Aalto University

Keyboard project

MEC-E5010 Advanced Project on Mechatronics D

Koivistoinen Pyry, 100648540
8-29-2023

Contents

Abbrevations:

CAM	Computer Aided Manufacturing
CAD	Computer Aided Design
DOF	Degrees of freedom
MCU	Microcontroller unit
PLA	Polylactic Acid
PCB	Printed Circuit Board
QMK	Quantum Mechanical Keyboard

1. Abstract

This project is about designing and manufacturing a mechanical keyboard from scratch. Project includes mechanical and electrical design with some programming. The aim of the project is to make a functional keyboard, learn mechanical design in the limits of what is possible to manufacture and learn electrical design and about keyboard matrices.

2. Introduction

2.1 Motivation

I had over 7 years old Rantopad MXX keyboard and I grew tired of the loud switches it had. I mostly use a keyboard for typing, so the switches need to feel good, and they need to be silent. I had been planning on buying a new mechanical keyboard soon, but I decided it would be more interesting to try to build one myself instead of buying. There would be a chance to learn mechanical and electrical design at the same time and create something myself from the scratch.

2.2 Inspiration

Early inspiration for this project was Varmilo VA88M. Its sleek and simple design pleased the eye, and it fits well with simplified pc setup. Keyboard layout is TKL, tenkeyless, or 80%, meaning that it doesn't have number pad on the right side of the keyboard. Example of this board can be seen in figure 1. The color theme in CMYK model was pleasant: White background and colorful accents. Another inspiration came from TGR Jane v2. The case's shape and angle looked good. Additionally, Jane v2 had a F13 extra function key, that seemed like a nice addition. Jane v2 uses ANSI layout for the keys which looks more compact than ISO layout.

Figure 1. Varmilo VA88M with CMYK color theme.

2.3 Requirements and budget

When doing the research, Varmilo VA88M was around 120 €. The plan was to aim at similar price, keeping in mind that there might be mistakes in the design process, and redesigns would cost more.

More expensive keyboard use aluminum or brass cases or have extra weights. This keeps the keyboard steady while typing, makes the case more rigid, which reduced different feelings in key presses and makes the overall quality feel better.

Most important requirements for the keyboard are as follows:

- Heavy aluminum case.
- TKL with ANSI layout.
- Sleek and aesthetically good-looking design.
- Tactile and silent switches.
- USB-C cable.

All requirements should be met. Keyboard assembly should be as cheap as possible when requirements are met.

2.4 Required tools

For this project it was necessary to have a way to machine the case and assemble the PCB. For my disposal there was a machining center and soldering equipment. 3d-printer or laser cutter was also required.

3. Methods

3.1 Component design

Component design was done by using Siemens NX. PCB design was done using KiCad. Explosion view of all components can be seen in figure 2. Components are numbered and corresponding names can be found from table 1.

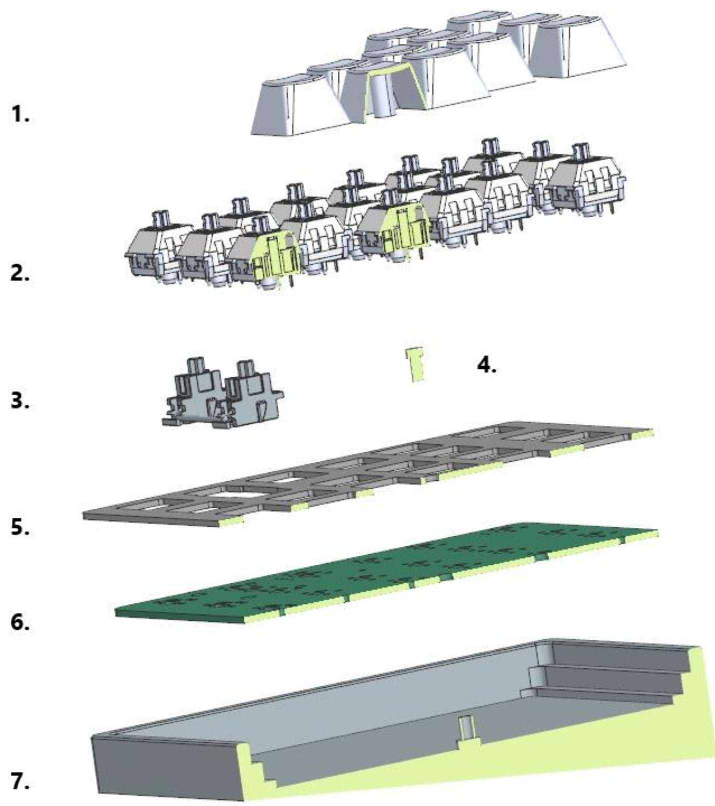

Figure 2. Explosion view of all components. Figure is a section of the keyboard.

