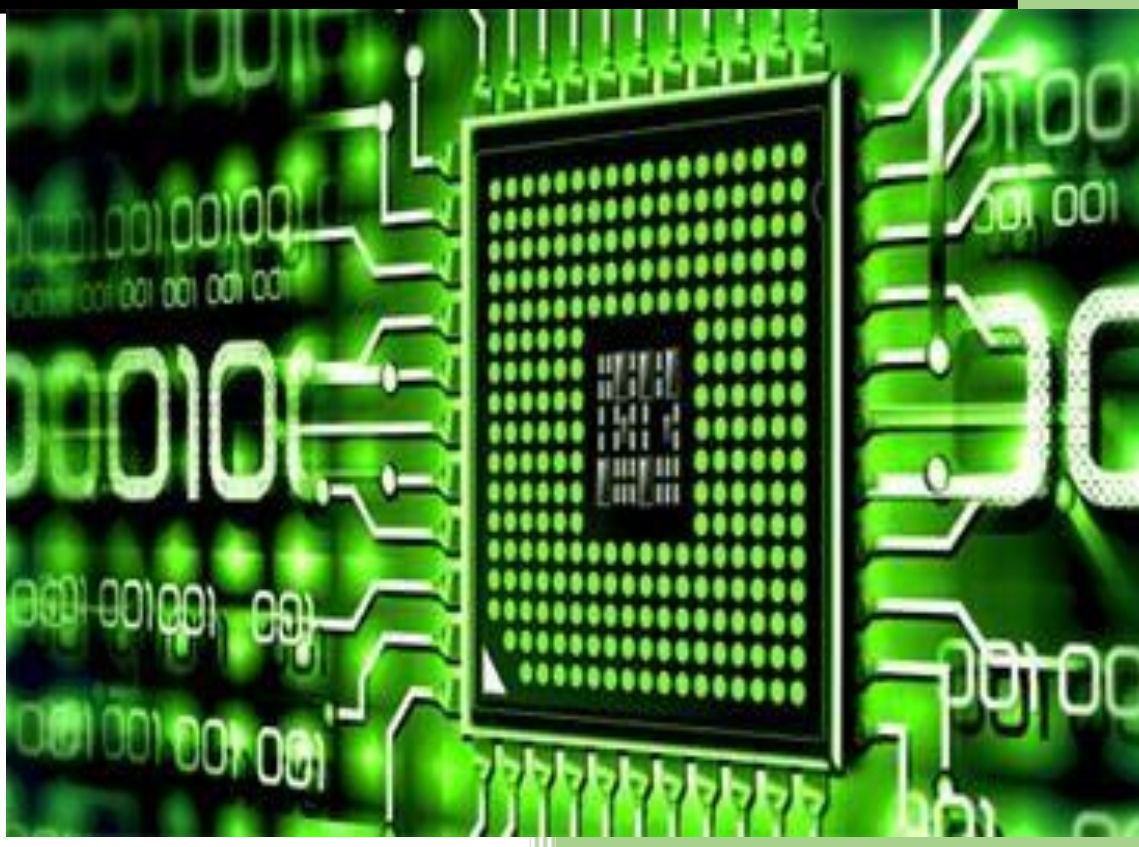
2022

CNC Control Software



Catalin Goga

Technical University of Cluj-Napoca

10/01/2022

Contents

1. Introduction

1.1 Context

What is a CNC?

A CNC is a *Computer Numerical Control* device, a machine for moving precisely in one or more directions. A CNC machine needs (at least) one motor for each axis (direction of movement). All the motors are controlled by an electronic circuit. The motors and the circuit use a power supply to create the motion you want. The circuit may also need sensors to stay calibrated and do the job right. The circuit also needs to follow your programmed instructions. Put another way, your circuit needs to support your software, your motors, your power supply, and your sensors. (Marginally Clever, 2015)

The goal of this project is to design and implement firmware for controlling a CNC, the software that sends *gcode* commands to the CNC and the actual CNC machine.

The software is intended to be used for a hobby CNC router and it can later be improved so that it can be used in more advanced projects.

1.2 Specifications

The CNC will be built using an Arduino Uno development board and two CD-ROM readers. The Arduino will be the controller for the stepper motors which will move on the two-dimensional axes and the servo motor which will lift and lower the instrument for plotting.

The PC software should be able to perform the following actions:

- Read a file line by line
- Let the user choose the file to be processed
- Communicate with the Arduino trough the USB serial bus
- Let the user control the CNC in real time by specific keyboard keys
- Let the user modify communication parameters

The firmware on the CNC side should be able to perform the following things:

- Interpret the G commands received on the serial bus
- Control the stepper motors of the CNC
- Implement and algorithm for computing the movement trajectory from the current position to the next position
- Track the current position of the pen head

1.3 Objectives

Design and implement software which takes in *gcode* commands and translates them into movements executed by the CNC, implement the firmware controlling the Arduino, build the CNC and make improvements to the current design.

