# Gergely **Csaba**


gergelycsaba37@yahoo.com

+40755621535

Romania, Târgu Mures


## LANGUAGES

Hugarian ●●●●●

English ●●●●○

Romanian ●●●●○

German ●●○○○

## Education

**High School** (2012-2016)
"Márton Áron" National College, Miercurea Ciuc
In high school, I studied computer science and mathematics.

**University** (2016-2020)
Politehnica University of Timişoara,
Electronics, Telecommunications and Information Technology
Learned about digital, analog and power electronics.
Programmed various boards and microcontrollers such as MSP430, Raspberry Pi, multiple types of PIC-s, and FPGA.
Designed, populated, programmed, and tested multiple PCBs.

## Employment

**Student (2018-2020)**
Symph Electronics, Timișoara
During university, I worked for Symph Electronics for 2 years. The company was engaged in designing, assembling PCBs, and programming the final product. Given the company's size, I primarily focused on PCB design and programming. The programming tasks were carried out using C.

**Automation Engeneer (2020.09-2023.01)**
Engmatec Gmbh- Engmatec S.R.L, Radolfzell-Miercurea Ciuc
I was doing commissioning and programming industrial production line controllers (PLC) for companys such Continental, Bosh, Hella, BHTC in five different languages, C was one of them.

**Automation Engeneer (2023.02-2023.12)**
Controlsoft-Automatika Kft.- Controlsoft Automatizare SRL., Veszprém-Miercurea Ciuc
I programmed PLCs for water utility systems and successfully commissioned them.

**Project Manager (2024.01-Present)**
Controlsoft Automatizare SRL., Miercurea Ciuc
I manage projects from the bidding process to the final deliverable.

## Main projects as an Automation Engineer

**1. End of line-Toyota front camera tester**
The machine was assembled, programmed and tested in Germany. It includes an infeed conveyor for the product. Once the product is loaded into the machine, it is picked up by a Stäubli robot and placed in one of the 4 test chambers. Inside the chamber, the camera is tested. If the product passes the test, it is placed by the robot on a round table where a barcode is laser-etched onto the product case and optically checked to see if the connector pins are bent. Finally, the robot picks up the product and places it on an output conveyor.
My tasks in this project were:

- Write a part of the PLC code
- Troubleshoot possible electrical problems
- Correct the final positions of the robot by reteaching it

**2. Porsche production line**
This line is designed to assemble 9 different products for Porsche from scratch. The product is a 4-zone air heating/cooling, seat heating controller with lots of fancy features. The production line is quite big, 17m*8m. The machine starts with the main cell, where 3 Stäubli robots work together with a high precision feedback vision camera. This is where the front part of the product, which contains touch buttons, is assembled with great precision. The partially assembled product then goes out from the main cell. The operators insert two PCBs on the product then connect them to the display and to the buttons. Multiple vision sensor verifying weather the PCBs are correctly installed. The process continues to

another robot to place additional parts and again check with a vision sensor. Finally, an automatic greasing and screwing station follows and a vision sensor checks the final product.
My tasks in this project were:

- Write a part of the PLC code
- Program the 11 vision sensors
- Teach the 4 robots final positions
- Debug possible electrical problems

**3. Water and wastewater utilities**

The project involved modernising the automation/control systems of more than 400 water and wastewater utilities in Satu Mare. At the end of the project, the status of all water and wastewater utilities could be monitored from the dispatching centre. This system used PLCs for control and data collection, and the company's own proprietary software, WebScada, to display the data in the dispatch centre.
My tasks in this project were:

- PLC programming
- Commissioning

## Main Python Projects

### 1. AI-Powered Classification of Social Media Memes

The goal of the project is to filter out negative, offensive, or anger-inducing memes that appear on social media. The model is trained on a dataset containing 4,000 images — 2,000 labeled as positive and 2,000 as negative.
The system is built using three main components:

1.EasyOCR
2.BLIP
3.ROBERTa

EasyOCR extracts the text from all 4,000 images. In parallel, BLIP generates a visual description for each image. After this, both the extracted text (from EasyOCR) and the generated captions (from BLIP) are passed as input to the RO-BERTa language model.

ROBERTa then determines the category (positive or negative) and gives a confidence score. This combined EasyOCR + BLIP + RO-BERTa architecture achieves very high accuracy — up to 97–98%.

This high performance is mainly due to the fact that RO-BERTa, as a language model, brings background knowledge. It understands the meaning of the extracted text and is able to connect it with the visual description of the image.

### 2. AI Flappy Bird

Initially, I programmed a classic Flappy Bird game from scratch, implementing the essential mechanics that allowed the bird to jump and maneuver towards the pipes in response to keyboard inputs. Just like in the classic Flappy Bird game, scoring occurred when the bird successfully passed through the gap between pipes without colliding with any obstacles. Once this foundational aspect was perfected, I advanced to the next level by integrating AI into the game. Utilizing the NEAT (NeuroEvolution of Augmenting Topologies) algorithm, I enhanced the existing codebase to create a bird capable of learning and mastering the game, ensuring it never fails. This involved significant development efforts to ensure the AI receives accurate information and makes informed decisions while playing.
During this project I learned about:

- AI in general
- NEAT algorithm
- OOP

### 3. Multiplayer (server-client) Rock-Paper-Scissor Game

It is a classic rock-paper-scissors game where two players can play against each other. At first, it seems like a silly game, but I encountered the most headaches so far during its development. After I created the single-player version of the program, I had to address the following questions: How will it run with two (or even more) clients simultaneously? How do I handle it if only one player quits the game? How do I keep track of which client has made a move? What will happen if the communication stops?
In the end, I succeeded in answering these questions and created a fully functional multiplayer rock-paper-scissors game.
During this project I learned about:

- Threading
- Server-client relation and how to create them
- Network communication
- OOP

**For more projects, please check out my portfolio website:**
https://gercsaba.github.io/Website/