Table 1. Corresponding names for components from figure 2.

Part	Name
1.	Keycaps
2.	Switches
3.	Switch stabilizer
4.	M3 x 6 screw
5.	Plate
6.	PCB
7.	Case

3.1.1 Case

The case was decided to be made from one piece to ease the manufacturing. The weight of the keyboard was important. It needed to feel heavy, so the keyboard stays firmly in place and doesn't move around while typing. Material chosen for the case was Al6082 for its availability and mechanical properties. The estimated weight for the case was 1800 g**.** Case design can be seen in figure 3.

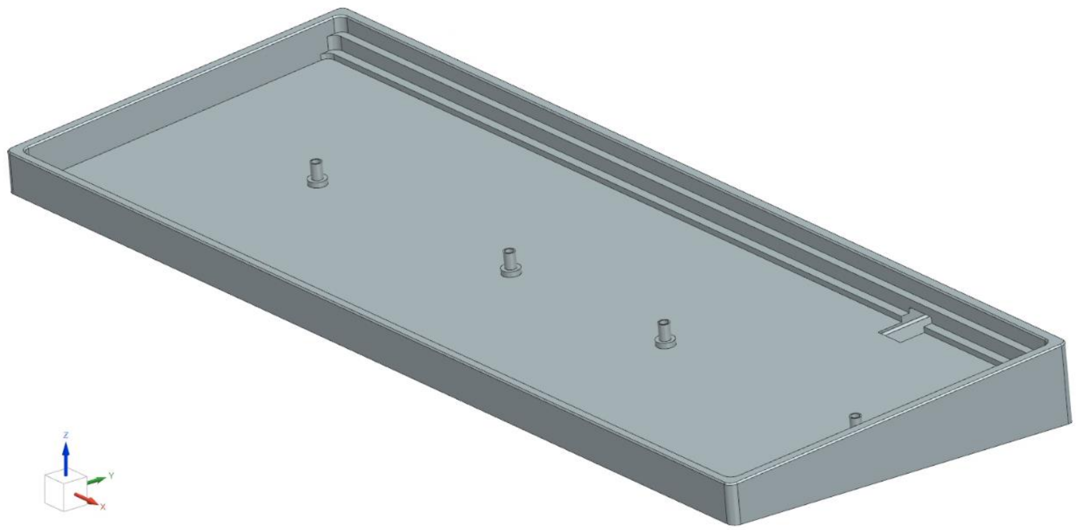

Figure 3. Case design in Siemens NX.

Case acts as a base for the whole keyboard. All other components needed to be able to be fastened to the case. For securing method, four shafts that had a thread inside were machined inside the case. Keyboard's PCB and plate were constrained with the shafts and pressed down with bolts that prevented the components from moving. Cross section of the fastening method can be seen in figure 4. Screw is pressing plate down for 0,5 mm which prevents it from rattling.

