

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

تصميم وتنفيذ ماكينة CNC

بحث تكميلي لنيل درجة البكالوريوس مرتبة الشرف في الهندسة الكهربائية (تحكم)

إعداد الطلاب :

حسن جعفر حسن الامين

خالد يس حسن محمد

محمد عبدالرحمن الرحيمة الامين

إشراف :

د/ الامين كمال عبدالحى ابراهيم

قسم الهندسة الكهربائية

كلية الهندسة

جامعة الشيخ عبدالله البدرى



مارس 2022م

الآية

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

﴿وَاللَّهُ أَخْرَجَكُمْ مِنْ بُطُونِ أُمَّهَاتِكُمْ لَا تَعْلَمُونَ شَيْئًا وَجَعَلَ لَكُمُ السَّمْعَ
وَالْأَبْصَرَ وَالْأَفْئِدَةَ لَعَلَّكُمْ تَشْكُرُونَ﴾

صدق الله العظيم

سورة النحل الآية (78)

الاهداء

الهي لا يطيب الليل الا بشكرك ولا يطيب النهار الا بطاعتك ... ولا تطيب اللحظات الا بذكرك ... ولا
تطيب الاخرة الا بمغفرتك ... ولا تطيب الجنة الا برؤيتك

" الله ﷻ "

الى من بلغ الرسالة وادى الامانة ... ونصح الامة ... الى نبي الرحمة ونور العالمين

" سيدنا محمد ﷺ "

الى من كلفه الله بالهبة والوقار... الى من علمني العطاء دون انتظار الى من احمل اسمه بكل افتخار
ارجو من الله ان يمد في عمرك لترى ثمارا قد حان قطافها بعد طول انتظار وستبقى كلماتك نجوما اهتدي
بها اليوم وفي الغد والى الابد ...

" والدي العزيز "

الى ملاكي في الحياة ... الى معنى الحب الى معنى الحنان والتفاني ... الى بسمة الحياة وسر الوجود الى
من كان دعائها سر نجاحي وحنانها بلسم جراحي الى اعلى الحبايب

" امي الحبيبة "

الى الزهور التي تفتحت امامي وغمرتني بعطرها الفواح الى من رافقونا منذ ان حملنا حقائب صغيرة
ومعا سرنا الدرب خطوة بخطوة حتى الان لا اعرف اهديكم بحثي هذا ام انتم من زرعت شجرة المعرفة
في نفسي والان لكم ثمرة هذه الشجرة

" اخوتي واخواتي الأعزاء "

الى الاخوة والاخوات الى من تحلو بالاخاء وتميزو بالوفاء والعطاء الى ينابيع الصدق الصافي الى من
معهم سعدت وبرفقتهم في دروب الحياة الحلوة والحزينة سرت الى من كانوا معي على طريق النجاح
والخير الى من عرفت كيف اجدهم وعلموني الا اضيعهم

" اصدقائي "

الشكر والعرفان

في مثل هذه اللحظات يتوقف اليراع يفكر قبل ان يخط الحروف ليجمعها في كلمات تتبعثر الاحرف وعبثا ان يتم تجميعها في سطور ...

سطورا كثيرة تملأ كالخيال ولا يبقى لنا في نهاية المطاف الا قليلا من الذكريات وصور تجمعنا برفاق كانوا الى جانبنا

اليكم يا من كان لكم السبق في ركب العلم والتعليم

اليكم يا من بذلتم ولم تنتظرو العطاء

ونخص بالشكر والتقدير :

الدكتور / الامين كمال عبدالحى ابراهيم

الذين نقول لكم بشراكم قول رسول الله ﷺ : (ان الحوت في البحر والطير في السماء ليصلون على معلم الناس الخير) لابد لنا ونحن نخطو خطواتنا الاخيرة في الحياة الجامعية من وقفة نعود الى اعوام قضيناها في رحاب الجامعه مع اساتذتنا الكرام الذين قدموا لنا الكثير باذلين بذلك جهودا كبيرة في بناء جيل الغد لتبعث الامة من جديد ...

وقبل ان نمضي نقدم اسمى ايات الشكر والامتنان والتقدير والمحبة

الى الذين حملوا اقدس رسالة في الحياة ...

الى الذين مهدوا لنا طريق العلم والمعرفة ...

