface). Raspberry Pi -laitetta ja sen ohjelmistoa voidaan testata JTAG-liitännällä (Joint Test Action Group). [15.]



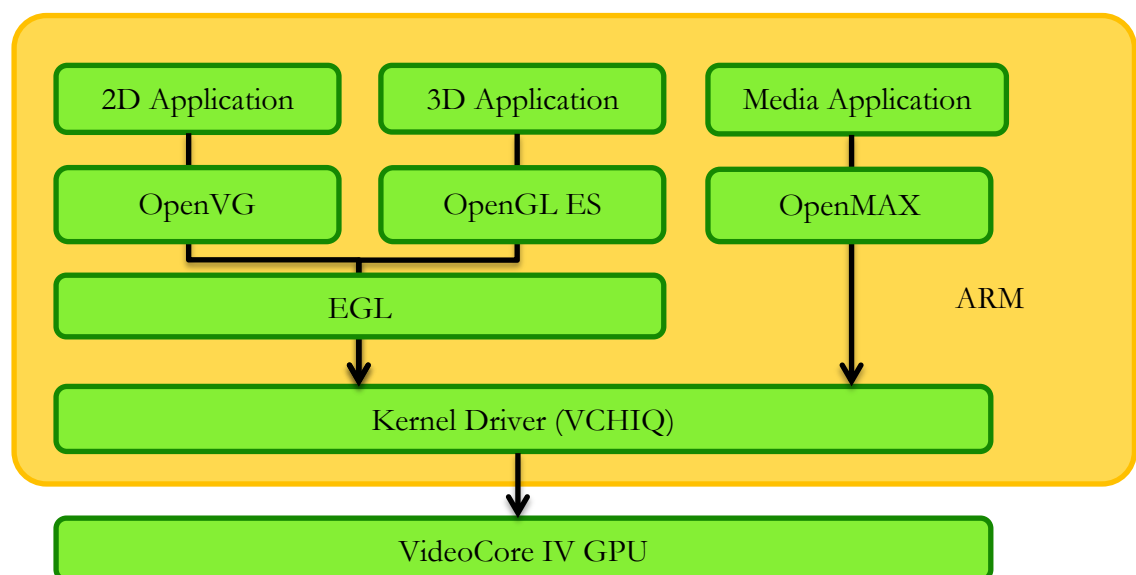
Kuva 2. Lohkokaavio Raspberry Pi -mallin B ominaisuuksista [15].

Kuvan 2 mukainen lähiverkon ohjausyksikkö LAN (Local Area Network) Controller ohjaa Raspberry Pi:n Ethernet- ja USB-portteja. Raspberry Pi voi muodostaa yhteyden internetiin kytkemällä Ethernet-kaapeli Raspberry Pi:stä käytössä olevaan laajakaistamodeemiin. Langaton verkkoyhteys voidaan muodostaa kytkemällä Raspberry Pi:n USB-porttiin USB Wi-Fi -adapteri. USB-porttiin on myös mahdollista kytkeä USB-hubi, jonka avulla laitteeseen saadaan lisää USB-portteja. [15.]

Raspberry Pi:ssä on viisi LEDiä (Light-Emitting Diode): OK versiossa 1 ja ACT versiossa 2, PWR, FDX, LNK, sekä 10M versiossa 1 ja 100 versiossa 2. OK- ja ACT-LEDit ilmoittavat vihreällä valolla SD-kortin aktiivisuudesta. PWR-LEDi palaa koko ajan punaisena laitteen saadessa virtaa. FDX-LEDi palaa vihreänä, kun Ethernet-yhteys toimii full-duplex-tilassa, eli tietoa voidaan lähettää ja vastaanottaa samanaikaisesti [17]. LNK-LEDi palaa vihreänä, kun Ethernet-kaapeli on kytketty Raspberry Pi -laitteeseen. 10M- ja 100-LEDit palavat oranssina, kun 100 MB:n LAN-yhteys on muodostettu. [18.]

### 3.3 Ohjelmiston arkkitehtuuri

Raspberry Pi:n Broadcom BCM2835 SoC -piirin ohjelmiston arkkitehtuurina on ARM. Ohjelmisto sisältää järjestelmän ytimen lisäksi grafiikan tuottamiseen vaikuttavia ohjelmointirajapintoja. Kuvassa 3 on esitetty Broadcom BCM2835 SoC -piirin ohjelmiston arkkitehtuuri. [19.]



Kuva 3. Broadcom BCM2835 SoC -piirin ohjelmiston arkkitehtuuri [20].

Multimediaprosessori VideoCore IV GPU (General Processing Unit) keskustelee järjestelmän ytimen Kernel Driverin (VCHIQ) kanssa. Kernelin päällä on Khronos Groupin EGL-rajapinta, jota käytetään OpenVG:llä ja OpenGL ES:llä toteutetun 2D- ja 3D-grafiikoiden luontiin sovellustasolla. [19.]

OpenMAX-ohjelmointirajapintaa käytetään median, kuten kuvan ja äänen tuottamiseen [19]. OpenVG on matalan tason 2D-vektorigrafiikan esittämiseen tarkoitettu järjestelmäriippumaton API (Application Programming Interface). Se on suunnattu ensisijaisesti käytettävän kannettavissa laitteissa, jotka vaativat korkealaatuisen vektorigrafiikan laitteistokiihdytyksen. Tämä API myös sallii laitteistokiihdytyksen tuottaakseen sulavampaa grafiikan esitystapaa vähäisellä virrankulutuksella. OpenGL ES on järjestelmäriippumaton API, jota käytetään 2D- ja 3D-grafiikoiden esittämiseen sulautetuilla laitteilla. [21.] [22.]

## 4 EGL-RAJAPINTA

EGL-rajapintaa käytetään luomaan grafiikkaa Khronos Groupin ohjelmointirajapintojen eli API:en, kuten OpenGL, OpenVG tai OpenGL ES, ja natiivin eli alustakohtaisen ikkunajärjestelmän välille. EGL käsittelee grafiikan kontekstin hallintaa, pinnan ja puskurin yhdistämistä, esittämisen synkronointia. Lisäksi se mahdollistaa suuren suorituskyvyn omaavan, kiihdytetyn 2D- ja 3D-grafiikoiden esittämisen käyttäen muita Khronos API-rajapintoja. [5.]

### EGL-rajapinnan toteutus

KajaPro Oy:n graafiseen etäkäyttöliittymäsovellukseen piti toteuttaa tuki EGL-rajapinnalle, jota käytetään sovelluksen grafiikan esittämiseen Raspberry Pi:llä. Rajapinnan koodaus toteutettiin C++-ohjelmointikielellä Windowsin Microsoft Visual Studio 2010 -ohjelman avulla. Raspberry Pi sisältää valmiiksi tämän rajapinnan vaatimat kirjastot ja header-tiedostot. Kirjastot löytyvät hakemistosta `/opt/vc/lib` ja header-tiedostot hakemistosta `/opt/vc/include`. Raspbian wheezy tukee OpenGL ES 2 API:a, jota käytettiin tässä työssä grafiikan luontiin EGL-rajapinnan avulla.

EGL-rajapinnan toteutuksessa ohjelmakoodissa ensimmäisenä pitää hakea yhteys näyttölaitteeseen järjestelmäpiirin `bcm_host_init()`-funktioikutsun avulla, jonka jälkeen alustetaan EGL:n yhteys. Tämän jälkeen haetaan sopivat kehyspuskurin asetukset ja luodaan konteksti sille asetetuilla parametreilla. Seuraavaksi luodaan EGL-ikkuna alustakohtaisilla parametreilla ja yhdistetään luotu konteksti kyseiseen ikkunaan. Viimeisenä käytetään grafiikan esittämiseen ja päivittämiseen tarvittavia funktioita. EGL-ikkuna tuhotaan destruktorilla sovelluksen sulkeutuessa.

## 5 SDL-RAJAPINTA

SDL on ohjelmistokehittäjän kannalta suunniteltu järjestelmäriippumaton kirjasto, jotta kehittäjä pääsee käsiksi laitteen äänen, näppäimistön, hiiren, joystickin ja grafiikan toiminnallisuuteen OpenGL:n ja Direct3D:n avulla. SDL-rajapintaa käytetään esimerkiksi videon toistossa, emulaattoreissa ja useissa Valven peleissä. Virallisesti SDL tukee Windows-, Linux-, Android-, Mac OS X - ja iOS-käyttöjärjestelmiä. [4.]

SDL on kirjoitettu C-ohjelmointikielellä ja toimii C++-ohjelmointikielen kanssa. SDL:stä on myös olemassa tukia muille ohjelmointikielille, kuten C# ja Python. SDL 1.2 ja tätä vanhemmat versiot ovat GNU LGPL:n (Lesser General Public License) alla. SDL 2.0 on zlib-lisenssin alla, joka sallii käyttämään SDL:ää ilmaiseksi missä tahansa ohjelmassa. [4.] [23.]

### 5.1 SDL-rajapinnan asennus Raspberrylle

KajaPro Oy:n ohjelmistokehitys tarjoaa valmiin toiminnallisuuden SDL-rajapinnalle. Tässä työssä SDL:llä suunniteltiin toteutettavan yrityksen sovelluksen hiiren ja näppäimistön toiminnallisuus Raspberry Pi:ssä. SDL luo oman ikkunan toiminnallisuudelle, jonka päälle sijoitetaan EGL:n luoma grafiikkaikkuna, joka käyttää OpenGL ES 2 -kirjaston funktioita 3D-grafiikan luontiin. SDL-1.2.15.tar.gz voidaan ladata osoitteesta <http://www.libsdl.org/download-1.2.php>.

Rajapinnan implementointi aloitettiin lataamalla SDL-1.2.15.tar.gz-paketti, joka siirrettiin Raspberry Pi:lle. Tämän jälkeen paketti purettiin, suoritettiin konfigurointi sekä asennustoimenpiteet Raspbian-ympäristössä. SDL-paketti puretaan komennolla **sudo tar xzvf SDL-1.2.15.tar.gz**. Konfiguroinnin avulla luodaan oma makefile-tiedosto, jota käytetään ohjelmakoodin kääntämiseen. Konfigurointi suoritetaan SDL-1.2.15-kansiossa komennolla **sudo ./configure**. SDL-lähdekoodi käännetään komennolla **sudo make**, jonka jälkeen suoritetaan SDL:n asennus komennolla **sudo make install**. [24, s. 133.]

Graafisen etäkäyttöliittymän SDL-rajapinnan toteutus Raspberry Pi:lle vaati myös SDL-image-paketin asennuksen. SDL-image-1.2.tar.gz-paketti voidaan ladata osoitteesta [https://www.libsdl.org/projects/SDL\\_image/release-1.2.html](https://www.libsdl.org/projects/SDL_image/release-1.2.html). Paketti puretaan, konfiguroidaan ja asennetaan samoilla toimenpiteillä kuin SDL-1.2.15-paketti Raspberry

Pi:lle. Komennot ovat siis **sudo tar xzvf SDL-image.1.2.tar.gz**, **sudo ./configure**, **sudo make** ja **sudo make install**.

## 5.2 Kirjastojen ja header-tiedostojen käyttöönotto ristikäntäjässä

SDL-kirjastot kopioidaan Raspberry Pi:ltä Windowsin puolelle SDL:n asennuksen jälkeen. Rajapinnan kirjastojen pitää sijaita sekä Raspberry Pi:llä että Windowsilla. SDL:n asennuksen jälkeen Raspberry Pi:n komentoriville tulostuu polku, johon SDL-kirjastot ovat asennettuina. Oletuksena tämä polku on /usr/local/lib. Kirjastot siirretään Windowsin puolella ristikäntäjän hakemistoon C:\SysGCC\Raspberry\arm-linux-gnueabi\lib. Tässä tapauksessa Raspberry Pi:ltä kopioitiin seuraavat SDL-kirjastot Windows 7 -käyttöjärjestelmällä toimivaan ristikäntäjän hakemistoon:

- libSDL.a
- libSDL.la
- libSDL.so
- libSDL\_image.a
- libSDL\_image.la
- libSDL\_image.so
- libSDL\_image-1.2.so.0
- libSDL\_image-1.2.so.0.8.4
- libSDL-1.2.so.0
- libSDL-1.2.so.0.11.4
- libSDLmain.a
- libSDLmain.la

Tiedostojen nimet alkavat aina etuliitteellä lib, jonka jälkeen seuraavana on kirjaston nimi, tiedostopääte ja mahdollinen versionumero. Staattisia kirjastoja ovat .a-päätteiset tiedostot ja jaettuja tai dynaamisia kirjastoja .so-päätteiset tiedostot. Libtool-työkalun käyttämät kirjastot ovat .la-päätteiset kirjastot. Tiedostotyyppin voi tarkistaa suorittamalla Raspberryn komentoriviltä esimerkiksi komento **file libSDL.so**. Osa näistä kirjastoista on symbolisia linkkejä eli eräänlaisia pikakuvakkeita varsinaisiin kirjastoihin. [24, s. 567]. [24, s. 134.]