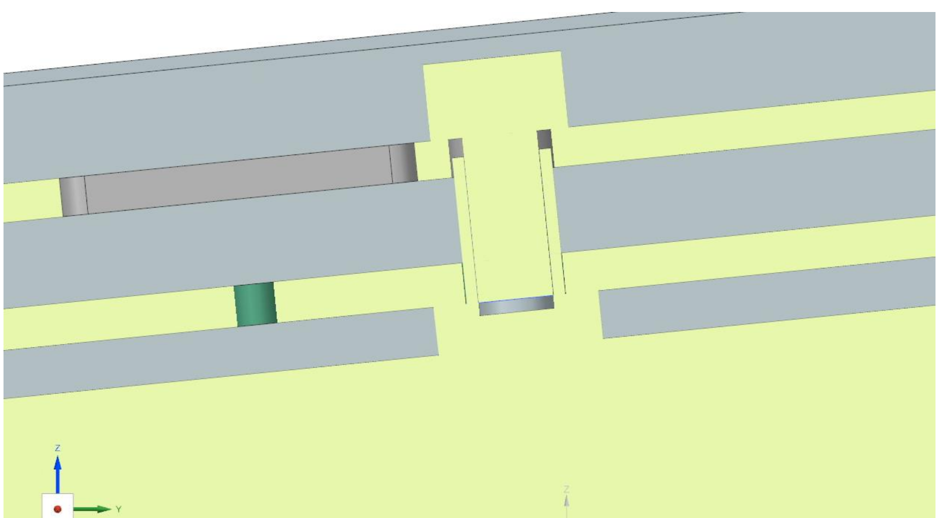

Figure 4. Cross section of keyboard fastening method.

3.1.2 Switches

My previous keyboard had Kailh Blue switches. The switch type is tactile and loud. I like the tactile bump while typing but prefer more silent keys.
The best option was Holy Panda switches which are advertised as the most tactile switches in the world [1]. There was no opportunity to test the switches out, but based on the sound tests they seemed good. Unfortunately, set of 90 switches would've costed 87 euros, which made them inaccessible for this project.
Next option was Cherry MX Brown switches. They are tactile, more silent compared to blue switches and very common. Because of their commonness they are accessible with competitive price.
One of the requirements for the keyboard was to make it as cheap as possible. Because of this it was decided to buy old keyboard which had the desired switches instead of buying the switches new. Cheap used keyboard was found from tori.fi that had the desired switches. It was bought and switches removed. The PCB also had a led for every key. They can be used as indicator leds.

3.1.3 Plate

Plate is the piece locking the keycaps in place and keeping them from moving. Plate is the most top part. It acts as a lid for the keyboard and will be visible in some areas. Since the case is made from aluminum, the plate should be too. This matches the materials and gives better structural properties compared to e.g. PLA plastic.

Plate had to be 1,5 mm thick, so the switches can lock properly. Hole sizes were done according to Cherry's datasheet [2].

3.1.4 Keyboard units

When designing a keyboard, keycap size standard is commonly used. The standard specifies keycap size, placeholder space, distance between switches and hole for the plate. Standard of keyboard unit can be seen in figure 5. Normal key is one unit, but some keys are different sizes, space bar being the longest one at 6,25 units.

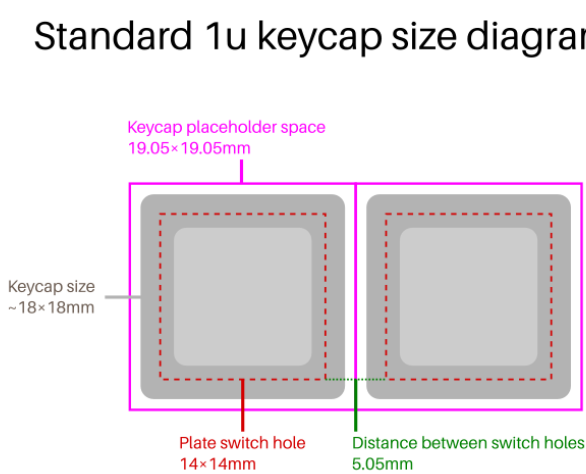

Figure 5. Standard dimensions for keycap size.

3.1.5 Switch stabilizers

It is required to have a switch stabilizer for keycaps longer than 2 units. The stabilizer consists of two buttons and a rod connecting them. When the key is pressed, the keycap is supported from the whole area, and it makes the keypress more stable and haptically better. Switch stabilizers were snapped in place to the plate to make them rattle less while being pressed.

3.1.6 Keycaps

Varmilo VA88M had a pleasant color combination, so it was chosen to have something similar. The bright colors felt too intensive, so it was chosen to have a white base with blue or green pastel accents. In AliExpress there was a PBT keycap set that had the correct color theme. The selected keycap set can be seen in figure 6.