الى جميع اساتذتنا الافاضل في كلية الهندسة

فهرس المحتويات

الصفحة	الموضوع	الرقم
IV	الآية	
II	الإهداء	
IV	شكرو عرفان	
IV	فهرس المحتويات	
IV	فهرس الاشكال	
VII	المستخلص	
VIII	Abstract	
الفصل الاول (المقدمة)		
1	المقدمة	1-1
1	مشكلة البحث	2-1
1	اهمية البحث	3-1
1	اهداف البحث	4-1
1	منهجية البحث	5-1
2	بنية البحث	6-1
الفصل الثاني (الاطار النظري للبحث)		
3	مقدمة	1-2
3	معادلة السرعة لمحركات التيار المستمر	2-2
4	التحكم في محركات التيار المستمر	3-2
4	التحكم في محركات التيار المستمر عن طريق التحكم في الجهد	1-3-2
5	التحكم في السرعة عن طريق ربط مقاومة على التوالي مع عضو الانتاج	2-3-2
5	التحكم في السرعة عن طريق تغيير الفيض المغناطيسي	3-3-2
5	طريقة وارد – ليونارد للتحكم في السرعة	4-3-2
7	التحكم بريوستات	5-3-2
الفصل الثالث (تصميم ماكينة ال CNC)		
8	مقدمة	1-3
8	المكونات	2-3
8	متحكم الاردوينو	1-2-3
8	الدرع L293D	2-2-3
9	محرك السيرفو	3-2-3
9	مشغل اقراص ال DVD	4-2-3
9	محرك التيار المستمر	5-2-3
9	برنامج Inkscape	6-2-3
9	برنامج pronterface	7-2-3

9	مصدر التغذية	8-2-3
10	الدائرة العملية	3-3
10	طريقة العمل	1-3-3
10	المكونات المادية	2-3-3
11	البرامج المستخدمة	2-3-3
الفصل الرابع (النتائج والمناقشة)		
15	المخطط الصندوقي	1-4
16	المخطط الانسيابي	2-4
17	شرح المخطط الانسيابي	1-2-4
الفصل الخامس (الخلاصة والتوصيات)		
19	الخلاصة	1-5
19	التوصيات	2-5
20		المراجع
21		الملاحق

فهرس الاشكال

الصفحة	اسم الشكل	الرقم
6	نظام وارد – ليونارد للتحكم في السرعة	(1-2)
7	جهاز ريوسات للتحكم في سرعة محرك التيار المستمر	(2-2)
10	الدائرة العملية لماكينة ال CNC	(1-3)
11	متحكم الارديينو اونو	(2-3)
11	الدرع L293D	(3-3)
12	محرك السيرفو	(4-3)
12	محرك اقراص ال DVD	(5-3)
13	محرك التيار المستمر	(6-3)
13	الواجهة الرئيسية لبرنامج ال Inkscape	(7-3)
14	الواجهة الرئيسية لبرنامج ال Pronterface	(8-3)
14	مصدر التغذية	(9-3)
15	المخطط الصندوقي لماكينة ال CNC	(1-4)
16	المخطط الانسيابي لماكينة ال CNC	(2-4)
17	الشكل تم رسمه بواسطة المنظومة التي تم تصميمها	(3-4)
18	صورة توضيحية للماكينة التي تم تصميمها	(4-4)

المستخلص :

تقنية التحكم الرقمي تعرف على انها التحكم فى الماكينة او عملية الانتاج عن طريق استخدام اكواد مكونة من حروف وارقام وقد بدأ ظهور هذه التقنية فى منتصف القرن العشرين وبالتحديد فى عام 1952 م حيث تم أول تطوير لتقنية التحكم الرقمي فى معهد ماسوشوست للتقنية (MIT) بالتعاون مع شركة جون بارسونس للطائرات فى مدينة متشجان بالولايات المتحدة الأمريكية . ويرجع ذلك للحاجة الماسة لإنتاج قطع غاية فى الدقة لأشكال هندسية معقدة تشكل أجزاء من الطائرات الحربية وخصوصا مراوح الطائرات العمودية.

تكمن مشكلة البحث فى انه تم تصميم ماكينة ال CNC بحيث تعمل على الرسم فقط .

بدأت ماكينات ال CNC تغزو العالم لذا كان لابد من معرفة طرق ونظريات عملها وكيفية تصميم ماكينة CNC من المواد المحلية المتاحة وباقل تكلفه ممكنه .

الهدف من البحث تصميم و تنفيذ ماكينة CNC قادرة على العمل بدقة وبسرعة عالية لزيادة كفاءة الانتاج وبالإضافة لمقارنتها ببقية الماكينات وملاحظة ادائها .

تم استخدام المنهج التجريبي لتصميم وتنفيذ ماكينة ال CNC ومقارنتها مع الماكينات المعروفة فى مواد التصميم من حيث التكلفة وتم استخدام لغة Arduino C لتحرير برنامج الاربوينو للتشغيل وفق النظم الحديثة .