## 6 OHJELMAKOODIN KÄÄNTÄMINEN

Tässä luvussa kerrotaan yleistä tietoa GCC-kääntäjästä, GDB-työkalusta (GNU Project Debugger), ristikääntäjästä sekä siitä, kuinka ristikääntäjällä käännetään Raspberry Pi binääri Windows 7 -ympäristössä. Näiden lisäksi kerrotaan C- ja C++-ohjelmointikielillä toteutetun ohjelmakoodin kääntämisestä objektitiedostoihin, sekä kuinka kääntäjä hyödyntää kirjastoja linkkerin avulla. Lopuksi esitetään näiden tietojen pohjalta, kuinka KajaPro Oy:n graafisesta etäkäyttöliittymästä on toteutettu suoritettava tiedosto Windows 7 -käyttöjärjestelmällä.

### 6.1 GCC-kääntäjä

GNU Compiler Collection on Richard Stallmanin vuonna 1987 kehittämä ilmainen kääntäjien kokoelma GNU-projektia varten. Nykyään GCC-kääntäjää on kehittämässä monia eri ohjelmoijia ympäri maailman ja se on käännetty usealle eri suorittimelle ja käyttöjärjestelmälle. Ilmaista GCC-kääntäjää käytetään lähinnä Linux-käyttöjärjestelmässä, mutta myös UNIXin kaltaisissa käyttöjärjestelmissä, kuten Mac OS X:ssä, NetBSD:ssä, FreeBSD:ssä (Berkeley Software Distribution) ja Solariksessa. Free Software Foundation jakaa GCC:n GNU GPL:n (General Public License) alle. [25.] [26, s. 24 – 26.] [27.]

GCC:n versio 4.6.3:n standardi tukee ohjelmointikieliä C, C++, Objective-C, Objective-C++, Ada, Fortran sekä Java [28]. GCC-kääntäjä sisältää viisi eri vaihetta, jotka kääntäjä käy läpi kääntäessään ohjelmakoodin. Näihin vaiheisiin kuuluvat esikäsittely, käänнос, optimointi, assemblerin - ja linkkerin toiminnot. [26, s. 29]

#### 6.1.1 GDB-työkalu

GDB on GNU-projektin ilmainen työkalu ohjelmakoodin testaukseen, jolla voidaan nähdä, mitä sovellus tekee, kun se suoritetaan. Testattava sovellus voi olla kirjoitettuna Ada-, Assembly-, C-, C++-, Fortran-, Java-, Modula-2-, Objective-C- tai Pascal-ohjelmointikielillä [29]. Käyttäjä voi hyödyntää työkalua havaitsemaan ohjelmakoodissa esiintyvät virheet. GDB:tä voidaan käyttää määrittämään sovellukseen vaikuttavia tekijöitä sen suorituksen alussa, pysähtymään tiettyyn kohtaan koodissa, tutkimaan sovelluksen pysähtymisen syytä,

sekä muuttamaan sovellusta niin, että käyttäjä voi kokeilla virheenkorjauksen vaikutusta ja jatkaa eteenpäin seuraavaan virheeseen. [30.] [31.]

Ohjelmakoodin kääntämisen yhteydessä on kääntäjälle määriteltävä lippu `-ggdb`, joka sallii sovelluksen testaamisen GDB:n avulla. Taulukossa 4 on esitetty muutamia GDB:lle annettavista komennoista.

Taulukko 4. GDB-komentoja [32].

Komento	Tarkoitus
<code>gdb HelloWorld.exe</code>	Suorittaa sovelluksen GDB:llä.
<code>break luokanNimi::funktionNimi</code> <code>b luokanNimi::funktionNimi</code>	Asettaa pysäytyspisteen (breakpoint) määrättyyn funktioon ja keskeyttää ohjelman tässä funktiossa.
<code>break tiedostonNimi.cpp:rivinumero</code> <code>b tiedostonNimi.cpp:rivinumero</code>	Asettaa pysäytyspisteen määrättylle riville ja keskeyttää ohjelman kyseisessä kohdassa.
<code>run</code> <code>r</code>	Käynnistää sovelluksen.
<code>continue</code> <code>c</code>	Jatkaa sovelluksen suoritusta ja pysähtyy pysäytyspisteeseen tai katselukohtaan.
<code>delete</code> <code>d</code>	Tuhoaa kaikki pysäytyspisteet, katselukohdat (watchpoint) ja sieppauspisteet (catchpoint).
<code>kill</code>	Lopettaa ohjelman (prosessin) suorituksen.
<code>quit</code> <code>q</code>	Sulkee GDB testauksen.

Sovellus voidaan testata suorittamalla se Raspbian wheezy:n komentoriviltä taulukossa 4 esiintyvän komennon avulla. Pysäytyspisteitä voidaan asettaa ohjelmakoodiin ennen kuin sovellus on käynnistetty `run`-komennolla tai aina kun ohjelma on keskeytetty. [32.]

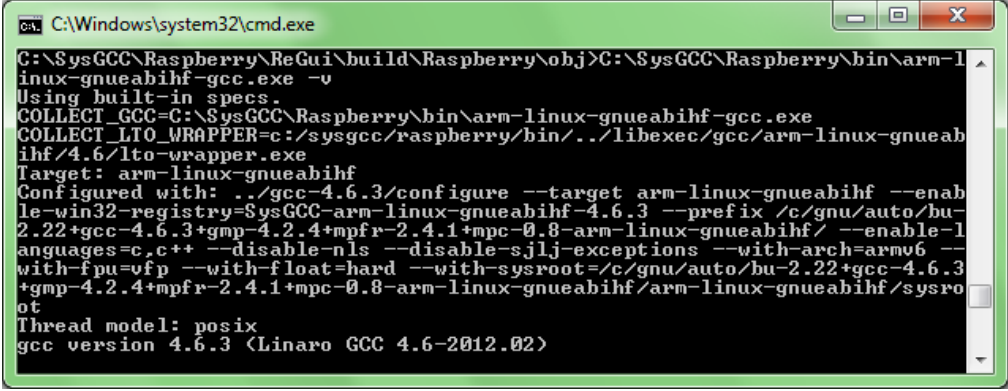
### 6.1.2 Ristikääntäjä

Ristikääntäjällä tarkoitetaan ohjelmakoodin kääntäjää, jonka avulla käännettävä koodi toimii useassa eri laitteistoarkkitehtuurissa. Tässä insinööriyössä ristikääntäjää on käytetty



kääntämään koodi Windows 7 -ympäristössä suoritettavaksi tiedostoksi, joka toimii Raspbian wheezy -käyttöjärjestelmällä Raspberry Pi:llä arkkitehtuurilla ARMv6. Ristikääntäjä voi tarvita tiettyjen kirjastojen lisäyksen riippuen sovelluksen toiminnallisuuden ja grafiikan toteutustavasta.

Ristikääntäjälle voidaan asettaa lippu `-v`, joka tulostaa komentoriville konfigurointitietoja, joilla kääntäjä on rakennettu. GCC käyttää näitä asetuksia kääntäessään ohjelmakoodia. Kuvassa 4 ristikääntäjä suoritetaan kyseisellä lipulla Windowsin komentoriviltä. [33.]



```

C:\Windows\system32\cmd.exe
C:\SysGCC\Raspberry\ReGui\build\Raspberry\obj>C:\SysGCC\Raspberry\bin\arm-l
inux-gnueabi\gcc.exe -v
Using built-in specs.
COLLECT_GCC=C:\SysGCC\Raspberry\bin\arm-linux-gnueabi\gcc.exe
COLLECT_LTO_WRAPPER=c:/sysgcc/raspberry/bin/./libexec/gcc/arm-linux-gnueab
ihf/4.6/lto-wrapper.exe
Target: arm-linux-gnueabi
Configured with: ../gcc-4.6.3/configure --target arm-linux-gnueabi --enab
le-win32-registry=SysGCC-arm-linux-gnueabi-4.6.3 --prefix /c/gnu/auto/bu-
2.22+gcc-4.6.3+gmp-4.2.4+mpfr-2.4.1+mpc-0.8-arm-linux-gnueabi/ --enable-l
anguages=c,c++ --disable-nls --disable-sjlj-exceptions --with-arch=armv6 --
with-fpu=vfp --with-float=hard --with-sysroot=/c/gnu/auto/bu-2.22+gcc-4.6.3
+gmp-4.2.4+mpfr-2.4.1+mpc-0.8-arm-linux-gnueabi/arm-linux-gnueabi/sysro
ot
Thread model: posix
gcc version 4.6.3 (Linaro GCC 4.6-2012.02)

```

Kuva 4. Ristikääntäjä suoritettuna Windowsin komentoriviltä lipulla `-v`.

`COLLECT_GCC` on polku `arm-linux-gnueabi-gcc`-kääntäjälle ja `COLLECT_LTO_WRAPPER` polku wrapperille, joka muuntaa datan yhteensopivaan muotoon. [33.]

## 6.2 C- ja C++-koodien kääntäminen

Tässä insinööriyössä käytettiin GCC- ja G++-kääntäjiä `arm-linux-gnueabi-gcc` ja `arm-linux-gnueabi-g++` kääntämään KajaPro Oy:n graafisen etäkäyttöliittymän C- ja C++-koodit. Esikäsittelijä `arm-linux-gnueabi-cpp` aloittaa kääntämisen ensimmäisen vaiheen, jossa se käsittelee #-alkuiset komennot eli makrot, kuten `#define`, `#include`, `#ifdef`, `#ifndef`, `#endif` ja `#macro` [34]. Käännösvaiheessa kääntäjä kääntää ohjelmakoodin assembly-koodiksi, jonka jälkeen kääntäjä tekee assembly-koodiin suorituskykyä parantavia muutoksia eli optimoi koodin. Assembler `arm-linux-gnueabi-as` kääntää assembly-koodin objektitiedostoiksi. Linkkeri kasaa objektitiedostot ja lisäkirjastot ajettavaksi tiedostoksi. [26, s.29]

### 6.2.1 Objektitiedostot, linkkeri ja tarvittavat kirjastot

Assemblerin avulla saadut objektitiedot sisältävät konekieltä, jolla tarkoitetaan käännettä ohjelmakoodia, joka ei ole suoraan suoritettavassa muodossa yksittäisissä tapauksissa, mikäli objektitiedostoja on useita yhden ohjelman suorittamiseen. GCC-kääntäjän linkkeri (ld) käyttää objektitiedostoja ja kirjastoja luodakseen suoritettavan tiedoston. Objektitiedostot ja kirjastot linkitetään keskenään linkkerin avulla. Tässä työssä toteutettiin SDL- ja EGL-kirjastot, joiden metodeja sovellus tarvitsee toimiakseen Raspberry Pi:llä. [35.]

### 6.2.2 Suoritettavan tiedoston luominen

Suoritettavan tiedoston luo lopulta kääntäjän linkkeri. Työssä käytettiin kolmea eri bat-tiedostoa, `compileRpi.bat`, `linkRpi.bat` ja `buildAllRpi.bat`, jotka voidaan suorittaa suoraan Windows 7:n komentoriviltä. Tiedoston `compileRpi.bat` avulla suoritetaan C- ja C++-koodien kääntäminen objektitiedostoiksi GCC:n avulla. Tiedostoa `linkRpi.bat` käytetään linkkerin toimintoihin. Tiedosto `buildAllRpi.bat` suorittaa ensin `compileRpi.bat`-tiedoston ja tämän jälkeen `linkRpi.bat`-tiedoston.

Kääntäjää varten toteutettu `compileRpi.bat`-tiedosto kääntää kaiken C- ja C++-ohjelmointikielillä kirjoitetun koodin annetuilla määrittelyillä ja tarvittavilla lippuparametreilla. Komentoriviltä suoritettaessa käännoystä pitää kyseiseen bat-tiedostoon määrittellä polku, johon GCC on asennettu. Linkkeriä varten toteutettu `linkRpi.bat`-tiedosto linkittää GCC:n kääntämät objektitiedostot yhdessä kirjastojen kanssa yhdeksi suoritettavaksi tiedostoksi. Myös linkkerille pitää antaa tietyt määrittelyt, polku kirjastojen sijaintiin sekä niihin tiedostoihin, jotka ovat tarpeellisia ohjelman toiminnan kannalta.

## 7 KEHITYSYMPÄRISTÖN KOKOONPANO

Kehitysympäristön kokoonpanoon tarvitsee PC:lle laitteiston puolelta Windows 7 -käyttöjärjestelmän ja Raspberry Pi -sulautetun laitteen Raspbian wheezy -käyttöjärjestelmällä. Raspbian wheezy -käyttöjärjestelmä asennetaan SD-kortille, joka kytketään Raspberry Pi:lle. Tässä insinööriyössä käytettiin Raspberry Pi -laitteen malli B:n versiota 1 ja Microsoft Visual Studio 2010 -ohjelmaa Windows 7:llä C- ja C++-ohjelmointikielien ohjelmointityökaluna. Raspberry Pi:n käyttö vaatii vähintään tietokoneen näppäimistön ja näyttölaitteen käyttöä, mutta on suositeltavaa myös kytkeä hiiri Raspberry Pi:n USB-porttiin, kuten kuvassa 5 on esitetty.