2. Bibliographic study

G-code is the most widely used **computer numerical control (CNC)** programming language. It is used mainly in computer-aided manufacturing to control automated machine tools and has many variants.

G-code instructions are provided to a machine controller (industrial computer) that tells the motors where to move, how fast to move, and what path to follow. G-codes, also called preparatory codes, are any word in a CNC program that begins with the letter G. Generally, it is a code telling the machine tool what type of action to perform, such as:

- Rapid movement (transport the tool as quickly as possible in between cuts)
- Controlled feed in a straight line or arc
- Series of controlled feed movements that would result in a hole being bored, a workpiece cut (routed) to a specific dimension, or a profile (contour) shape added to the edge of a workpiece
- Set tool information such as offset
- Switch coordinate systems

(Wikipedia, 2022)

I found a video on YouTube [1] which inspired me, and I want to reproduce the CNC plotter from there. It uses two DVD-ROM stepper motors for moving on the two axes and one servo motor for controlling the writing instrument.

I also found a website [5], [6] describing the process of building the machine and software which should be uploaded on the Arduino. It helped me understand the overview of the project and the steps I must take to build this project. There are two articles on the webpage, one describing the first version of the CNC plotter, and the second one describing some of the improvements that can be made to the machine to be able to plot some circle arcs as well.

For the drawing part, I used the Bresenham's Line Drawing Algorithm which can be found on Wikipedia and the Arduino Forum.

3. Analysis

The project will be divided into two parts: the computer side software, which will be the interface between the CNC and the user, and the actual CNC machine consisting of the hardware and the software for controlling the hardware and processing the commands coming from the computer.

3.1 Computer Software

The software on the computer side should provide the user with a way of interacting with the CNC machine. The software will take as input a file in *gcode* format and send it line by line to the trough the serial bus to the CNC. The application will provide the user with a way of selecting the file to be sent and a way to select the COM port on which the CNC will listen to. The application will listen to an acknowledgement message from the CNC before sending the next line. It should also be able to provide the user with a way of controlling the CNC machine in real time for testing purposes.

3.2 CNC

The firmware should translate the commands received on the COM port in *gcode* format to linear movement on the three axes of the 3D plane. The movements of the X-axis and Y-axis are performed by the stepper motors which move a carriage. The movement on the Z-axis is performed by the servo motor which lifts and lowers the pen for writing on the canvas.

The motors stepper will move two carriages: the X-axis carriage on which the Z-axis mechanism is mounted, and the Y-axis mechanism, on which the canvas support is mounted.

The motors will be controlled by the Arduino using an L293D motor shield [9] with an external power supply (pc power source). The stepper motors will be powered by 12V, and the servo motor will be powered by 5V.

4. Design

4.1 Software Design

Firmware

The firmware will be the software responsible for reading a command and translating it so that the head of the pen is positioned on paper in the point encoded in the command. To do that, an algorithm must be defined for reading each command received from the serial bus.

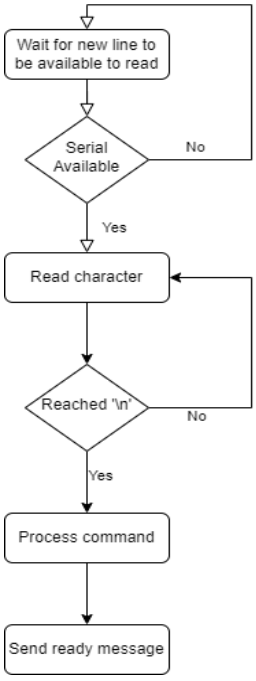


Figure 1: Command Process Diagram

After reading a line, it must be decoded, and proper action must be taken. This firmware will implement basic G commands for drawing straight lines, circles and moving all the three axes independently. The commands which can be interpreted by the firmware are listed below:

- G00 – Go to the home position (0,0)
- G01 – Linear interpolation (move to a certain point)
- G02 – Circular interpolation clockwise (move on a circle arc)
- G03 – Circular interpolation counterclockwise
- G04 – Wait for n milliseconds
- I – Defines the arc center on X-axis for G02 and G03 commands
- J – Defines the arc center on Y-axis for G02 and G03 commands

PC Software

The software on the computer side will be responsible for opening the input file, reading it, and sending each line on the serial bus to the CNC. It will wait until an acknowledgement message is received, then send the next line.

