Lauri Marjanen, Tuomas Autio, Taneli Voutilainen

XYPlotter

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tutkinto-ohjelman nimi

Embedded Systems Project

5.10.2020

|  |  |  |
| --- | --- | --- |
| Tekijä  Otsikko  Sivumäärä  Aika | | Etunimi Sukunimi  Insinöörityön otsikko  xx sivua + x liitettä  30.8.2018 |
| Tutkinto | | insinööri (AMK) |
| Tutkinto-ohjelma | | tutkinto-ohjelman nimi |
| Ammatillinen pääaine | | ammatillisen pääaineen nimi |
| Ohjaajat | | tehtävänimike Etunimi Sukunimi  tehtävänimike Etunimi Sukunimi |
|  | | |
| Avainsanat |  | |

|  |  |  |
| --- | --- | --- |
| Author  Title  Number of Pages  Date | First name Last name  Title of the Thesis  xx pages + x appendices  30 August 2018 | |
| Degree | Bachelor of Engineering | |
| Degree Programme | Name of the degree programme | |
| Professional Major | Name of the professional major | |
| Instructors | First name Last name, Title (for example: Project Manager)  First name Last name, Title (for example: Principal Lecturer) | |
|  | | |
| Keywords | |  |

Sisällys

[1 Intro 2](#_Toc52290371)

[2 Hardware 3](#_Toc52290372)

[2.1 LPCXpresso 1549 3](#_Toc52290373)

[2.2 XYPlotter 3](#_Toc52290374)

[2.3 Signal Capture Board 3](#_Toc52290375)

[3 Software 4](#_Toc52290376)

[3.1 FreeRTOS 4](#_Toc52290377)

[3.2 mDraw 4](#_Toc52290378)

[3.3 The Code? 4](#_Toc52290379)

[4 Outcome 5](#_Toc52290380)

Liitteet

Liite 1. Liitteen nimi

Liite 2. Liitteen nimi

# Intro

The main goal for this project was to accomplish a working solution to draw with the XY-Plotter using LPCXpresso 1549 microcontroller. This project was done for the Embedded Systems Project course. This project employed three students.

# Hardware

## LPCXpresso 1549

The LPCXpresso 1549 Cortex-M3 is a low budget microcontroller from NXP. The LPC1549 is an ARM-based developments platform. The microcontroller was chosen by the course teacher for this project.

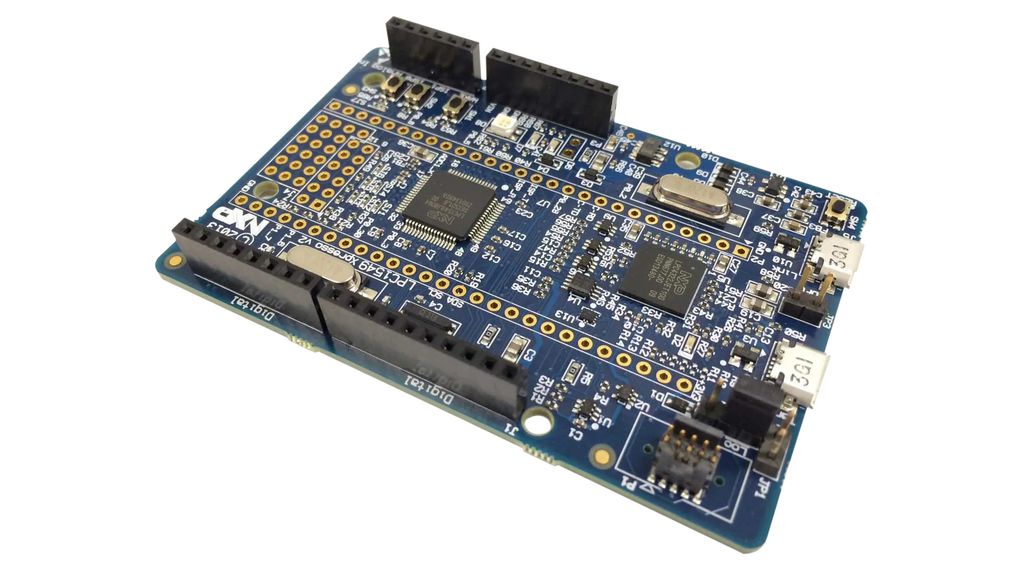


Figure 1: Picture of the LPC1549

## XYPlotter

XYPlotter is a robot kit provided by the Makeblock company. XYPlotter is used to move a pen or other drawing instruments to create artwork on a flat surface. It can also be used as a laser engraver.