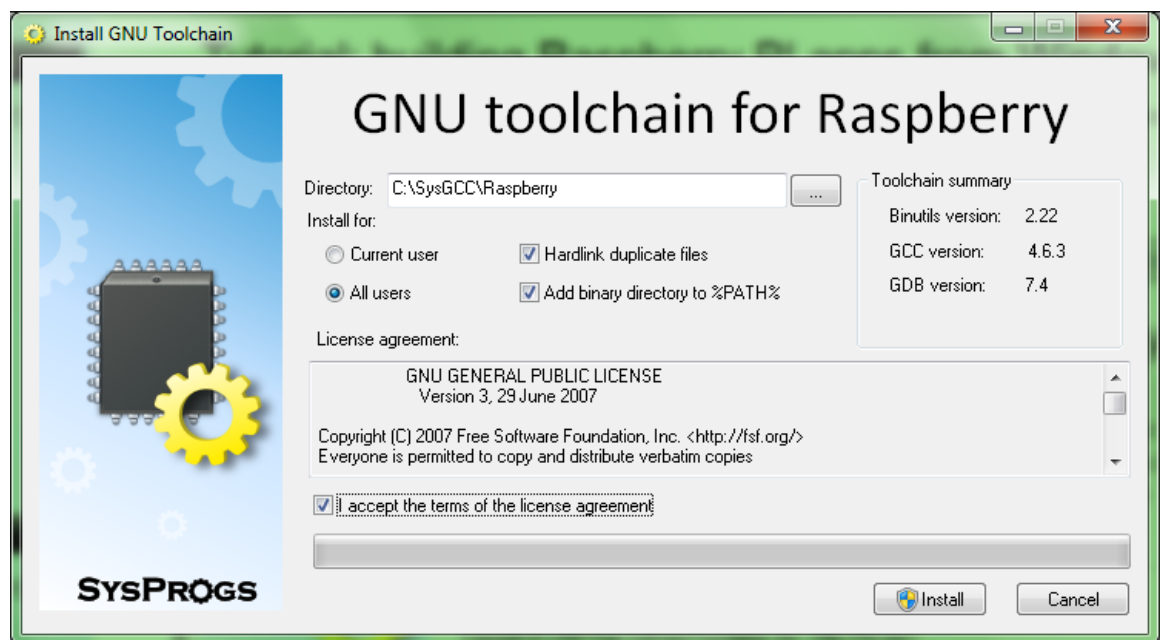


Kuva 5. Kehitysympäristön laitteisto.

## 7.1 Ristikääntäjän asennus GNU-työkalun avulla

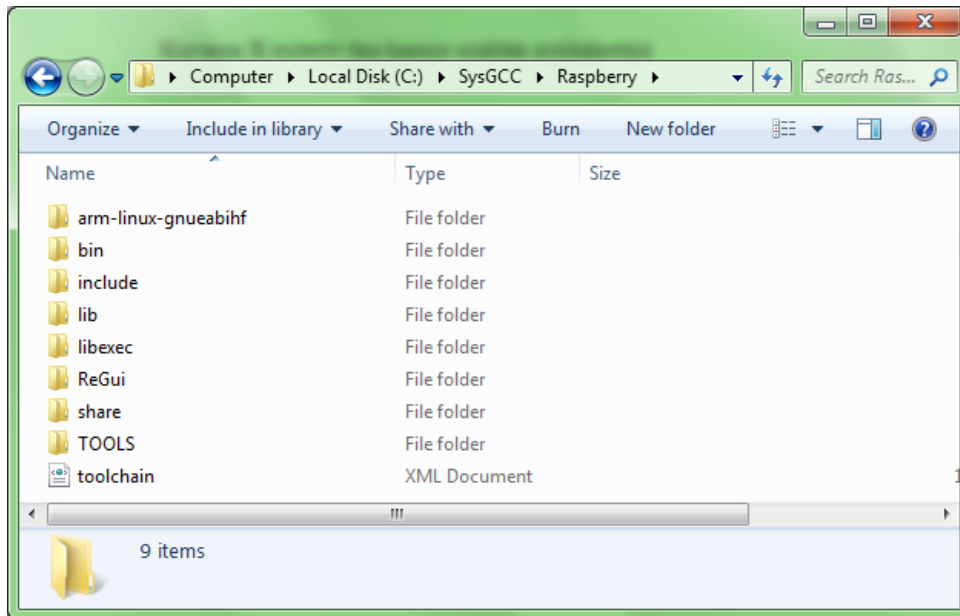
GNU on ilmainen UNIXin kaltainen käyttöjärjestelmä. GNU-projekti alkoi vuonna 1984 tuottaa GNU-järjestelmää. Se on sovelluskokoelma, joka koostuu ohjelmista, kirjastoista ja kehitystyökaluista. Tämän ohjelman avulla myös varataan laitteiston resursseja ja keskustellaan laitteiston ytimen eli kernelin kanssa. [36.]

Ristikääntäjä saadaan toteutettua kuvan 6 mukaisen GNU-työkalun avulla, joka sisältää valmiiksi tarvittavat kääntäjät C- ja C++-ohjelmointikielille, muutamia kirjastoja sekä header-tiedostoja Raspberry Pi:ltä. Työkalu on suoritettava tiedosto (raspberry-gcc4.6.3.exe), joka asentaa peruselementit tarvittavan kehitysympäristön kokoonpanoa varten. [37.]



Kuva 6. GNU-työkalun asennusikkuna.

Ristikääntäjä asennetaan oletuksena polkuun C:\SysGCC\Raspberry. KajaPro Oy:n graafinen etäkäyttöliittymä ReGui siirretään Raspberry-hakemiston sisään, kuten kuvassa 7 on esitetty.



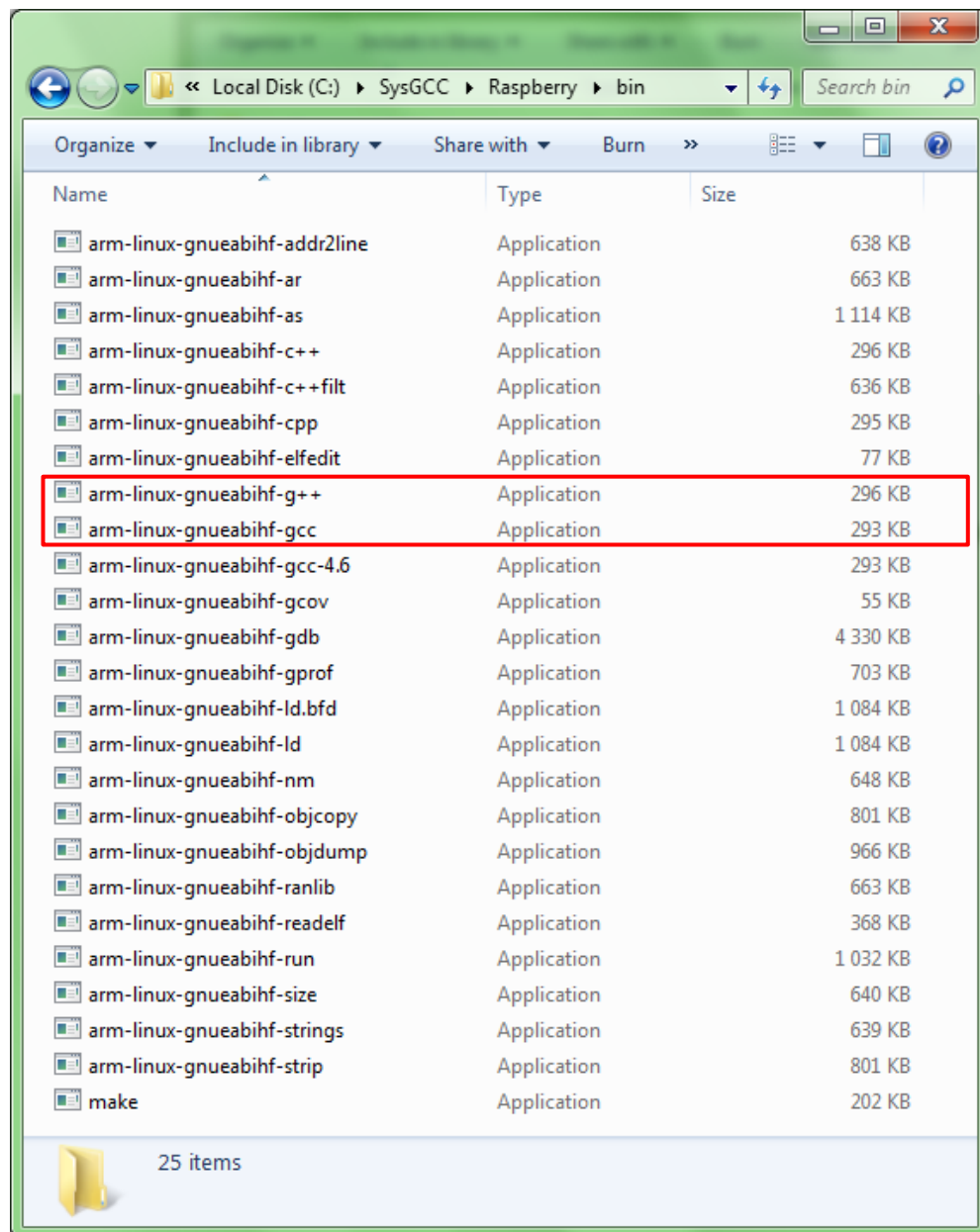
Kuva 7. Ristikääntäjän hakemistorakenne.

Kuvassa 7 esiintyvä `arm-linux-gnueabi`-kansio sisältää `bin`-, `include`-, `lib`- ja `sysroot`-kansiot. Raspberry Pi:n hakemistosta `/opt/vc/include` kopioidaan `EGL`-, `GLES2`-, `interface`-, `KHR`- ja `vcinclude`-kansiot Windowsin hakemistoon `C:\SysGCC\Raspberry\arm-linux-gnueabi\include`. Hakemistosta `/opt/vc/lib` kopioidaan tarvittavat jaetut kirjastot Windowsin hakemistoon `C:\SysGCC\Raspberry\arm-linux-gnueabi\lib`. Nämä kirjastot ovat:

- `libbcm_host.so`
- `libEGL.so`
- `libGLESv2.so`
- `libvchiq_arm.so`
- `libvcos.so`

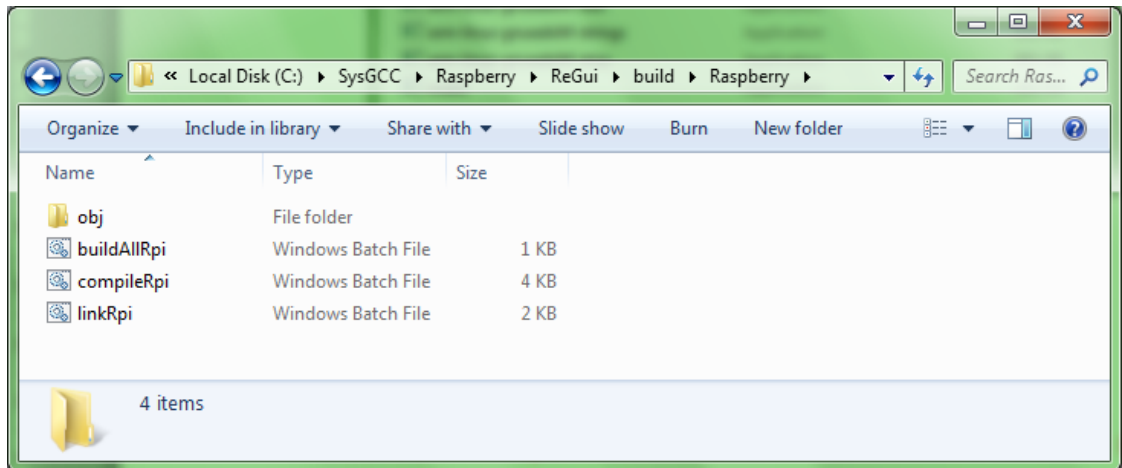
Kirjasto `libbcm_host.so` on Raspberry Pi:n Broadcom-järjestelmäpiirin käyttöliittymän kirjasto, joka sisältää VideoCore API:t, joita tarvitaan näyttölaitteiden alustukseen [38]. Khronos Groupin kirjasto `libEGL.so` sisältää EGL-rajapintaan liittyviä metodeja. Grafiikkakirjasto `libGLESv2.so` sisältää OpenGL ES 2:n menetit. Kirjastot `libvchiq_arm.so` ja `libvcos.so` sisältävät Raspberryn kernel-tason ajureihin liittyviä metodeja.

