# Title

Cristian Stéfan Talos

B.Eng., M.A.Sc., CPI

Systems Integration Designer, Robotics Programmer, and 3D printing enthusiast.

Learning new things is my greatest source of motivation!

# About Me

## Polytechnique Montréal

I have a bachelor's degree in Mechanical Engineering with a specialisation in Mechatronics, and am currently earning my Master of Applied Science.

In my first three years at Polytechnique, I was a member of the student group “Oronos.” As the Systems Integration Designer, I helped build a composite rocket for the Spaceport America Cup [year] in New Mexico.

In 2019, I was involved in the student group “Elikos,” where we built an autonomous drone with a robotic arm for [name of the competition] in Alma, Chicoutimi. My role was to make the 3 DOF robotic arm that would be mounted on the drone.

I learned a lot about electronic circuits and with that experience, I started to design PCBs for a few of my school projects.

My interest in programming and electronics led me to work on my thesis about the development of an active carrier for the braiding of composite textiles. A patent process is underway and a publication is expected in the summer of 2022.

## École nationale d'aérotechnique

I have a Diploma of College Studies in Aerospace Engineering. During this time, I learned a lot about CAD design and manufacturing, as well as circuits and programming with Arduino. This is also when my passion for 3D printing took off.

In 2014, I was involved in the student group “Cargo Plane,” where we had to build an RC aircraft for a competition in Florida.

# Projects

## 3 Axis CNC Router

### Description

The goal of this project was to familiarize myself with the Solidworks CAD software and to build a CNC router that would be useful for any projects requiring the machining of plywood or even aluminum sheets. I also wanted to acquire more knowledge on electronics for CNC machines.

This project was done in collaboration with two other engineering students whose focus was on the CAD. My role was to make the electronics box, the electronics circuit and to program the CNC shield.

We used v-slot aluminum extrusions for the structure and for the linear guide and eccentric nuts on the carriage of all axis to tighten the grip on the extrusions. Aluminum sheet metal was used to make most of the carriage structures, and some more complex parts were 3D printed in PLA.

All the electronic components (such as the limit switches) were wired through a drag chain and then plugged in the electronics box. The electronics box has a 24v power supply for the stepper motors and for the CNC shield and it has a 220v 600W transformer for the spindle. An Arduino microcontroller runs the GRBL firmware and connects via USB to a computer to the machining program.

### Specifications

* 600W Spindle (max 15000 RPM)
* ER11 Collets with HSS End Mill
* GRBL Controller for the stepper motors (Arduino shield)
* Working volume: 100cm x 100cm x 20cm
* Control via USB, using a Universal G-Code Sender Software
* Emergency Stop Button and other Safety Features

### What I learned

* Use Solidworks (part and assembly design)
* All the electronic wirings
* Integration of all the electronics
* How the GRBL controller firmware works
* Rubber v-wheels are not sturdy enough, and it causes vibrations when machining
* Cable management and connectors selection (wires tend to take a lot more space than expected)

## Hexacopter with autopilot

### Description

My participation as a member of Elikos at Polytechnique, and my interest in videos of autonomous drone assembly, sparked my interest in drones and motivated me to build my own.

I used an DJI 550 frame on which I designed and 3D printed fixtures for the various components (such as the battery and gimbal fixtures). I soldered all the connectors on the frame and on the power distribution board.

The flight controller is a Pixhawk system, which is connected to a 6-channel radio controller that has a range of up to 2km. The remote can also control the orientation of the gimbal mounted on the drone.

The Mission Planner software uses real-time telemetry to control the drone. A flight plan can be made with multiple setpoints that use GPS data and the altimeter readings. There are many ways to program the flight controller using Mission Planner, and it is even possible to plan topographic drone surveying. Then, in post-processing, it is possible to stitch the images together to generate a map.