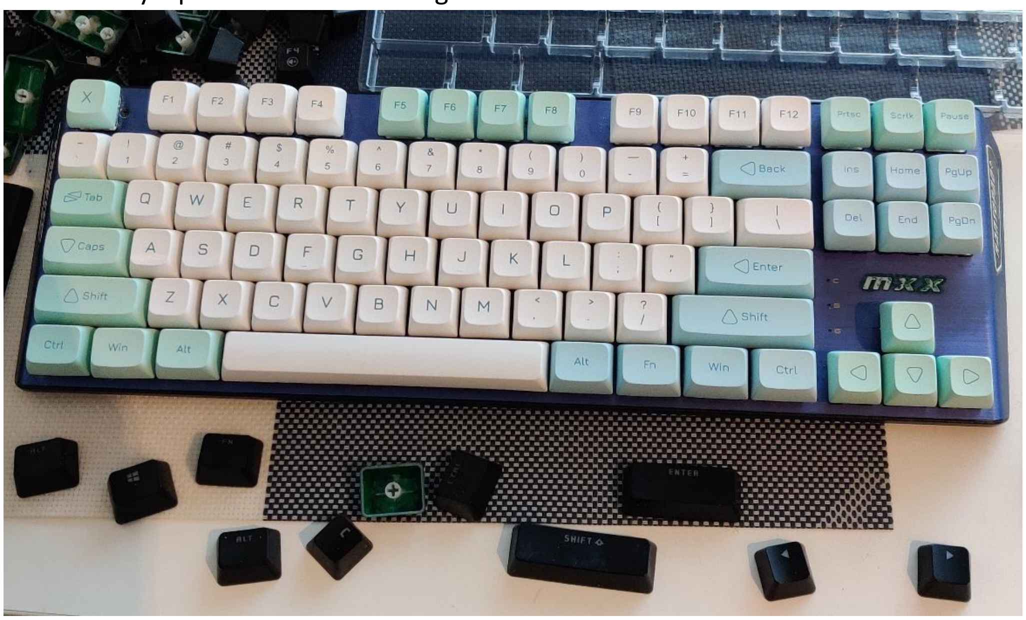

Figure 6. Keycap set on the old keyboard.

3.2 Electrical design

3.2.1 Requirements

It is possible to buy readymade keyboard PCBs, but I wanted to make my own, because:

- I have some experience in PCB design.
- It is cheaper to make one yourself.
- I wanted to know how keyboards work.
- I designed the case, so I needed the PCB to fit and be able to be secured to the case.

3.2.2 Design and layout

I didn’t know where to start with the PCB design, so I searched online for open-source keyboard projects. I found one that was based on Hineyboard h88. Differing from normal TKL, this keyboard had F13 extra function key. This project was used as a base for the PCB design. Attachment points were modified, and the keyboard matrix redone, since it wasn’t functional.

3.2.3 Keyboard matrix

There are 87 keys in TKL keyboard. To have a functional keyboard, every key press should be detected. To achieve this the keyboard would require a microcontroller that has 87 input pins. Such microcontrollers exist, but there is much more efficient way of detecting key presses, which comes in a form of matrix.

Keyboard was distributed into rows and columns in a shape of matrix. Every key has assigned row and column. When a key is pressed, it connects one row to one column. This is detected in the MCU, and correct key is recognized. Matrix of 12 rows and 9 columns was chosen. It required only 21 IO pins rather than 87. Schematic of matrix can be seen in figure 7.