Kuvassa 8 esitetty `bin`-kansio sisältää työssä käytettävät kääntäjät `arm-linux-gnueabi-gcc` ja `arm-linux-gnueabi-g++`. C-ohjelmointikielillä kirjoitetut koodit käännetään `arm-linux-gnueabi-gcc`-kääntäjällä ja C++-koodit käännetään `arm-linux-gnueabi-g++`-kääntäjällä.



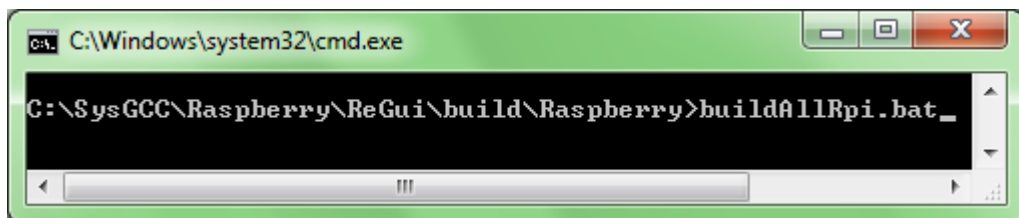
Kuva 8. Bin-kansion sisältö.

Kuvan 9 mukaisen etäkäyttöliittymähakemiston sisältämä Raspberry-kansio sisältää ristikäännökseen tarkoitetut buildAllRpi.bat-, compileRpi.bat- ja linkRpi.bat-tiedostot. Ristikääntäjä kääntää C- ja C++-ohjelmointikielillä toteutetut koodit objektitiedostoiksi obj-kansioon compileRpi.bat-tiedoston avulla. Kyseiseen kansioon luodaan myös lopullinen suoritettava tiedosto ReGui.exe linkRpi.bat-tiedoston avulla.



Kuva 9. Etäkäyttöliittymän Raspberry-kansion sisältö.

Raspberry-hakemistosta löytyvä buildAllRpi.bat-tiedosto suoritetaan Windowsin komentoriviltä kuvan 10 mukaisesti. Kyseinen tiedosto käynnistää ensin compileRpi.bat-tiedoston, joka aluksi poistaa kaikki tiedostot obj-kansiosta ja sen jälkeen suorittaa KajaPro Oy:n etäkäyttöliittymän ohjelmakoodin kääntämisen objektitiedostoiksi. Tämän jälkeen buildAllRpi.bat-tiedosto suorittaa linkRpi.bat-tiedoston, joka linkittää ristikäntäjälle kaikki tarvittavat kirjastot ja yhdistää kaikki objektitiedostot yhdeksi suoritettavaksi tiedostoksi.



Kuva 10. Etäkäyttöliittymän kääntäminen buildAllRpi.bat-tiedoston avulla.

Ristikääntäjälle pitää asettaa tietyt liput tai määrittelyt, jotta halutut koodit ovat voimassa käännöksen aikana, mikäli koodit käyttävät esimerkiksi ehtoa `#ifdef RASPBERRY_BUILD`. Tämän ehdon sisällä oleva koodi on voimassa, kun kääntäjälle on määritely `-DRASPBERRY_BUILD`. Tässä työssä ristikäntäjälle asetettiin `compileRpi.bat`- ja `linkRpi.bat`-tiedostoissa määrittelyt `-O2`, `-march=armv6`, `-mfloat-abi=hard`, `-ggdb`, sekä `-fpic`, joiden tarkoitukset on esitetty taulukossa 5. Näiden lisäksi kääntäjälle pitää lisätä polku komennolla `-I`, josta työssä käytettävät header-tiedostot löytyvät.

Taulukko 5. Ristikääntäjän määrietykset [39].

Määritys	Selitys
-O2	Optimointiasetus, joka nopeuttaa koodin kääntämistä ja parantaa tuotetun koodin suorituskykyä.
-march=armv6	Määrittää kohdelaitteen arkkitehtuuriksi ARMv6.
-mfloat-abi=hard	Määrittää laitteen liukuluku-ABI:ksi hard-float ABI:n.
-ggdb	Sallii GDB-työkalun käytön.
-fpic	Tuottaa paikasta riippumatonta koodia, jota voidaan käyttää jaetussa kirjastossa.

Ristikääntäjän linkkerille pitää määritellä lipulla `-L` polku, josta työssä käytettävät kirjastot löytyvät, sekä kirjastojen nimet ilman etuliitettä `lib` ja tiedostopäätteitä. Komento `-l` linkittää kirjaston. Näin ollen suoritettava tiedosto luodaan esimerkiksi `Testi.bat`-nimisellä tiedostolla, joka sisältää seuraavan koodin:

```
set PATHG++=C:\SysGCC\Raspberry\bin\arm-linux-gnueabihf-g++.exe
set LDLIBS=-LC:\SysGCC\Raspberry\arm-linux-gnueabihf\lib
set DEFINES=-O2 -DRASPBERRY_BUILD -march=armv6 -mfloat-abi=hard -ggdb -fpic
-I..\..\..\arm-linux-gnueabihf\include\SDL

cd obj
del *.exe

%PATHG++% -o Test.exe MainApp.o EGLApp.o SDLApp.o %DEFINES%
%LDLIBS% -ISDL -ISDL_image -IGLESv2 -IEGL -lvcos -lbcm_host -lvchiq_arm
cd ..
```

Tiedostossa komento `set PATHG++` käynnistää ristikääntäjän, `set LDLIBS` määrittelee polun kirjastojen sijaintiin ja `set DEFINES` määrittelee kääntäjälle lippuparametrit sekä header-tiedostojen sijainnin.

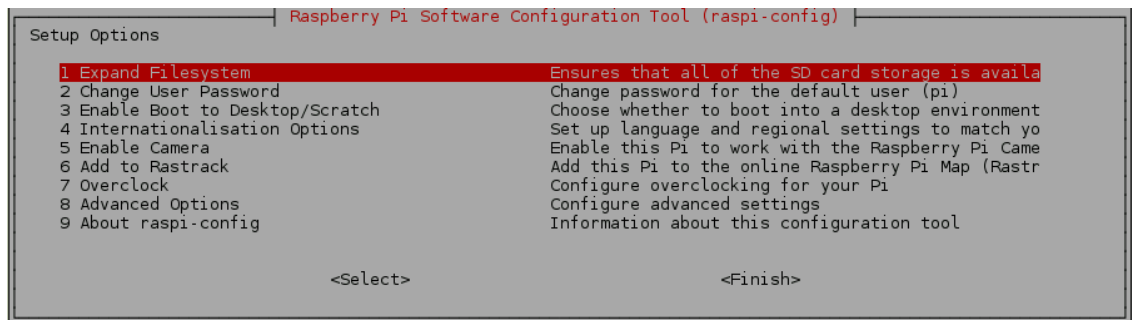
## 7.2 SSH-yhteyden luominen

SSH (Secure Shell), joka tunnetaan myös nimellä Secure Socket Shell, on UNIX-käyttöjärjestelmään perustuva komentorajapinta ja protokolla suojatun yhteyden muodostamiseen etänä toimivaan tietokoneeseen. Tietoverkon järjestelmänhallitsijat käyttävät SSH:ta verkon ja muiden servereiden kontrollointiin. SSH on `slogin`-, `ssh`- ja `scp`-apuohjelmien yhdistelmä, jotka ovat suojattuja versioita aiemmista UNIXin `rlogin`-, `rsh`- sekä



rcp-apuohjelmista. SSH-komennot ovat salattuja ja suojattuja useilla eri tavoilla. Molempien, sekä asiakasohjelma- että serveri-päätteiden, yhteydet on todennettu käyttäen digitaalista sertifikaattia ja salasanat on turvattu salauksilla. SSH käyttää RSA- julkisen avaimen salaustekniikkaa SSH-yhteydessä ja todennuksessa. [40.]

SSH-yhteyden muodostamiseksi täytyy Raspberry Pi:llä sallia SSH-yhteyden muodostus. Tämä tapahtuu muokkaamalla Raspberry Pi:n asetuksia. Raspberry Pi:n asetuksiin pääsee kirjoittamalla komentoriville ja suorittamalla komento **sudo raspi-config**. Komennon suorittamisen jälkeen avautuu kuvan 11 mukainen asetussivetti. Käyttäjä voi sallia SSH-yhteyden valitsemalla asetuskohdan 8 Advanced Options, tästä edelleen kohta A4 SSH ja <Enable>. Tämän jälkeen Raspberry Pi:n komentoriville päästään menemällä takaisin kuvan 11 mukaiseen näkymään ja valitsemalla <Finish>.



Kuva 11. Raspberry Pi:n asetukset-ikkuna.

Tässä työssä Windows 7- ja Raspbian wheezy -käyttöjärjestelmien välille muodostettiin SSH-yhteys suoraan kytketyn Ethernet-kaapelin ja reitittävän modeemin avulla.

Kytken jälkeen on syytä tarkastaa, saako Raspberry Pi IP-osoitetta (Internet Protocol). Tarkistus suoritetaan Raspberry Pi:n komentoriviltä komennolla **sudo ifconfig**. Komentoriville tulostuu kuvan 12 mukaiset tiedot, joista eth0-kohdan inet addr on Raspberryn saama IP-osoite. Windowsilla kyseinen toimenpide suoritetaan komennolla **ipconfig**.

```

pi@raspberrypi ~ $ sudo ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:e9:99:08
          inet addr:192.168.1.37  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:241 errors:0 dropped:0 overruns:0 frame:0
          TX packets:262 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:58964 (57.5 KiB)  TX bytes:30569 (29.8 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:12 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:600 (600.0 B)  TX bytes:600 (600.0 B)

```

Kuva 12. Raspberry Pi:llä suoritettu **sudo ifconfig** -komento.

Raspberry Pi:lle voidaan asettaa staattinen IP-osoite, jonka ansiosta IP-osoitetta ei tarvitse aina tarkistaa **sudo ifconfig** -komennolla, koska IP-osoite ei muutu laitteen käynnistyksen yhteydessä. Staattinen IP-osoite kirjoitetaan Raspberryn interfaces-tiedostoon. Raspberryltä täytyy ensin tarkistaa kernelin IP-reititys, joka suoritetaan komentoriviltä komennolla **netstat -r**, jonka jälkeen komentoriville tulostui kuvan 13 mukaiset tiedot.

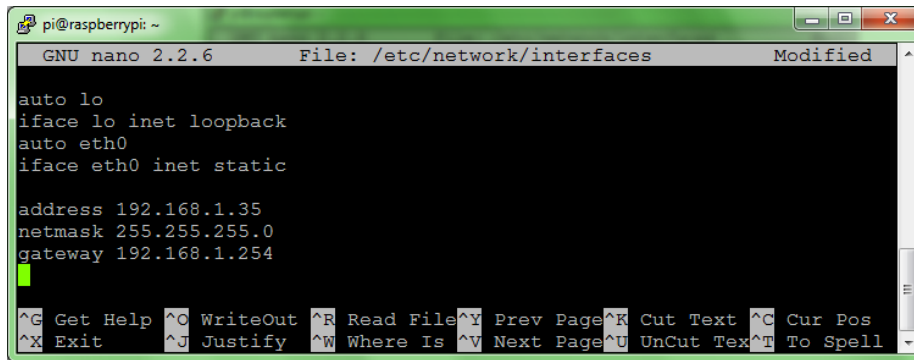
```

pi@raspberrypi ~ $ netstat -r
Kernel IP routing table
Destination        Gateway            Genmask           Flags   MSS Window  irtt Iface
default            192.168.1.1       0.0.0.0           UG      0 0        0 eth0
192.168.1.0        *                  255.255.255.0     U       0 0        0 eth0

```

Kuva 13. Raspberryn kernelin IP-reititys.

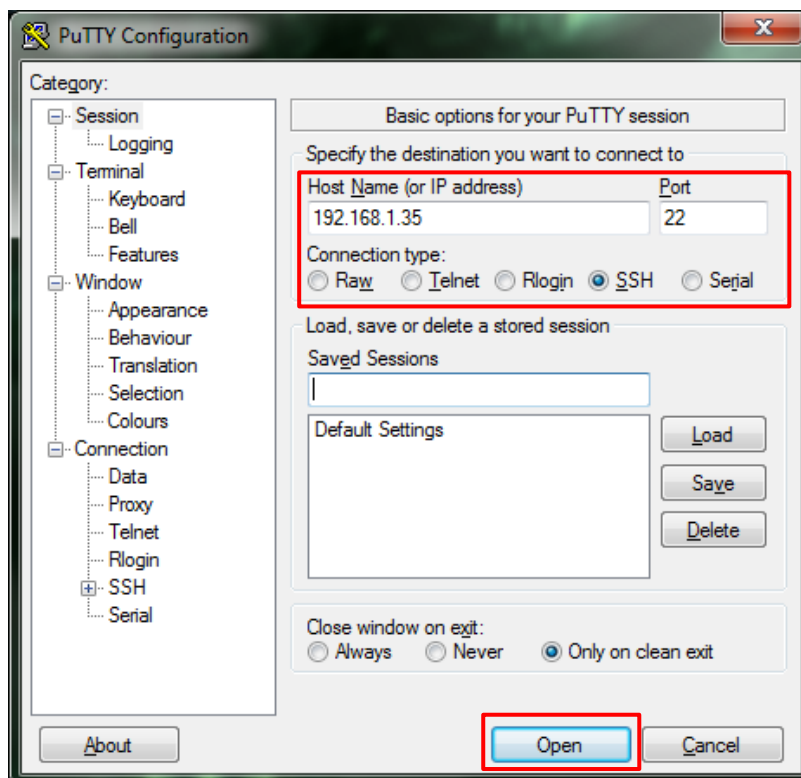
Staattisen IP-osoitteen asettamisen kannalta kuvasta 13 tärkeät tiedot ovat määränpään (Destination) ja yhdyskäytävän (Gateway) IP-osoitteet, joita tarvitaan interfaces-tiedoston muokkaamiseen. Tiedosto muokataan komentoriviltä komennolla **sudo nano /etc/network/interfaces**. Komentoriville tulostuvan interfaces-tiedoston sisältö muokataan kuvan 13 perusteella kuvan 14 mukaiseksi.