### Specifications

* Pixhawk autopilot
* 2 Axis Gimbal for a GoPro Camera
* 5000mAh 4 Cells Li-Po battery
* Telemetry up to 2km
* High and wide landing gear for stability
* Mass of 2.5kg (with battery and camera)

### Next Steps

* Get the Transport Canada pilot license
* Configure the remote controller
* Configure the autopilot for the Pixhawk
* Validate that all the features are working (altimeter, GPS, telemetry, etc.)
* Cable management
* Design and print a cover for the electronics
* Incorporate a radio camera for live view

## Hot Wire CNC Foam Cutter

### Description

Ever since my involvement with the “Cargo Plane” project, I have had a growing fascination for aircrafts. I thought it would be really cool to be able to cut foam using a hot wire (nickel-chrome alloy) to make the airfoil of the plane. Since nothing open-sourced existed at the time, I decided to build my own CNC foam-cutting machine.

I made a first version (middle video) to try out the electronics for the project. I used an Arduino Mega compatible board called Ramps 1.4 which runs the Marlin firmware v1.1. At the time, I wanted to be able to make a variable airfoil, so I programmed a MATLAB script that would import an airfoil .csv file, and then adjust the scale and rotation of the profile. Finally, the script would compute the g-code of the motors to be able to generate the desired shape.

The nichrome wire was heated up using a 36v power supply and a lot of tweaking had to be made to adjust the speed of the cutting process (there is some sort of equilibrium between the heat dissipation of the wire by contact of the foam, and the feed rate of the wire). Ultimately, the project took too much space and it was disassembled.

### Specifications

* The working volume is 30cm x 20cm x 150cm
* The motor controller is Ramps 1.4 with Marlin v1.1 firmware
* The hot wire is always under tension and the voltage is adjustable
* A LCD screen displays various information

### What I Learned

* Gained design experience
* Learned how to use the Ramps control board
* Learned how to configure the Marlin firmware
* Gained programming experience

## E-paper Weather Display with ESP8266 and Raspberry Pi

### Description

Since 2015, I became more and more familiar with programming microcontrollers from the Arduino family (AVR processors) and I always kept reading things about the micro-computer Raspberry Pi. I decided that I wanted to learn Python and how to program on the Raspberry Pi.

I searched for Python libraries to control an e-paper display and then I looked for different ways of getting temperature data. I decided to use the OpenWeather API to get meteorological data for Montreal and I used the BM280 temperature sensor connected to the Raspberry Pi.

The Raspberry Pi uses multithreading to run a MQTT server, to get data from other MQTT clients and to display the data on the e-paper. There are multiple ESP8266 microcontrollers in the various rooms of my home that are connected to their respective BM280 sensors, and they send their temperature via MQTT to the Raspberry Pi.

The Raspberry Pi then stores the data from the MQTT clients and from the OpenWeather API inside global variables and the e-paper display thread accesses them to update the display every 15 minutes. Since I use a Raspberry Pi Zero W, the screen refresh sequence is quite slow, but the whole system is very power efficient.

### Specifications

* Raspberry Pi Zero W acts as the MQTT host
* ESP8266 microcontrollers act as MQTT clients
* BM280 temperature sensors connected to the ESP8266 clients and to the Raspberry Pi, sending data via I2C
* OpenWeather API used to get meteo data for the region
* The display updates the info every 15 minutes

### What I learned

* Use multithreading on Python
* Use API's from the Web
* How to connect devices together using MQTT
* Familiarize with the ESP8266 microcontrollers
* Programming knowledge in C++ and in Python

## CNC Pen Plotter

### Description

I was interested in redesigning and improving the functionality and accuracy of an open-sourced pen plotter CNC that I found on the internet. The electronics and the controls were not included in the project tutorial, so I had to familiarize myself with the many existing CNC controllers and pick one that would suite the application. Since I am very familiar with Arduino, I picked a GRBL controller shield for an Arduino Uno.

I was able to learn how to use the Marlin firmware with the GRBL control board and how to vectorize images to be able to generate g-code after.

### Specifications

* Nema 17 stepper motors used for the three axis
* GRBL shield with Arduino used to control the motors
* Core-XY design for the X and Y motions
* Images vectorized using Inkscape
* CAM parameters and g-code generation using the Inkscape extension GCodeTools

### What I learned

* Design experience
* Motor controls experience
* How to vectorize images
* Experience regarding CNC programming
* Familiarize with Marlin firmware and GRBL controller