Wire Cutter and Stripper

Cassiers Antoine
Electronics and telecommunications
Université Libre de Bruxelles
Bruxelles, Belgium
antoine.cassiers@ulb.be

***Abstract*—With the rise of automation in nowadays world, lots of handmade machine start to appear to get rid of repetitive tasks such as wire cutting and stripping a high number of cables. New technologies such as 3D printing allow anyone to build his own machine and numerous design ideas start to flow. In this vein, this project aim to depict the whole process of design and construction of an automated wire stripping machine going from design idea, component selection, code implementation and issues encountered to utilisation and conclusion. Main focuses of the prototype creation are it's ease of construction, price and duplication possibility. A final working product was achieved while possible amelioration are also discussed in the end of the paper. All 3D models, used material and codes are open source and can be found on XI.**

***Index Terms*—Electronic, DIY, Wire Cutter, Stripper**

I. Introduction

The purpose of this project is to design and construct a wire cutting and stripping machine from scratch. This machine should be able to cut and strip a number N of cables of length L given by the user. The main parts of the report will be about the idea of the prototype, a brief study of the possible components and the reasons behind the final choices, a discussion of the implemented code, some difficulties that were faced, how to use the prototype and how to easily duplicate it and finally the improvement that are possible to make.

II. Prototype ideas

Many angles of attack are possible in this project. The main idea focused to create a prototype was it's easiness of construction and duplicability. The goal was to take inspiration from already commonly used machines. The product can be separated on two main parts. The first one aiming on feeding the cable of the needed length in the second part which either cut or strip the cable. For the first part, main inspirations came from cable 3D printer. Indeed, these kind of machines also have to 'feed' a plastique cable of a very precise length. The extruder idea has thus been kept for the cable guiding process. However, two types of extrusion are still possible. As shown on figure 1 bowden or direct feed extrusion (respectively pushing or pulling the cable to the needed place) can be done. Bowden extrusion was chosed for this project as at the end of the guide, the cable should be able to move right or left in the stripper. A guide will be used to be sure the cable is not bending in the process and to allow a servomotor to perform the placement in the stripper. The second part consist in stripping or cutting the cable. Inspiration was taken from an already existing model found on internet [1]. A pulley-like system use a nylon cable to tighten more or less the stripper when needed. Advantage with this system is that same motors can be used for both part of the prototype making it easier to power with a single supply. A first prototype idea can be found on figure 2. Component used should be chosen wisely depending on the requirements (power consumption, torque needed, I/O,...). Next section will discuss about these requirement and the component choices.

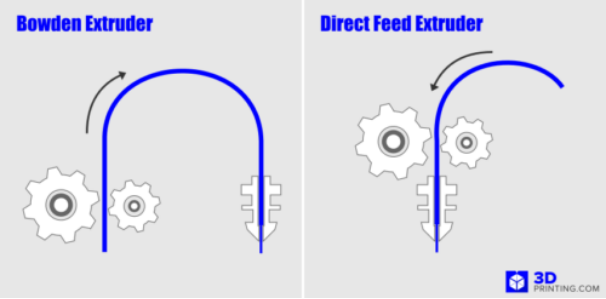


Fig. 1: Different types of feedind [2]

III. Component choices

A. Motors

Motors are one of the main part of the prototype. careful choice can drastically simplify the product creation, duplicability and robustness. Nowadays, several motor types exist. A first separation can be made between AC and DC powered motors. In this project, DC motors will be used as they are much easier to power and as the rest of the prototype will also be DC powered. Lots of different DC motors still exist but two mains type seems to stand out : Brushless DC

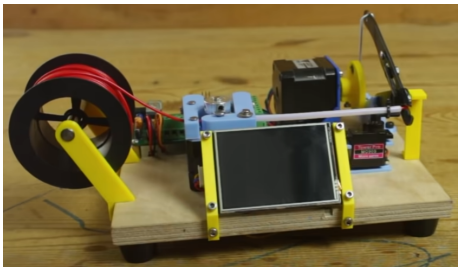


Fig. 2: Prototype example from [1]

and Steppers. Theses two types of motor are both fairly easy to use but steppers seems more qualified for the job. Indeed, brushless motors are more fitted for fast speed, low torque applications while stepper have a lot more torque and rotation precision which will be needed for moving and cutting the cable. The main disadvantages of the steppers are firstly that they require 4 inputs instead of 2 for a brushless and that they are a little bit harder to control. As explained earlier, both used motors will be the same to simplify the prototype. Finally, two NEMA17 type stepper motor are chosed[1](visible on figure 3)-. Theses components are powered with around 8 to 10v which is important to take into account in the next component choices simplify the prototype powering. Another motor is needed to guide the cable in the stripping or cutting part of the stripper. This time a servo motor is chosen as it allow even more precise angle control and it can be powered directly by a microcontroller such as an arduino. As said earlier, drivers are needed to control stepper. The next section will discuss about these driver choice.