Kuva 14. Staattisen IP-osoitteen asettaminen Raspberry Pi:lle.

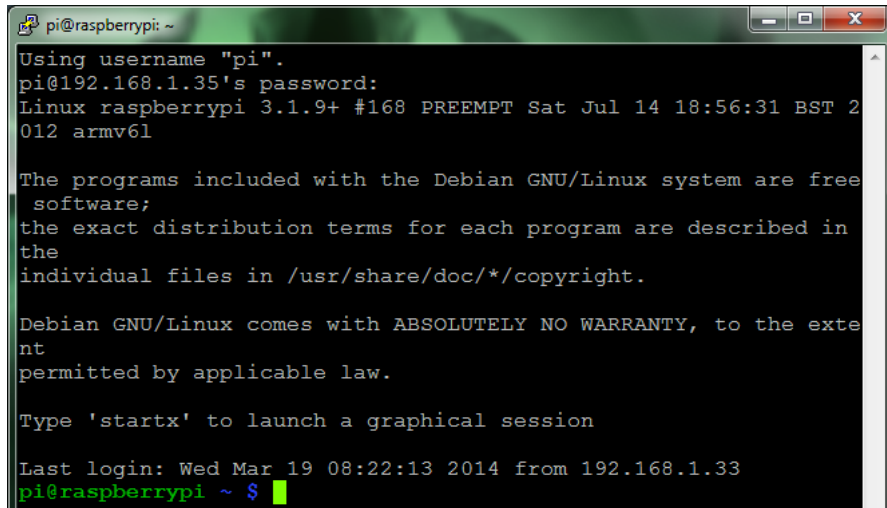
Tiedoston päällekirjoitus suoritetaan näppäinyhdistelmällä Ctrl+o, ja muutokset hyväksytään painamalla Enter. Tämän jälkeen tiedostosta poistutaan näppäinyhdistelmällä Ctrl+x. Lopuksi Raspberry Pi käynnistetään uudelleen ja tarkistetaan laitteen IP-osoite, jotta se vastaa asetettua staattista IP-osoitetta.

SSH-yhteyden muodostamiseen Windowsin puolelta käytetään ilmaista PuTTY-testiohjelmaa, jota voidaan käyttää esimerkiksi myös sarjaportti- ja telnet-yhteyksien tiedonsiirrossa. Lataamalla PuTTY Windows 7 -käyttöjärjestelmälle ja käynnistämällä sovellus avautuu kuvan 15 mukainen näkymä.



Kuva 15. PuTTY-testiohjelma.

Raspberry Pi:n saama tai sille asetettu staattinen IP-osoite kirjoitetaan Host Name (or IP address) -kenttään ja Port-kenttään SSH-yhteydelle määritelty portti numero 22. Yhteyden tyyppi (Connection type) valitaan SSH, ja lopuksi avataan yhteys painamalla Open-painiketta. PuTTY aukaisee yhteydenmuodostusikkunan ja Raspberry pyytää varmenteena omaa salasanaan. SSH-yhteys on muodostettu Windowsin ja Raspberry Pi:n välille kuvan 16 mukaisesti Raspberryn saatua oikean salasanan.



```

pi@raspberrypi: ~
Using username "pi".
pi@192.168.1.35's password:
Linux raspberrypi 3.1.9+ #168 PREEMPT Sat Jul 14 18:56:31 BST 2
012 armv6l

The programs included with the Debian GNU/Linux system are free
software;
the exact distribution terms for each program are described in
the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the exte
nt
permitted by applicable law.

Type 'startx' to launch a graphical session

Last login: Wed Mar 19 08:22:13 2014 from 192.168.1.33
pi@raspberrypi ~ $

```

Kuva 16. SSH-yhteys muodostettu Windowsin ja Raspberry Pi:n välille.

Raspberry Pi:llä oleva graafinen etäkäyttöliittymä voidaan käynnistää etänä Windowsilta SSH-yhteyden kautta.

SSH-yhteyden muodostuksessa voi ilmetä ongelmia, vaikka SSH-yhteys on aikaisemmin saatu toimimaan. On mahdollista, että serveri ei pysty lähettämään isännän avainta, koska tiedosto voi puuttua tai se voi olla korruptoitunut. Tämä voidaan korjata suorittamalla Raspberry Pi:n komentorivillä komennot **sudo rm /etc/ssh/ssh\_host\_\*** ja **sudo dpkg-reconfigure openssh-server**. Ensimmäinen komento poistaa kaikki aikaisemmat isännän avaimet, ja toinen komento luo uudelleen SSH-yhteyden isännän avaimet. Tämän jälkeen Raspberry Pi käynnistetään uudelleen komennolla **sudo reboot**. [41.]

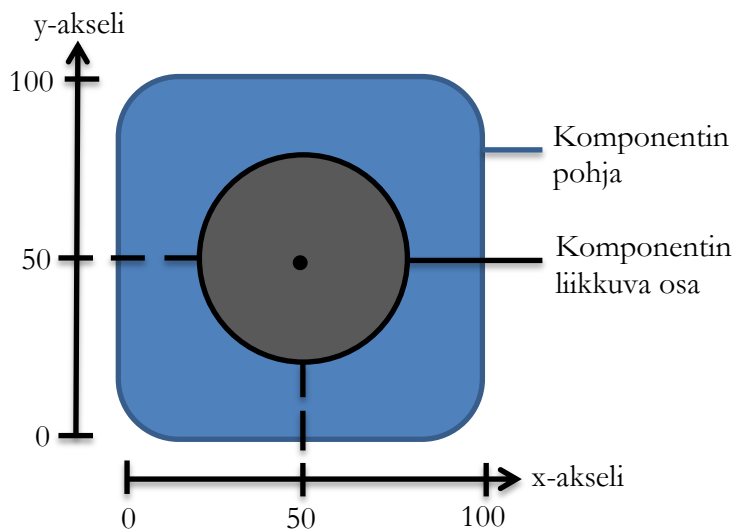
## 8 JOYSTICK-KOMPONENTTI

Tässä luvussa esitetään KajaPro Oy:n graafiseen etäkäyttöliittymään suunniteltu, toteutettu ja testattu joystick-komponentti. Komponentin toiminnallisuuden toteutuksen lisäksi toteutettiin myös komponentin grafiikka käyttäen Adobe Photoshop Elements 9 -ohjelmaa. Suunnitteluvaiheessa määriteltiin, mitkä ovat joystick-komponentin vaatimukset sen toiminnan kannalta, sekä mitä komponentilla pitää pystyä tekemään.

### 8.1 Komponentin suunnittelu

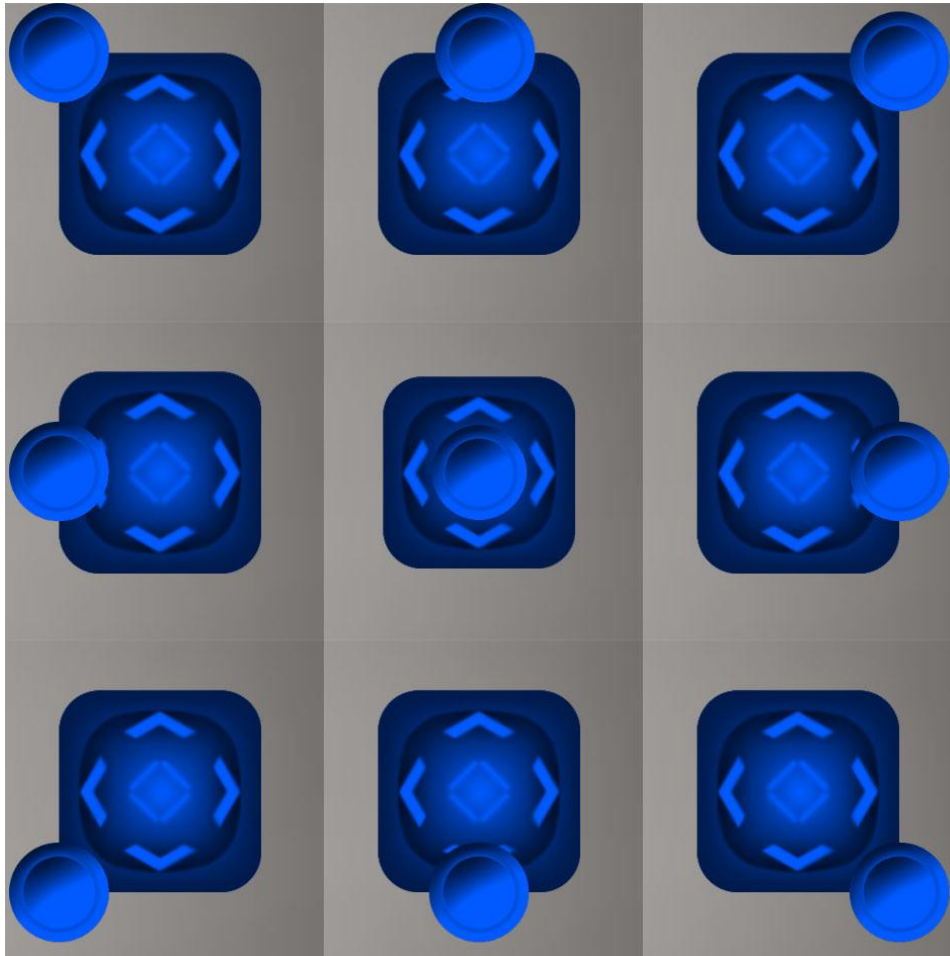
Joystick-komponentti suunniteltiin niin, että graafisen etäkäyttöliittymän käyttäjä voi liikuttaa komponentin keskimmäistä osaa komponentin pohjan päällä. Joystick-komponentin piti lähettää numeerista tietoa sen ollessa aktiivinen. Tätä kyseistä tietoa voidaan käyttää apuna jonkin ulkopuolisen laitteen ohjauksessa. Tarkoituksena oli, että komponentin avulla voidaan ohjata servomoottoria.

Komponentti suunniteltiin lähettävän tietoa, missä kohtaa varsinaista komponentin aluetta eli pohjaa komponentin liikkuva osa sijaitsee. Tieto esitetään x- ja y-akseleiden avulla, joissa molemmissa nolla on pienin ja 100 on suurin määritelty arvo. Komponentin liikkuva osa on sijoitettu ensisijaisesti keskelle komponentin pohjaa, jossa sekä x- että y-akselin paikkatieto ovat molemmat 50, kuten kuvassa 17 on esitetty.



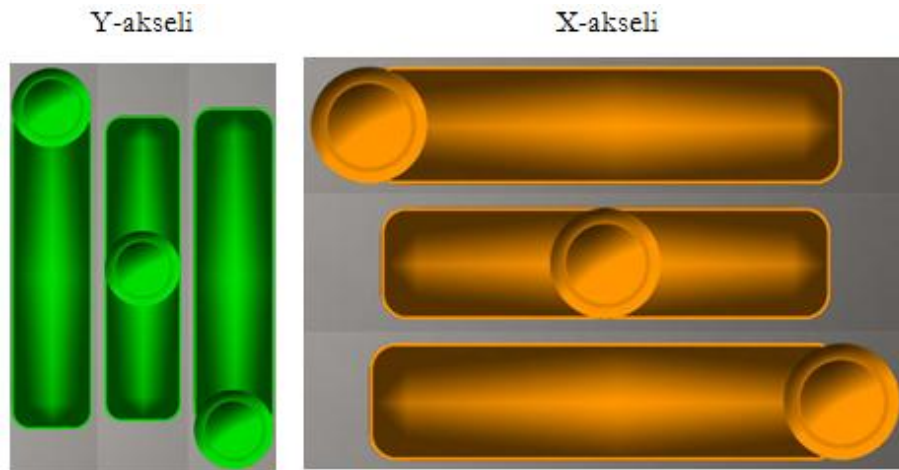
Kuva 17. Joystick-komponentti kahdella akselilla.