At the start of the program, the user should choose the COM port on which the CNC will listen. Then, the user should choose the input fille which will be processed. Finally, the user will press the start button and the transmission of the files content will begin.

The program will be a console application in the beginning but can be improved by adding a GUI. The program will listen for input keys from the keyboard and will perform actions based on the input of the user. The key encodings will be displayed to the user.

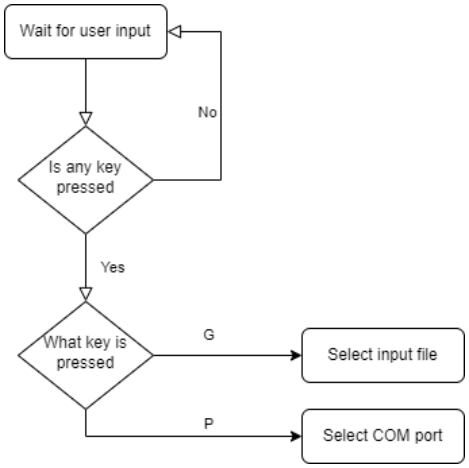


Figure 2: Computer Software Flow

4.2 Hardware Design

The hardware representing the CNC will be composed of an Arduino Uno, two DVD-ROM readers, one servo motor, one pen and a PC power supply.

The cases of the two DVR-ROM readers will be positioned perpendicular on each other and on each case, motor mechanism will be fixed. On the carriage on the bottom side will be glued a square plywood which will be the support for the paper. The carriage on top will be holding the mechanism for lowering and raising the pen.

The mechanisms which hold the stepper motors, and the carriages will be fixed on the metal cases with x mm screws.

The stepper motors are bipolar motors, meaning that they have two windings, thus 4 wires. The wires will be connected to the motor shield through the terminals M1 to M4, which will control their movement. The motor shield will be connected to the external supply to 12V.

The servo motor will be fixed on the pen mechanism and its data pin will also be connected to the motor shield, but the GND and VCC pins will be connected to the external power supply to 5V.

The Arduino board will be connected to the PC through a USB cable which will serve as power supply and UART communication bus.

Figure 3: CNC Hardware

5. Implementation

The firmware is implemented in C++ in Arduino IDE.

The PC software is implemented in Java in Processing IDE.

Below is a screenshot of the user interface of the current version of the software on the PC side:

Figure 4: PC Software UI

6. Testing and validation

Testing the whole system was simple; I drew some text in Inkscape and exported it in *gcode* format. Then I loaded the file in the PC program and the CNC started to draw the text previously created.

To make sure that the gcode was generated correctly, I used CAMotics software for visualizing the 3D model, the head trajectory and all the parameters which the CNC will need to properly execute the requested job.

Through testing, I came up with some limits for the servo motor doing the movement of the pen: 90 degrees for the UP position and 49 degrees for the DOWN position. For each millimeter, the stepper motors on the X and Y axes do 100 steps. The maximum position for both stepper motors is 40mm. The speed of the steppers is 600 revolutions per minute.

7. Conclusions

The design is not very complex and there is a lot of room left for improvement. There are many competing software programs which provide much more complex functionality like GRBL, Marlin, Klipper, Maslow, CAMotics and so on. More functionalities could be added like:

- More gcode commands to be supported
- Obeying all the gcode standards
- A module of the PC software to preview the job before sending it to the CNC
- Testing routines
- Limit switches for the X and Y axes
- A bigger drawing canvas

- A more robust skeleton for the whole machine
- Possibility of selecting between absolute and relative positioning modes
- More reliable motors could be used

8. Bibliography

[1] https://www.youtube.com/watch?v=QT3dAxqICC8
[2] https://en.wikipedia.org/wiki/G-code
[3] https://www.youtube.com/watch?v=CsNLeWpOZ_k
[4] https://github.com/naveedamir484/GcodeGen
[5] https://www.marginallyclever.com/2013/08/how-to-build-an-2-axis-arduino-cnc-gcode-interpreter/
[6] https://www.marginallyclever.com/2014/03/how-to-improve-the-2-axis-cnc-gcode-interpreter-to-understand-arcs/
[7] https://en.wikipedia.org/wiki/Bresenham%27s_line_algorithm#:~:text=Bresenham's%20line%20algorithm%20is%20a,straight%20line%20between%20two%20points.
[8] https://forum.arduino.cc/t/bresenhams-line-algorithm-in-arduino-cnc-plotters/65509
[9] https://cdn-learn.adafruit.com/downloads/pdf/adafruit-motor-shield.pdf
[10] https://www.marginallyclever.com/2014/03/how-to-improve-the-2-axis-cnc-gcode-interpreter-to-understand-arcs/