Fig. 3: Nema 17 stepper motor used [3]

B. Drivers

As said earlier, drivers are needed to efficiently use the stepper motors. Their role is to transmit pulses in the right order to the motor to allow them to properly turn. Lots of models exist in the field of drivers, first test were made using A4988. Theses drivers works fine at low tension and low current but some current problems were encountered during the tests. To ensure safety and functionality of the prototype along the time, other drivers are investigated. After some researches, L298N are used as they allow more current. Downside of using these drivers is that they require more inputs of the microcontroller (from 2 to 4). After some test, the NEMA-L298N combination seems to work well and theses components are thus kept for the final product.

C. Microcontroller

The microcontroller is the brain of the prototype. As for the motors and drivers, lot of possibilities come in mind when choosing a microcontroller (dsPIC, Arduino or even raspberry). The main characteristics that needs to be checked

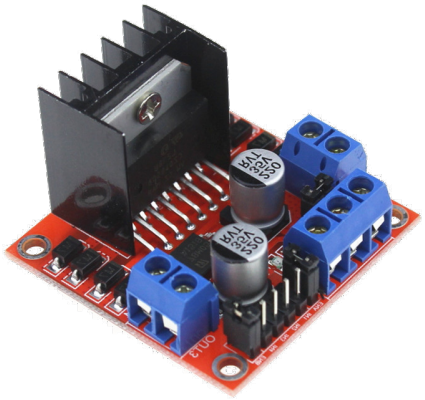


Fig. 4: L298N driver

are the number of I/O, the power source needed, the capabilities of the controller it's size and ease of utilisation. At first glance, Arduino seems the best choice as they are easy to code using the built in IDE (in C++) and several libraries and documentation already exist for controlling servo or stepper. Arduinos can also be powered with the same tension as the motors and drivers (around 8 to 12V). Several sizes of arduino still exist going from a broad range of I/O, size and prices. An Arduino Nano is chosen as not that much I/O are required for the project : 2x 4 outputs for the drivers and 1 output for the servo motor[2].

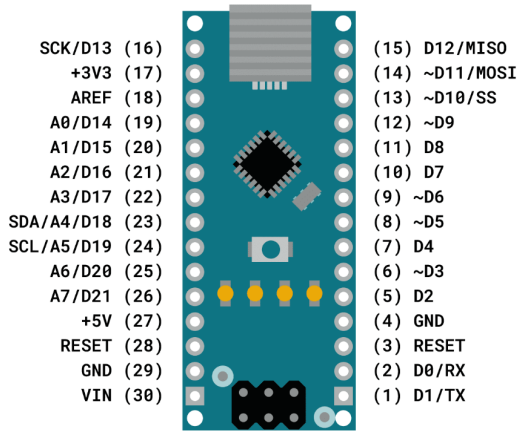


Fig. 5: Arduino nano I/O specification [4]

D. Stripper

A wise selection should also be made when chosing the stripper depending on the design idea. In this case, the electric cable is fed perpendicularly to the stripper. Furthermore, the stripper is tighten by a pulley-like system and a nylon cable. A stripper as light n and easy to close as possible is thus

[1]wantai 42byghw811 models

[2]arduino uno have 13 digital I/O some room is left to add new components

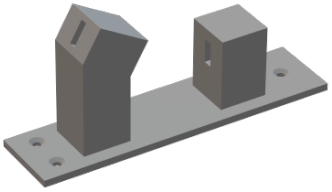
searched to allow the stepper motor to have enough torque to cut the cable without consuming too much current.

E. Power supply

As said earlier, all components were chosen to be able to use a single power supply for the whole prototype. Every components can be powered around 9v. The current drained by the whole prototype still needs to be known to chose the right supply. Power test were performed with a laboratory power supply displaying the drained current in real time. The maximum drained current occur when cutting the cable (as torque depend on the current for the stepper motor and most of the torque is needed in the cutting part) and turn around 2A. A 9V 2A is thus chosen to ensure a sufficient current supply without overloading the drivers with too much current.