Komponentti suunniteltiin myös toimimaan pelkästään x- tai y-akselin suuntaisesti, riippuen komponentille annettujen korkeuden ja leveyden arvoista. Joystick-komponentille on voimassa ehto, jossa määritellään komponentin käyttävän molempia akseleita, kun komponentin korkeus ja leveys ovat samat. Tällöin kyseessä on neliön muotoinen komponentti, jota voidaan liikuttaa kuvan 18 mukaisesti. Komponentin liikkuva osa liikkuu sulavasti käyttäjän painaessa komponentti aktiiviseksi ja liikuttamalla tätä esimerkiksi tietokoneen hiirellä tai sormella kosketusnäytöllisessä mobiililaitteessa.



Kuva 18. Joystick-komponentin liikuttaminen x- ja y-akseleiden suhteen.

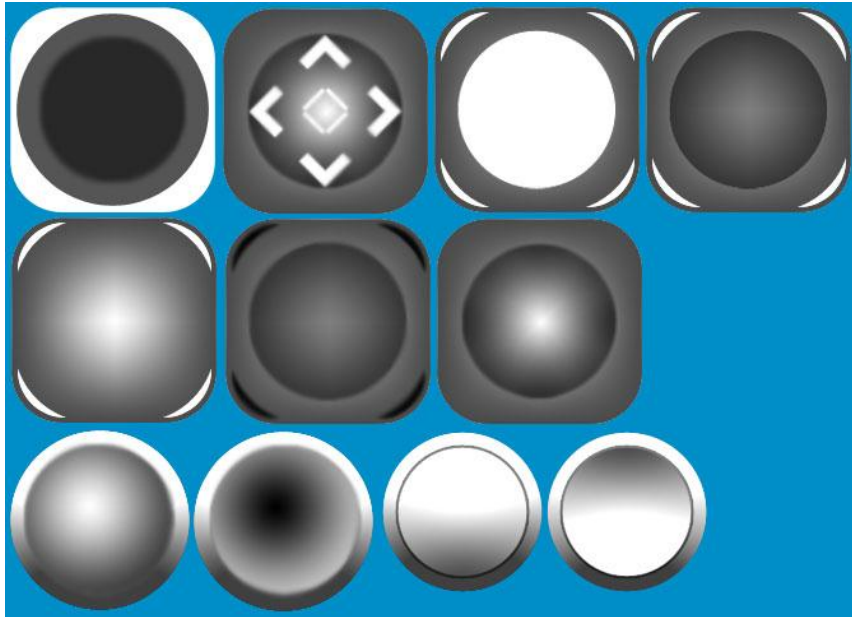
Kuvassa 19 on esitetty erilliset x- ja y-akselin joystick-komponentit. Komponentille annetun korkeuden ollessa suurempi kuin leveys on joystick-komponentilla käytössä vain y-akseli. Komponentilla on käytössä vain x-akseli, jos leveys on suurempi kuin korkeus.



Kuva 19. Joystick-komponentin liikuttaminen x- ja y-akseleiden suhteen.

## 8.2 Komponentin toteutus

Komponentti toteutettiin käyttäen pohjana KajaPro Oy:n etäkäyttöliittymän olemassa olevien komponenttien runkoa. Microsoft Visual Studio 2010 -ohjelmaa käytettiin komponentin ohjelmoimiseen C++-ohjelmointikielellä. Joystick-komponentti toimii graafisen etäkäyttöliittymän määrittelemien sääntöjen mukaisesti. Komponentille toteutettiin mustavalkoisena oma graafinen ulkoasu, jonka eri variaatioita on esitetty kuvassa 20. Kuvan ylä- ja keskirivillä on komponentin seitsemän eri pohjaa ja alarivillä on komponentin kaksi liikuteltavaa osaa kahtena osana, joista toista käytetään, kun komponentti on painettu pohjaan, eli komponentti on tällöin aktiivisena.



Kuva 20. Joystick-komponentin graafisen ulkoasun eri variaatiot.

Joystick-komponentille voidaan asettaa voimaan suoraan joko ohjelmakoodissa tai kääntäjälle lippuparametri, jonka avulla komponentin viereen saadaan näkyviin x- ja y-akseleiden arvot.

### 8.3 Komponentin testaus

Joystick-komponenttia testattiin Android 2.1 -käyttöjärjestelmän omaavalla tablet-tietokoneella, Windows 7:llä ja Raspberry Pi:llä. Komponentin toteutusvaiheessa testauksia suoritettiin vain Windows 7:llä. Testauksia suoritettiin tablet-tietokoneella komponentin ensimmäisen version ollessa valmis. Testauksessa varmistettiin, että komponentti lähetti tietoa ja toinen laite otti tiedon vastaan onnistuneesti.

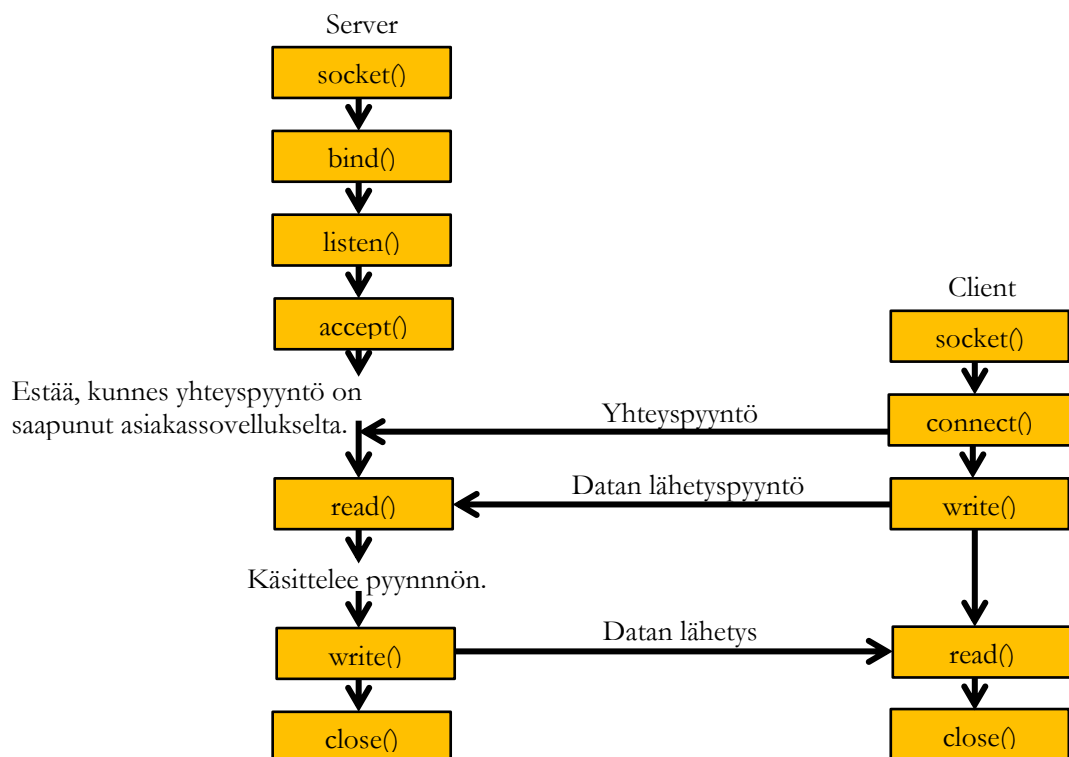
Raspberry Pi:llä komponenttia testattiin hercules-testityökalun avulla, joka lähetti Raspberry Pi:lle socket-yhteyden välityksellä graafisen käyttöliittymän, jossa oli kolme eri joystick-komponenttia. Yksi komponentti käytti sekä x- että y-akseleita, toinen pelkästään x-akselia ja kolmas y-akselia. Komponentille asetettiin näkyviin x- ja y-akseleiden arvot ohjelmakoodin käännöksen yhteydessä lipulla `-DSHOW_LABELS`. Testauksessa liikutettiin komponentin liikkuvaa osaa ja katsottiin, että x- ja y-akseleiden arvot ovat oikein.



## 9 SOCKET-YHTEYS

Socket on TCP (Transmission Control Protocol) tai UDP (User Datagram Protocol) internet-protokollan mukainen tietoliikenneyhteys. Socket-rajapinta eli socket API luodaan ja sitä käytetään ohjelma- tai funktiokutsujen avulla. TCP-protokollan mukaisesti socket on päätepiste tietokoneen ytimelle kaksisuuntaisen tietoliikenneyhteyden kahden eri sovelluksen kommunikoinnissa, jotka ovat käynnissä tietoverkossa. Yleisin API on Berkeley UNIX C -rajapinta socketeille. Socket on sidoksissa portin numeroon niin, että TCP/IP-kerros voi tunnistaa sovelluksen, jolle tieto on määrätty lähetettävän. Socketeja voidaan myös käyttää prosessien väliseen kommunikointiin samalla tietokoneella paikallisena yhteytenä tai etänä verkkoyhteyden avulla [42]. [43.] [44.]

Socket-tietoliikenneyhteys koostuu serveripalvelimesta tai -sovelluksesta sekä asiakassovelluksesta, jotka lähettävät tietoa toisilleen. Serveri on käynnissä yhdellä tietokoneella, jolla on socket, joka on sidoksissa tiettyyn tietokoneen porttiin. Kuvassa 21 on esitetty tyypillinen jakso serverin ja asiakassovelluksen TCP-protokollan mukaiselle socket-yhteydelle. [43.]



Kuva 21. TCP-protokollan mukaisen socket-yhteyden jakso [42, s. 67].

Asiakassovellus tietää laitteen isännän nimen, jolla serveri on käynnissä, sekä portin numeron, jota serveri kuuntelee. Toteuttaakseen yhteyspyynnön asiakassovellus yrittää ”tavata” serverin sen laitteessa ja portissa. Asiakassovelluksen täytyy myös tunnistaa itsensä serverille, jotta se sitoo itsensä paikallisen portin numeroon, jota käytetään tässä yhteydessä. Portin numero on yleensä järjestelmän määräämä. [43.]

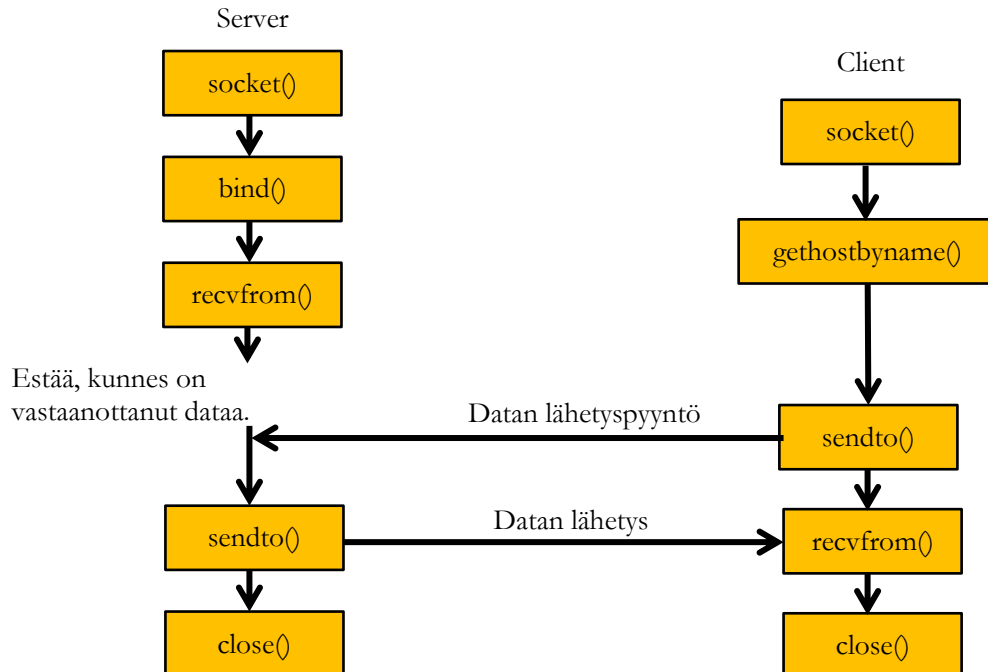
Serveri hyväksyy socket-yhteyden, jos yhteyspyynnön hyväksymisessä ei tule ongelmia. Usean asiakassovelluksen tapauksessa serveri luo uuden socketin jokaiselle asiakassovellukselle yhteyspyynnön hyväksynnän yhteydessä. Uusi socket on sidoksissa samaan paikalliseen porttiin, jolle on myös asetettu päätepiste asiakassovelluksen osoitteeseen ja porttiin. Tämän tarkoituksena on, että serveri voi jatkaa alkuperäisen socketin kuuntelemista yhteyspyyntöjä varten. Tällöin uusi socket-yhteys palvelee jonossa seuraavana olevaa asiakassovellusta. [43.]