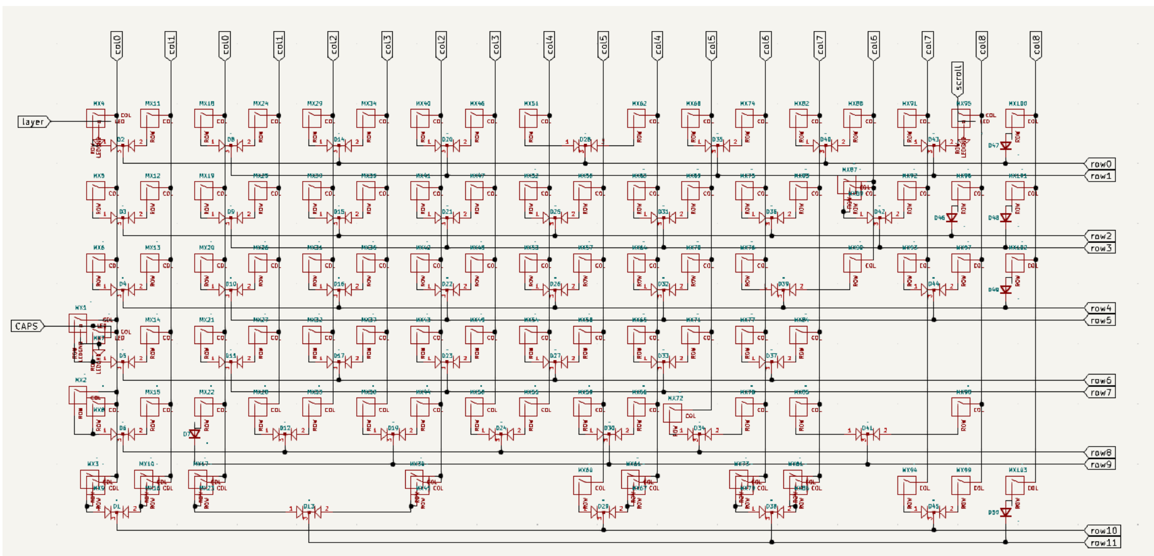

Figure 7. Keyboard matrix in KiCad. Columns and rows marked with tags.

It was required to have a diode after each switch to prevent ghosting. It would be inefficient use of space to use separate diode of every individual switch. This is why BAV70 diodes were chosen. They include two diodes in compact SOT-23 package, which reduces the number of components on the board and makes routing traces easier.

3.2.4 Microcontroller

The microcontroller or MCU was Atmel Atmega32u4 with a package of 44-VQFN. The MCU chosen is extensively used in keyboard projects and is supported by QMK. The number of IO pins was enough for rows and columns. Because it is common, it is widely available and cheap to purchase.

3.3 Component manufacturing

3.3.1 *Case and jig CAM*

All mechanical parts were designed using Siemens NX. CAM was done using NX CAM feature. Case had an angle in the bottom which meant that the piece had to be turned perpendicular to the spindle during machining. The top and the sides needed to be machined in one go.
To do that, a jig piece was designed. The jig was designed to hold the case from the screw holes and line it perpendicular with the machine z-axis. The mill turned b-axis to the right angle and the case bottom and sides was machined.

3.3.2 *Case machining*

Case was machined using HAAS UMC750 milling center. The machine has a capability of milling in five degrees of freedom, but only 3 was needed. Forth DOF made cutting the angles easier. The milling process can be seen in figure 8. Since the case was to be machined from one piece, the billet size had to be greater than the case. A billet with size of 375 x 146 x 30 mm was used.

Figure 8. Machining process of the keyboard case.

3.3.3 *Plate laser cutting*

The plate design was verified with a 3d-printed model. It had to be in two parts because of 3d-printer space limitations. Test plate fit after some filing, and switches aligned well with the PCB. Next the plate was cut out of aluminum using laser cutting. Laser cutter used was Primapower Platino Fiber 1530.

3.4 Programming

Flowchart representation of what keyboard software functioning is doing can be seen in figure 9. The process was mostly handled by QMK, but some parameter reconfiguration was needed, e.g. key mapping.

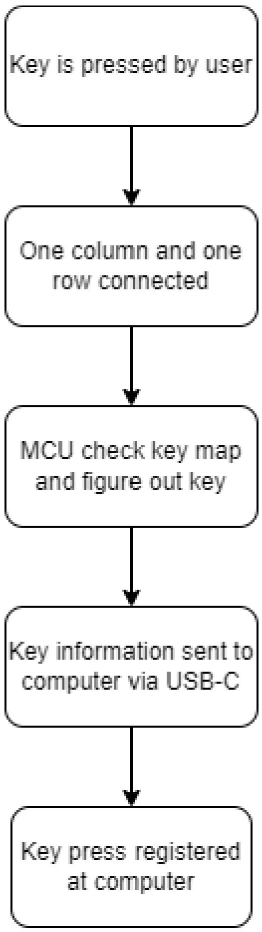

Figure 9. Flowchart of keyboard software functioning.

3.4.1 Arduino ISP

The microcontroller didn't have a bootloader preinstalled. Bootloader is a program that allows the loading of other programs to the microcontroller via more convenient interface like USB. The keyboard was designed with ISP communication protocol pins on the board, making it more convenient to connect the keyboard to an Arduino UNO and flash it with a bootloader. After the microcontroller had the bootloader installed, it was ready to be flashed with the keyboard code.