F. Other pieces

Some non electronics parts are also needed for example to support the stripper, the different used motors and drivers. 3D pieces has been designed using a 3D modelling software to support all the components. First test were made with low infill values to confirm the usability of the product and then printed with much more infill to ensure enough resistance as certain pieces such as the stripper support will take a lot of traction. Finally, a teflon tube is used as a guide for the cable between the feeding and stripping/cutting part. Some designed 3D parts are shown in the following figures :



(a) Stripper support

(b) Cutting stepper support

Fig. 6: Example of designed pieces

IV. Code explanation

The whole code as been written using the arduino IDE in C++ and following the arduino canvas. Whole code commented code is available on appendix. At the code startup, `Stepper.h` and `Servo.h` libraries are imported and the 3 motor object are instantiated on their respective pins. At the arduino launching, the `void setup()` routine is runned once. This routine set the different pin mode (input or output) using `pinMode()`, the stepper speed and servo pinout are also set. `void loop()` routine is then runned infinitely till `exit(0)` is called. There number and length of the cable needed are firstly selected. As the stepper library work in steps and not in length, a small function converting length to steps is created. As a step correspond to 1.8 angle, using the feeding wheel radius the corresponding length can be computed following $L = 2 \cdot \pi r$ number of step needed are then given by $N = \frac{200}{L}$. The cutting and stripping loop can then start and consist in :

1) feedforward the cable of approximately 1cm.
2) move the cable in the stripping part thanks to the servo.
3) perform the stripping by moving the cutting stepper motor forward and backward[3]
4) move the cable of the final length minus approximately 1cm.
5) perform the same strip as in step 3.
6) move the cable of the last centimeter.
7) move the cable to the cutting part thanks to the servo.
8) cut the cable.

all these steps are ran N+1 times to ensure a first control cable. To simplify the code, two other function have been made to handle respectively the whole cutting and feeding process. Theses function are simply called when one motor needs to be moved.

V. Issues encountered

Different issues were faced during the prototype creation. The first one was already partially discussed and concern the too high current drawn by the cutting motor. As said earlier, arduino IDE was used to create the code by the help of some libraries such as `stepper.h`. This library allow to just tell the number of steps that needs to be performed instead of sending the pulses directly to the driver. One major problem of this library lie in the idle motor state. Indeed when not in use, pinout of the motor are still set to high and thus drawing current. This is problematic for several reason as it will firstly not allow the driver to cool down damaging it prematurely and it will still draw current of the power supply for doing nothing. To solve this problem, all the output are set to low manually using `digitalWrite(Pin1, LOW)` for each 4 pins of the motor as following :

```
digitalWrite(3, LOW);
digitalWrite(4, LOW);
digitalWrite(5, LOW);
digitalWrite(6, LOW);
```

Codes/Idle.ino

Another faced problem consist in the length to step function translation. Indeed, radius of the feeding wheel is pretty small and step needs to be integers. The real translation performed

[3]the backward number of step is smaller than the forward one as stripper spring already move the motor a bit backward when in idle state.

is thus a rounding of the equation seen in previous section leading to a small error in the actual length of the cables.

A third faced issue came from the grounding. in fact, two ground were used in the first iteration of the prototype. The first one for all the component except the servo while this servo was grounded to the GND arduino pin supposedly at the same tension. After some test with a voltmeter, it seemed that the tension between the two grounds was not really at 0 potentially leading to ground loops. This problem was solved by grounding the whole prototype to the same voltage.

Finally, first nylon cable used to tighten the cable was a lot thinner and was breaking after a few cuts. A much larger cable was thus simply taken to solve the problem.

VI. How to use the prototype

The prototype is fairly straightforward to use. The cable coil is put in the 3D printed coil support and the cable can be inserted in the extruder part. A pc has to be connected to the arduino in order to setup the different parameters (length and number of cable needed can be changed in the attached code). Once the arduino code is televersed, the power plug can be set and the prototype will start his work for the given parameters. Each time the power supply is plugged in, machine will cut N cables of length L. A fine tune of the initial cutting stepper motor has to be done initially to ensure that the stripper will not cut the cable each time it is trying to simply strip it. Therefore, pulley needs to be set at 12 hour.