Socket-yhteys on saatu muodostettua serverin ja asiakassovelluksen välille, jos yhteys on hyväksytty myös asiakassovelluksen puolelta. Asiakassovellus ja serveri voivat keskustella keskenään kirjoittamalla niiden väliseen socketiin ja lukemalla socketia, kun yhteys on muodostettu. [43.]

Kuvassa 21 serveri luo socket-yhteyden `socket()`-funktiolla, jonka jälkeen socket sidotaan ja nimetään tiettyyn porttiin `bind()`-funktiolla. Asiakassovellus luo myös oman socketin, jolloin serveri kuuntelee tulevaa yhteyspyyntöä `listen()`-funktiolla, ja lähettää yhteyspyynnön funktiolla `connect()`. Serveri hyväksyy asiakassovelluksen yhteyspyynnön `accept()`-funktiolla. Asiakassovellus lähettää datapyynnön serverille kirjoittamalla socketiin `write()`-funktion avulla. Serveri lukee ja käsittelee vastaanotetun pyynnön `read()`-funktiolla ja lähettää asiakassovellukselle sen pyytämän tiedon `write()`-funktiolla. Asiakassovellus ottaa tiedon vastaan `read()`-funktiolla ja sulkee socket-yhteyden `close()`-funktiolla. Myös serveri sulkee oman socket-yhteyden. [43.]

TCP-protokolla sisältää virheentarkistuksen lähetetyille datapaketeille. Virheentarkistuksessa varmistetaan, että paketit saapuvat asiakassovellukselle siinä järjestyksessä missä serveri oli ne lähettänytkin. UDP-protokolla ei sisällä kyseistä virheentarkistusta ja on näin ollen nopeampi tiedonsiirtotapa kuin TCP-protokollan mukaisesti toteutettu socket-yhteys. [45.]

Kuvassa 22 on esitetty tyypillinen jakso socket-yhteydessä serverisovellukselle yksisuuntaisessa, ”yhteydettömässä” ja ei-jaksollisessa UDP internet-protokollassa, jossa serveri käsittelee asiakassovellusten pyynnön. [44.]



Kuva 22. UDP-protokollan mukaisen socket-yhteyden jakso [42, s. 68].

Tässäkin tapauksessa molemmat päätteet luovat ensin socketit, jonka jälkeen serveri sitoo ja nimeää socketin paikalliseen porttiin. Asiakassovellus hakee nimipalvelusta serverin IP-osoitteen `gethostbyname()`-funktiolla. Kun yhteys on muodostettu, lähettää asiakassovellus datan lähetyspyynnön serverille, joka käsittelee datan `recvfrom()`-funktiolla. Serveri lähettää `sendto()`-funktiolla asiakassovellukselle tämän pyytämän datan, jonka asiakassovellus ottaa vastaan. Lopuksi molemmat päätteet sulkevat socket-yhteyden `close()`-funktiolla. [44.]

Socketteja voidaan myös käyttää orientoituneena vuorovaikutuksena erilaisena jaksona C-kielen järjestelmäkutsujen tai funktioiden avulla. [44.]

## 10 TESTAUS

KajaPro Oy:n graafisen etäkäyttöliittymän testaus suoritettiin Raspberry Pi:llä vaiheittain. Ensimmäisenä testattiin ristikäntäjän toimintaa GCC- ja G++-kääntäjillä, jonka jälkeen suoritettiin SDL- ja EGL-rajapintojen testaus. Näiden jälkeen testattiin socket-yhteys ja lopuksi KajaPro Oy:n graafinen etäkäyttöliittymä. Tässä yhteydessä Raspberry Pi käynnistää automaattisesti graafisen työpöydän. Manuaalisesti tämä voidaan käynnistää komentoriviltä komennolla **startx**.

Testauksen yhteydessä huomattiin, että etäkäyttöliittymä tarvitsee SDL-rajapinnan kirjastojen lisäykset ristikäntäjälle, jotta sovelluksen toiminnallisuus, kuten hiiren ja näppäimistön komennot, toimivat kohdeympäristössään. SDL-rajapinnan testauksen jälkeen testattiin EGL-rajapintaa. Ristikääntäjä tarvitsi myös EGL-kirjaston lisäyksen yrityksen etäkäyttöliittymän grafiikan esittämiseen.

Graafisen etäkäyttöliittymän avulla toteutettiin käyttöliittymä, jossa testattiin joystick-komponenttia Raspberry Pi:llä. Socket-yhteyden testaus suoritettiin viimeisenä vaiheena ennen varsinaisen valmiin kehitysympäristön testausta.

### Ristikääntäjän testaus

Ristikääntäjää testattiin aluksi Hello World -ohjelmalla, joka tulostaa komentoriville ”Hello World”. Ohjelma tehtiin erikseen C- ja C++-ohjelmointikielillä. Hello-sovellusta voitiin testata Raspberry Pi:n LXTerminal-ohjelman eli komentorivin kautta, kun ristikäännös oli suoritettu Windows 7:llä onnistuneesti GCC:n avulla. Tiedosto suoritetaan sen sijaitsemassa hakemistossa Raspbianilla komentoriviltä komennolla **./Hello**. Eli jos tiedosto sijaitsee esimerkiksi hakemistossa /home/pi/Desktop/Hello.exe, suoritetaan tiedosto kyseisessä hakemistossa edellä mainitulla komennolla.

### SDL-rajapinnan testaus

SDL-rajapinnan toimivuutta testattiin usealla eri sovelluksella, jotka käyttävät SDL-rajapinnan funktioita luodakseen grafiikkaa ja käyttääkseen hiiren ja näppäimistön toiminnallisuuksia. Kyseisinä testisovelluksina käytettiin ladatun SDL-paketin mukana tulleita

valmiita C-ohjelmointikielellä toteutettuja koodeja, jotka sijaitsevat puretun SDL-paketin test-kansiossa. Testisovellukset käännettiin suoritettaviksi tiedostoiksi ristikäntäjällä Windows 7 -ympäristössä.

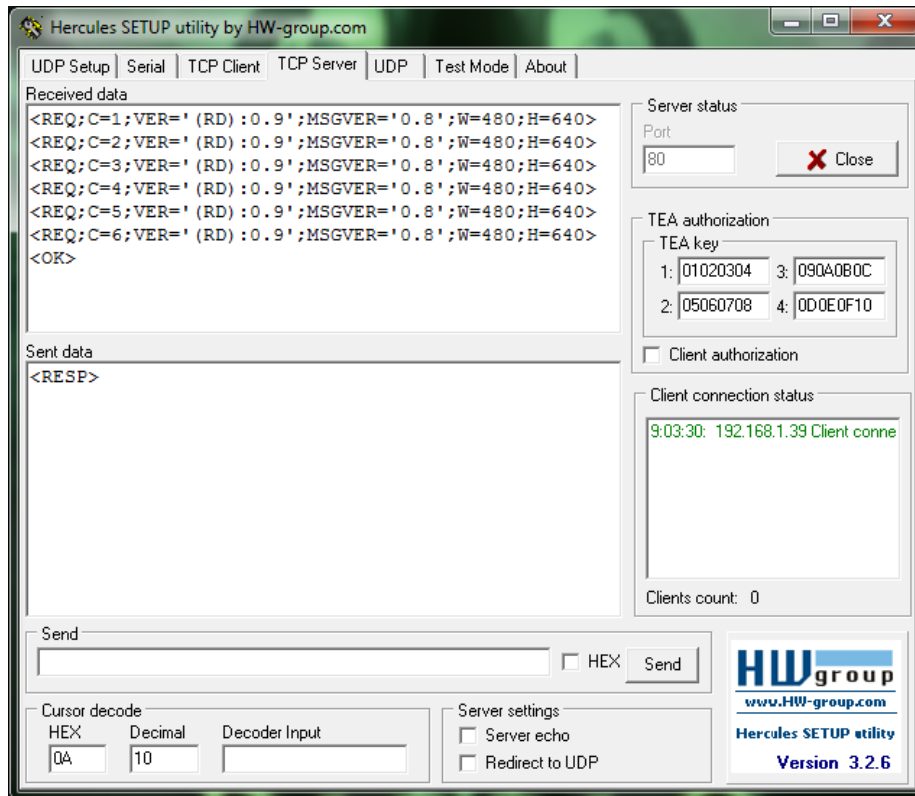
#### EGL-rajapinnan testaus

EGL-rajapinnan toimivuutta testattiin aluksi sovelluksella, joka käyttää OpenGL ES 2 -rajapintaa tuottamaan sovelluksen grafiikan EGL-rajapinnan avulla. Sovellus luo grafiikkaikkunan, joka vaihtaa väriä satunnaisesti sekunnin välein. Rajapinnan testauksessa ristikäntäjälle piti määritellä polku komennolla **-IC:\SysGCC\Raspberry\arm-linux-gnueabihf\include**, josta käänntäjä löytää EGL:n ja OpenGL ES 2:n header-tiedostot. Linkkerille piti määritellä kirjastojen sijainti komennolla **-LC:\SysGCC\Raspberry\arm-linux-gnueabihf\lib**. Linkitys EGL- ja OpenGL ES 2 -kirjastoihin suoritettiin komennoilla **-IEGL** ja **-IGLESv2**.

#### Socket-yhteyden testaus

Socket-yhteyttä testattiin erikseen hercules-testityökalun ja HTML-sivuston avulla. Hercules-testityökalun voi ladata ilmaiseksi osoitteesta [http://www.hw-group.com/products/hercules/index\\_en.html](http://www.hw-group.com/products/hercules/index_en.html). Testityökalua voidaan käyttää sarjaportin, TCP/IP- ja UDP/IP-tietoliikenteiden kommunikoinnissa. [46.]

Graafinen etäkäyttöliittymä ottaa asiakassovelluksena yhteyden Windows 7:ssä toimivaan hercules-testityökalun TCP-serveriin Windowsin IP-osoitteen avulla. Serveri kuuntelee porttia 80 ja vastaanottaa asiakassovelluksen lähettämän yhteyspyynnön <REQ>-komennon kuvan 23 mukaisesti, johon serveri vastaa yhteyspyynnön hyväksyvällä komennolla <RESP>.



Kuva 23. Hercules-testityökalu TCP-serverinä.

Kun yhteyden muodostus on suoritettu onnistuneesti, voi serveri lähettää asiakassovellukselle eli Raspberry Pi:ssä toimivalle sovellukselle graafisen käyttöliittymän. Tämä testattiin myös kahdella eri Raspberryllä samanaikaisesti, eli serveri pystyi lähettämään kahdelle asiakassovellukselle saman käyttöliittymän.

#### Kehitysympäristön testaus

Kehitysympäristön testaus suoritettiin asentamalla ristikääntäjä sekä SDL- ja EGL-rajapinnat tässä työssä annettujen ohjeiden mukaisesti. Ristikääntäjää testattiin Windows XP-, Windows 7- ja Windows 8 -käyttöjärjestelmillä. Graafinen etäkäyttöliittymäsovellus ristikäännettiin edellä mainituilla käyttöjärjestelmillä, ja jokainen sovellus testattiin Raspberry Pi:llä.

## 11 TYÖN ANALYSOINTI

### Joystick-komponentti

Tämä insinööri työ aloitettiin suunnittelemalla joystick-komponentin toiminnallisuutta yrityksen graafiseen etäkäyttöliittymään sille määriteltyjen vaatimusten mukaisesti. Yleiskäyttöisen komponentin avulla piti pystyä ohjaamaan servomoottoreita. Komponentin tekstuuri suunnittelu oli vapaata.

Joystick-komponentin toiminnallisuus toteutettiin C++-ohjelmointikielellä ja tekstuuri käyttäen Adobe Photoshop Elements 9 -ohjelmaa. Komponenttia testattiin graafisen käyttöliittymän avulla Windows- ja Android-käyttöjärjestelmillä, ja lopputuloksena komponentin avulla ohjattiin servomoottoreita onnistuneesti tablet-tietokoneella.

### Ristikääntäjä

Ristikääntäjää varten piti tehdä tutkimustyötä Raspberry Pi:stä ja yleisesti ristikääntäjästä. KajaPro Oy:n etäkäyttöliittymäsovelluksen grafiikan esitystapa suunniteltiin toteutettavaksi käyttäen OpenGL ES 2 -ohjelmointirajapintaa, jota Raspberry Pi tukee. Sovelluksen hiiren ja näppäimistön toiminnallisuus suunniteltiin toteutettavaksi käyttäen KajaPro Oy:n SDL-rajapintaa. Ristikäännös yrityksen sovellukselle ajateltiin suoritettavan suoraan Windowsin komentoriviltä ilman muita asennettavia ohjelmia.

GNU-työkalun avulla toteutettuun ristikääntäjään piti toteuttaa SDL- ja EGL-rajapintojen tuet. Tämä vaati tutkimustyötä rajapintojen toteutuksen kannalta sekä siitä, miten ne voidaan yhdistää toisiinsa. Tutkimuksissa huomattiin, että ristikääntäjä vaati rajapintojen header-tiedostojen sisällytyksen ja kirjastojen linkityksen, jotta suoritettava tiedosto voidaan luoda. Myös Raspberry Pi vaati niiden rajapintojen kirjastot, joita sovellus käyttää.