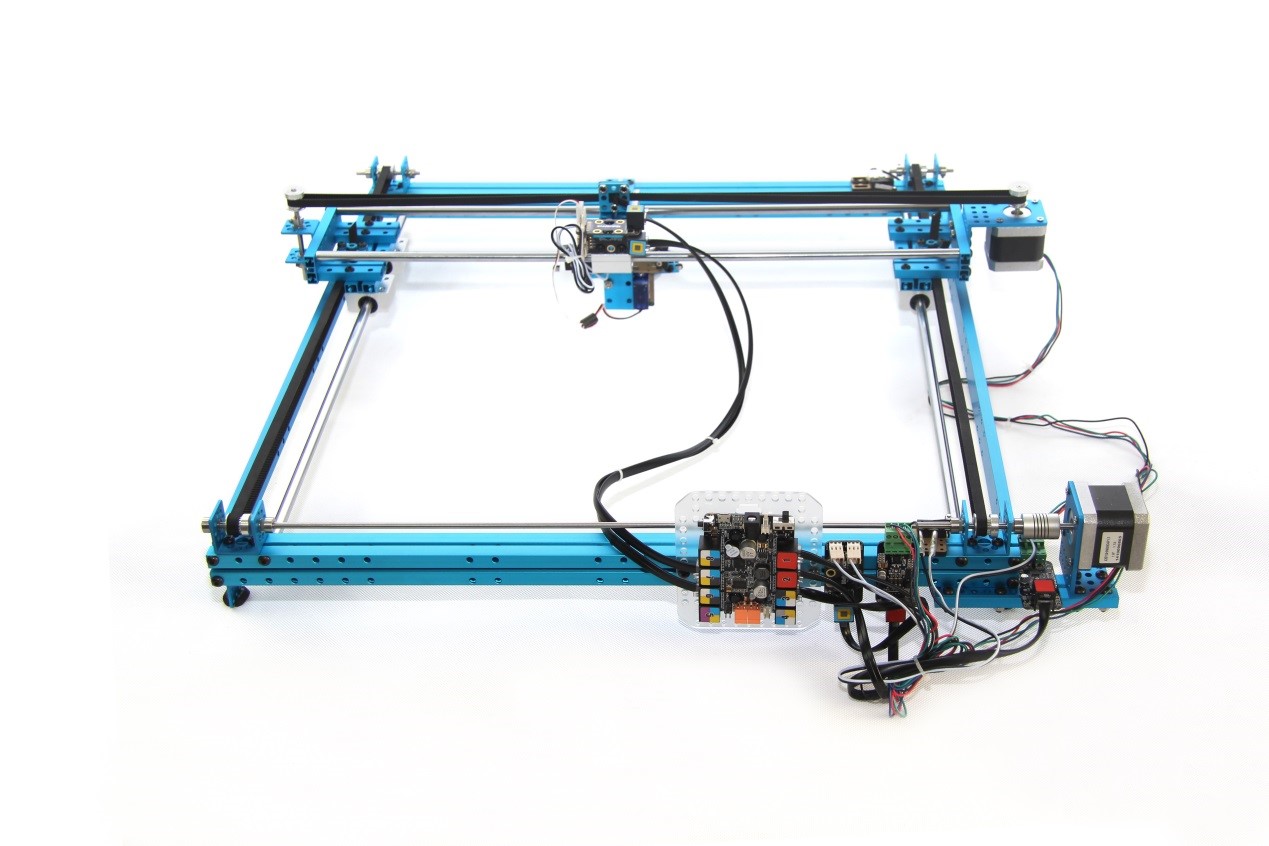


Figure 2: Makeblock's XYPlotter

The XYPlotter uses stepper motors to move in the XY-axis and can be controlled by sending pulses from the microcontroller. Limit switches are used to stop overflowing from the surface and activation stops the motors. Pen is controlled by a servo motor which lifts the pen from the paper. The laser engraver was supported in this plotter and can be used. The laser is a diode laser and can burn pictures to paper.

## Signal Capture Board

Signal capture board was used to test the plotter via simulator application. Project was made during the Covid-19 pandemic which meant there was limited access to the plotters.

# Software

## FreeRTOS

Helping the development side of the project was provided but the FreeRTOS project. FreeRTOS is a real-time operating system kernel distributed under the MIT License. FreeRTOS library is small and simple and easy to use and provides all the necessary functionalities for example: multithreading and tasks, mutexes, semaphores, timers etc.

## mDraw

mDraw is a piece of software by the Makeblock company used for many drawing projects. mDraw can digest SVG images and provide the gcode for the specific image. The commands are parsed by the software that was written for this project.

M10

G28

G1 X-0.50 Y101.50 A0

M1 90

G1 X149.50 Y101.50 A0

G1 X149.50 Y1.41 A0

G1 X-0.50 Y1.41 A0

G1 X-0.50 Y101.50 A0

Figure 3: Example output from mDraw

After getting the code from mDraw it was parsed by the software and according instructions were passed to the according pieces of hardware.

|  |  |  |
| --- | --- | --- |
| GCode | Inputs | Description |
| M10 | - | Start command |
| M1 | y | Set pen position |
| M4 | power | Change laser power |
| G1 | x, y | Go to position |
| G28 | - | Go to origin |

## Software

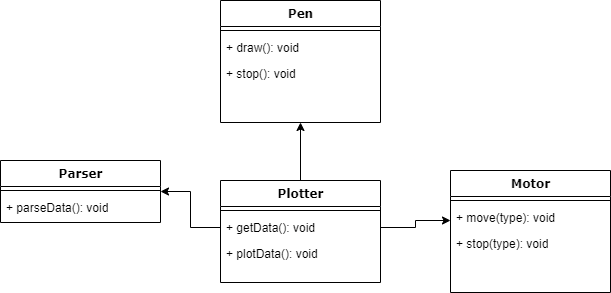
The whole software was written in C/C++. FreeRTOS library and standard libraries offered much help in the making process. First plan we had drafted looked something like this. 

Figure 4: Object diagram

We followed the plan quite a lot, but some changes needed to be made while we learned new things. We needed to add more classes to convert UART input to string and add some more functionality to other classes.

# Outcome