3.4.2 QMK MSYS

First the keyboard code had to be compiled using QMK firmware. Because of the changes in matrix layout, the matrix needed to be configured again. This happened by modifying the keyboard config.h and h88.h files. In config.h file the right rows and columns were defined for right pins in microcontroller. In h88.h file the correct keyboard layout was modified to match with the keyboard’s rows and columns. After modifications were done, the code was compiled as .hex file.

3.4.3 QMK Toolbox

QMK toolbox was used to flash the microcontroller. Before the microcontroller had any code written on it, it was already in programming mode. After the first flash the programming mode was toggled by grounding the reset pin twice in a row. This was made easy by implementing a reset switch on the keyboard PCB. QMK Toolbox detected the microcontroller, and the code was flashed to the keyboard. After this the keyboard appeared in windows devices as a device named h88. Key presses were tested by using tweezers and connecting rows and columns together one at a time. The program for monitoring the key presses was Switch Hitter.

3.4.4 Mapping keys

In QMK configurator the keys can be configured to whatever desired. In this project there was no need to configure anything else than the extra function key F13. This key was assigned as a sleep button: When pressing the sleep button, it will put the computer to sleep mode.

4. Results and discussion

4.1 Project results

Two versions were done of the keyboard PCB. Case has imperfections from machining, but they don’t affect the usability. Switch stabilizers can move too much in the plate, which causes rattling. All of the defined requirements were met. Some improvements still have to be done, but the final product was a functional keyboard that can be seen in figures 10 and 11.

Figure 10. Ready keyboard without keycaps.

Figure 11. Ready keyboard.

4.2 Cost

The budget was set around the price of Varmilo VA88M, which was 120 euros at the time of the project. The total price for one assembly of the keyboard can be seen in table 2.

Table 2. Total price for the project.

Component	Price paid (€)
Case	0,00 €
Plate	0,00 €
Switches	23,00 €
Keycaps	21,20 €
PCB	20,42 €
Components	16,61 €
USB-C cable	6,30 €
Total:	87,53 €

Case and plate didn't cost anything since it was possible to utilize available aluminum and machinery for free. The price quotation from Hubs.com for the case machined would've been 254 € which was completely out of budget. In addition, plate would've costed 20,85 € on Xometry.com, bringing the total price to 362,38 €**.** One possible way to stay under budget if machining is not an option would be to 3d-print the case and add dense material as weight to achieve the wanted properties.

The keyboard PCB needed to be ordered twice because of the matrix mistake in version one. This added 20 euros to the total price, which wasn't considered in the total price.

Price for one assembly of the keyboard stayed inside the budget limit, but ordering everything would've not been a viable option.

4.3 Occurred problems

4.3.1 Ver1 matrix problems

I didn't have experience in keyboard PCB design, so I searched online for open-source projects. One close to my requirements were found. The keyboard layout was what I wanted, the board looked good, and the project well made, so I decided to order the board from JLCPCB. While testing the board, it turned out that the switch matrix was completely wrong. Every second key was the same as the key before. Because of this I needed to redo the matrix with correct connections.

4.3.2 Switch stabilizers not compatible with first plate design

The holes for the switch stabilizers were too big. They stayed in place but rattle a lot while pressing keys, which makes the keyboard feel cheap. The plate needs to be redesigned with smaller holes for the stabilizers.

4.3.3 Ver2 matrix problem

There were few mistakes in PCB version 2. Two buttons were still tied to the same row and column, which made them register as the same button. This was fixed by cutting the column trace on the PCB and rerouting it to the correct column using a jumper wire.

4.3.4 MCU package difficulties

MCU had a package of 44-VQFN, which made it difficult to solder using soldering iron. This is why it was required to learn hot air soldering. If hot air would have not been an option, it would have been impossible to assemble the MCU. The package would need to be changed to 44-TQFP to make the soldering viable using only soldering iron.

4.3.5 Case imperfections

Only one version of the case was ever produced. Since it was the first time machining the case, imperfections were almost impossible to completely eliminate. The case came out as functional and important tolerances were met, but some machining mistakes can be seen.

5. References

[1] Drop, Accessed 12.08.2023, <https://drop.com/buy/drop-invyr-holy-panda-mechanical-switches?defaultSelectionIds=977750>.
[2] Cherry MX, Accessed 20.08.2023, <https://cdn.sparkfun.com/datasheets/Components/Switches/MX%20Series.pdf>.