### Kehitysympäristö

Kehitysympäristön ja ristikääntäjän käyttöönotto eivät tuoneet suuria ongelmia. Tutkimustyön tuloksina selvisi, mitä tiedostoja ja kirjastoja ristikääntäjä tarvitsi, jotta KajaPro Oy:n ohjelmistokehitys saadaan käännettyä ja toimimaan Raspberry Pi:n laitearkkitehtuurilla.

SDL- ja EGL-rajapintojen yhdistäminen toisiinsa tuotti haasteita tässä työssä. EGL-rajapinta loi oman ikkunan grafiikalle, joka on toteutettu OpenGL ES 2 -ohjelmointirajapinnalla. SDL-rajapinta loi oman ikkunan näppäimistön ja hiiren toiminnallisuudelle. Haasteena oli saada nämä kaksi ikkunaa kytkettyä toisiinsa SDL-ikkunakahvan avulla. SDL:n hiiren toiminnallisuus jää grafiikkaikkunan taakse SDL-ikkunakahvan toimimattomuuden vuoksi.

Socket-yhteyden testaus suoriutui ongelmitta Raspberry Pi:llä. Asiakassovelluksena toiminut Raspberry Pi sai graafisen käyttöliittymän Windows 7 -käyttöjärjestelmässä toimivalta TCP-serveriltä, jota käytettiin hercules-testityökalulla. Käyttöliittymässä ollut joystick-komponentti toimi määritysten mukaisesti.

#### Kehitys- ja parannusideoita

Graafisen etäkäyttöliittymän toiminnallisuutta voidaan testata Raspberry Pi:ssä GDB:n avulla. Ristikääntäjä olisi mahdollista lisätä Eclipse-ohjelmointiympäristöön. Eclipsellä pystyy myös muodostamaan SSH-yhteyden Raspberry Pi -laitteeseen. Tämä kokonaisuus mahdollistaisi etäkäyttöliittymän testauksen yhdellä ohjelmalla.



## 12 YHTEENVETO

Insinööriyöprosessin aikana on perehdytty graafisen etäkäyttöliittymän ristikäntämiseen eli lähdekoodin kääntämiseen toiselle laitteiston ohjelmistoarkkitehtuurille. Työn päätavoitteena oli toteuttaa ristikäntäjä Windows 7 -käyttöjärjestelmälle. Tämän avulla KajaPro Oy:n graafisen etäkäyttöliittymän lähdekoodi saadaan käännettyä Raspberry Pi -sulautetulla laitteella toimivaksi suoritettavaksi tiedostoksi.

Työ vaati tutkimustyötä Raspberry Pi -sulautetusta laitteesta, ristikäntäjästä sekä SDL- ja EGL-rajapinnoista. Graafiseen etäkäyttöliittymään piti toteuttaa C++-ohjelmointikielellä tuki Raspberry Pi:lle ja sen vaatimalle EGL-rajapinnalle. Sulautetulla laitteella piti myös testata socket-tietoliikenneyhteyden toimivuus sekä hercules-testityökalun että HTML-sivuston avulla.

Yrityksen sovellukseen oli tarkoitus toteuttaa yleiskäyttöinen joystick-komponentti. Tätä komponenttia käytettiin Android-mobiililaitteella servomoottoreiden ohjauksessa. Komponentin suunnitteluvaiheessa on pyritty ottamaan huomioon sen sovellettava käyttötarkoitus eri sovelluskohteissa.

Insinööriyö täyttää hyvin sille asetetut vaatimukset ja tavoitteet. Työn lopputuloksena toteutettu kehitysympäristö, ristikäntäjän käyttöohjeet sekä joystick-komponentti ovat käytössä KajaPro Oy:llä.

## LÄHTEET

- 1 Raspberry Pi. Downloads. [WWW-dokumentti].  
<<http://www.raspberrypi.org/downloads>>. (Luettu 27.01.2014)
- 2 Raspberry Pi. About us. [WWW-dokumentti].  
<<http://www.raspberrypi.org/about>>. (Luettu 15.02.2014)
- 3 Raspbian. Welcome to Raspbian. [WWW-dokumentti].  
<<http://www.raspbian.org/>>. (Luettu 04.02.2014)
- 4 Simple DirectMedia Layer. About SDL. [WWW-dokumentti].  
<<http://www.libsdl.org/>>. (Luettu 27.01.2014)
- 5 Khronos. Native Platform Interface. [WWW-dokumentti].  
<<https://www.khronos.org/egl/>>. (Luettu 27.01.2014)
- 6 The Guardian. Moorhead, J. Raspberry Pi device will 'reboot computing in schools'. [WWW-dokumentti].  
<<http://www.theguardian.com/education/2012/jan/09/raspberry-pi-computer-revolutionise-computing-schools?newsfeed=true>>. (Julkaistu 09.01.2012)
- 7 University of Cambridge. Raspberry Pi. [WWW-dokumentti].  
<<http://www.cl.cam.ac.uk/projects/raspberrypi/>>. (Luettu 18.02.2014)
- 8 element14. Raspberry Pi. [WWW-dokumentti].  
<<http://downloads.element14.com/raspberryPi1.html>>. (Luettu 28.01.2014)
- 9 Raspberry Pi. FAQs. [WWW-dokumentti].  
<<http://www.raspberrypi.org/faqs>>. (Luettu 27.01.2014)
- 10 eLinux. RPi Low-level peripherals. [WWW-dokumentti].  
<[http://elinux.org/RPi\\_Low-level\\_peripherals](http://elinux.org/RPi_Low-level_peripherals)>. (Luettu 29.01.2014)
- 11 Debian. Tietoa Debianista. [WWW-dokumentti].  
<<https://www.debian.org/intro/about>>. (Luettu 16.02.2014)
- 12 Debian. Debian ”wheezy” julkaisutietoja. [WWW-dokumentti].  
<<https://www.debian.org/releases/stable/>>. (Luettu 16.02.2014)
- 13 Raspbian. Raspbian FAQs. [WWW-dokumentti].  
<<http://www.raspbian.org/RaspbianFAQ>>. (Luettu 27.01.2014)
- 14 Upton E. & Halfacree G. Raspberry Pi User Guide, Second Edition. Sisäinen asiakirja. (Luettu 04.02.2014)
- 15 LinuxGizmos. U.S. swallows first \$25 Raspberry Pi batch in one gulp. [WWW-dokumentti]. <<http://linuxgizmos.com/raspberry-pi-model-a-arrives/>>. (Julkaistu 03.04.2013)

- 16 Raspberry Pi. What are the power requirements? [WWW-dokumentti].  
<<http://www.raspberrypi.org/faqs#powerReqs>>. (Luettu 24.02.2014)
- 17 Intel. Network Connectivity. [WWW-dokumentti].  
<<http://www.intel.com/support/express/switches/sb/CS-014409.htm>>. (Luettu 24.02.2014)
- 18 Raspberry Pi Spy. Raspberry Pi Status LEDs Explained. [WWW-dokumentti].  
<<http://www.raspberrypi-spy.co.uk/2013/02/raspberrypi-status-leds-explained/>>. (Julkaistu 10.02.2013)
- 19 Khronos. The Standard for Media Library Portability. [WWW-dokumentti].  
<<http://www.khronos.org/openmax/>>. (Luettu 24.02.2014)
- 20 Raspberry Pi. Bradbury, A. Open source ARM userland. 24.10.2012. [WWW-dokumentti]. <<http://www.raspberrypi.org/archives/2221>>. (Luettu 15.02.2014)
- 21 Khronos. The Standard for Vector Graphics Acceleration. [WWW-dokumentti]. <<http://www.khronos.org/opencv/>>. (Luettu 27.01.2014)
- 22 Khronos. The Standard for Embedded Accelerated 3D Graphics. [WWW-dokumentti]. <<http://www.khronos.org/openssl/>>. (Luettu 27.01.2014)
- 23 Simple DirectMedia Layer. Licensing the Simple DirectMedia Layer library. [WWW-dokumentti]. <<http://www.libsdl.org/license.php>>. (Luettu 28.01.2014)
- 24 Petersen Richard. Ubuntu The Complete Reference, McGraw-Hill , 2009. ISBN 978-0-07-159846-0
- 25 GNU. A Brief History of GCC. [WWW-dokumentti].  
<<http://gcc.gnu.org/wiki/History>>. (Luettu 24.02.2014)
- 26 Memory Management&GNU työkalut. Sisäinen asiakirja. [PDF-dokumentti]. (Luettu 04.02.2014)
- 27 GNU. Host/Target specific installation notes for GCC. [WWW-dokumentti].  
<<http://gcc.gnu.org/install/specific.html>>. (Päivitetty 27.03.2014)
- 28 GNU. Language Standards Supported by GCC. [WWW-dokumentti].  
<<http://gcc.gnu.org/onlinedocs/gcc-4.6.3/gcc/Standards.html#Standards>>. (Luettu 28.01.2014)
- 29 Delorie. Summary of GDB. Päivitetty keskuu 2003. [WWW-dokumentti].  
<[http://www.delorie.com/gnu/docs/gdb/gdb\\_98.html](http://www.delorie.com/gnu/docs/gdb/gdb_98.html)>. (Luettu 18.02.2014)
- 30 GNU. GDB: The GNU Project Debugger. Muokattu 06.02.2014. [WWW-dokumentti]. <<http://www.gnu.org/software/gdb/>>. (Luettu 18.02.2014)

- 31 Carnegie Mellon University. Debuggin Under UNIX: gdb Tutorial. [WWW-dokumentti]. <<http://www.cs.cmu.edu/~gilpin/tutorial/>>. (Päivitetty 07.04.2004)
- 32 YoLinux. GNU GDB Debugger Command Cheat Sheet. [WWW-dokumentti].<<http://www.yolinux.com/TUTORIALS/GDB-Commands.html>>. (Päivitetty 27.03.2014)
- 33 GNU. Options Controlling the Kind of Output. [WWW-dokumentti]. <<http://gcc.gnu.org/onlinedocs/gcc-3.4.6/gcc/Overall-Options.html#Overall-Options>>. (Luettu 04.02.2014)
- 34 GNU. The C Preprocessor. [WWW-dokumentti]. <<http://gcc.gnu.org/onlinedocs/cpp/>>. (Luettu 04.02.2014)
- 35 MyLinuxBook. Basics of GCC compilation process. [WWW-dokumentti]. <<http://mylinuxbook.com/basics-of-gcc-compilation-process-explained/>>. (Julkaistu 09.02.2012)
- 36 GNU. What is GNU? [WWW-dokumentti]. <<http://www.gnu.org/>>. (Päivitetty 16.01.2014)
- 37 Gnutoolchain. Windows toolchain for Raspberry/PI. [WWW-dokumentti]. <<http://gnutoolchains.com/raspberry/>>. (Luettu 27.01.2014)
- 38 ELinux. Raspberry Pi VideoCore APIs. [WWW-dokumentti]. <[http://elinux.org/Raspberry\\_Pi\\_VideoCore\\_APIs#libbcm\\_host](http://elinux.org/Raspberry_Pi_VideoCore_APIs#libbcm_host)>. (Päivitetty 28.12.2013)
- 39 GNU. GCC Command Options. [WWW-dokumentti]. <[http://gcc.gnu.org/onlinedocs/gcc-2.95.2/gcc\\_2.html](http://gcc.gnu.org/onlinedocs/gcc-2.95.2/gcc_2.html)>. (Julkaistu 17.03.2001)
- 40 TechTarget. Secure Shell (SSH). [WWW-dokumentti]. <<http://searchsecurity.techtarget.com/definition/Secure-Shell>>. (Luettu 03.03.2014)
- 41 DigitalOcean. Avoid Duplicate SSH Host Keys. [WWW-dokumentti]. <<https://www.digitalocean.com/company/blog/avoid-duplicate-ssh-host-keys/>>. (Päivitetty 26.07.2013)
- 42 Linux and system programming day 5. Sisäinen materiaali. [PDF-dokumentti]. (Luettu 27.02.2014)
- 43 Oracle. What Is a Socket? [WWW-dokumentti]. <<http://docs.oracle.com/javase/tutorial/networking/sockets/definition.html>>. (Luettu 15.02.2014)
- 44 TechTarget. Harschutz, J. sockets. Päivitetty huhtikuussa 2005. [WWW-dokumentti]. <<http://whatis.techtarget.com/definition/sockets>>. (Luettu 15.02.2014)

- 45 Diffeen. TCP vs UDP. [WWW-dokumentti]. <[http://www.diffeen.com/difference/TCP vs UDP](http://www.diffeen.com/difference/TCP_vs_UDP)>. (Luettu 27.2.2014)
- 46 HW-group. Hercules SETUP utility. [WWW-dokumentti]. <[http://www.hw-group.com/products/hercules/index\\_en.html](http://www.hw-group.com/products/hercules/index_en.html)>. (Luettu 24.02.2014)