VII. Prototype duplication

The possibility to recreate easily the prototype took a major place in his design. Lot of recuperation has been made to make the early versions of the prototype such as stepper motors from a old 3 printer. All the used parts of the framework can either be found on amazon (all the links are available following the drive on the appendix) or 3D printed following a .STL file given in the attached zip file. To recreate the prototype, following step have to be done :

1) buy all the component following the list.
2) print all the components supports [4] and fix them on the plank using M3 screws and following the scheme on fig 7.
3) place the components on their respective support and cable them following scheme on fig 8.
4) Either solder all the ground and Vcc cables to a pcb or use a breadboard to power the prototype.
5) Drill a small hole in the stripper and fix the cable from the cranck to the stripper.
6) upload the code in appendix into the arduino and select you needed parameters.

[4]all the 3D models can be found on the attached zip file or in :

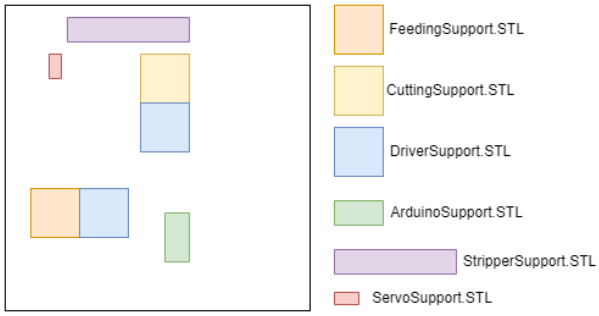


Fig. 7: 3D printed part placement on the wood board

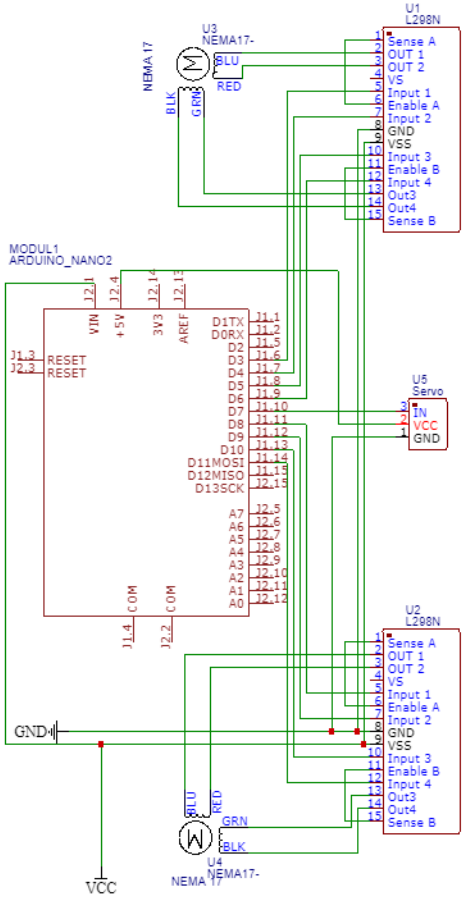


Fig. 8: Wiring scheme of the prototype

VIII. Possible amelioration

Knowing the limit of a prototype is mandatory to be able do perfect it. Different amelioration can be done to simplify the user interaction with the product and to ensure longer longevity. One first major amelioration that can be done is the implementation of a screen and buttons in the prototype allowing the user to select directly the number of cables needed and their length without any computer. LCD tactile screen can be added using only 2 inputs by implementing a I2C module between the screen and the arduino and coded using `Wire.h` and `LiquidCrystalI2C.h` libraries. Another

possible amelioration is to swap the cables for thicker ones. As lots of current is running in the cables, actual thin cables produce a lot of heat and can wear out fast.

IX. Conclusion

The goal of the project was to create a whole prototype from scratch requiring upstream work to first find design ideas and researches about possible components utilisation. Once all components gathered, some supports were needed for the components. 3D design and printing was used to obtain final parts. After some iterations, support and final components were placed on a wood plank following 7 and using M3 screws. Finally, after wiring everything, a 9V 2A power supply is used to power everything up and the prototype is ready to go. Some possible amelioration were also discussed for the future iteration of the product to ease user utilisation.

X. References

References

[1] Mr Innovative, "DIY Wire cutting and stripper Machine | Arduino project," Dec. 2020.

[2] "Advantages and Disadvantages of Direct and Bowden Extrusion," Nov. 2017.

[3] "Stepper Motor (nema 17)."

[4] "(3) Pinterest."

XI. Appendix

All used component, 3D models and commented arduino code are open source and can be found on : https://drive.google.com/drive/folders/1yKTraQngbUqXN2wD6IbRU49dOpOHUPjd?usp=sharing or in https://github.com/Gifted-kiddie/ProjectEIT-WireStripper