Abstract :

Digital control technology is defined as controlling the machine or the production process by using codes consisting of letters and numbers. The emergence of this technology began in the middle of the twentieth century, specifically in 1952 AD, when the first development of digital control . Technology was carried out at the Massachusetts Institute of Technology (MIT) in cooperation with the company John Parsons Aircraft Corporation in Michigan, USA. This is due to the urgent need to produce very precise parts for complex geometric shapes that form parts of warplanes, especially helicopters.

The research problem lies in the fact that the CNC machine is designed to work on drawing only.

The CNC is invading the world, so it was necessary to know the methods and theories of its work and how to design a CNC machine from the available local materials at the lowest possible cost.

The aim of the research is to design and implement a CNC machine capable of working accurately and at high speed to increase production efficiency, in addition to comparing it with the rest of the machines and observing its performance.

The method was used Experimental design and implementation of a CNC machine and compare it with known machines in design materials in terms of cost, and the Arduino C language was used to edit the Arduino program for programming Gil according to modern systems .

الفصل الاول

(المقدمة)

الفصل الاول

(المقدمة)

1-1 المقدمة :

تكنولوجيا التحكم الرقمي بالحاسب (Computer Numerical Control) :

تقنية التحكم الرقمي تعرف على انها التحكم فى الماكينة او عملية الانتاج عن طريق استخدام اكواد مكونة من حروف وارقام وقد بدأ ظهور هذه التقنية فى منتصف القرن العشرين وبالتحديد فى عام 1952 م حيث تم أول تطوير لتقنية التحكم الرقمي فى معهد ماسوشوست للتقنية (MIT) بالتعاون مع شركة جون بارسونس للطائرات فى مدينة منتشجان بالولايات المتحدة الأمريكية . ويرجع ذلك للحاجة الماسة لإنتاج قطع غاية فى الدقة لأشكال هندسية معقدة تشكل أجزاء من الطائرات الحربية وخصوصا مراوح الطائرات العمودية.

وهو ايضا يمثل نظام تحكم رقمي يستخدم فيه كمبيوتر (له ذاكرة لحفظ البرامج) للتحكم فى ماكينة التحكم الرقمي . ويمثل الكمبيوتر جزء لا يتجزأ من الماكينة , ويمكن برمجة ماكينة التحكم الرقمي مباشرة باستخدام لوحة مفاتيح الكمبيوتر أو بواسطة شريط مثقب (Punched Tape) يقوم الكمبيوتر بقراءته كما أن بعض ماكينات (CNC) يستطيع فيها الكمبيوتر بالإضافة إلي ما ذكر قراءة البرامج المسجلة علي اسطوانات.

2-1 مشكلة البحث :

تكمّن مشكلة البحث في انه تم تصميم ماكينة ال CNC بحيث تعمل على الرسم فقط .

3-1 اهمية البحث :

تكمّن اهمية البحث في ان ماكينات ال CNC بدأت تغزو العالم لذا كان لابد من معرفة طرق ونظريات عملها وكيفية تصميم ماكينة CNC من المواد المحلية المتاحة وبأقل تكلفه ممكنه .

4-1 اهداف البحث :

1- تصميم ماكينة CNC.

2- تنفيذ ماكينة CNC قادرة على العمل بدقة وبسرعة عالية لزيادة كفاءة الانتاج .

3-مقارنه الماكينه ببقية الماكينات وملاحظة ادائها .

5-1 منهجية البحث :

تم استخدام المنهج التجريبي لتصميم وتنفيذ ماكينة ال CNC ومقارنتها مع الماكينات المعروفة في مواد التصميم من حيث التكلفة وتم استخدام لغة Arduino C لتحرير برنامج الاردوينو للتشغيل وفق النظم الحديثة .

6-1 بنية البحث :-

يحتوي هذا البحث على عدة فصول تحتوي على :

- الفصل الاول بعنوان المقدمة ويحتوي على المقدمة ، مشكلة البحث ، اهمية البحث ، اهداف البحث ، منهجية البحث .
- الفصل الثاني بعنوان الاطار النظري للبحث ويحتوي على دراسات سابقة لمحركات التيار المستمر .
- الفصل الثالث بعنوان تصميم ماكينة ال CNC ويحتوي على الاجزاء المادية والبرامج المستخدمه والدائرة العملية للماكينة .
- الفصل الرابع بعنوان النتائج والمناقشة ويحتوي على المخطط الصندوقي و المخطط الانسيابي وشرح المخطط الانسيابي وصور للماكينه و رابط فيديو لعمل الماكينه .
- الفصل الخامس يحتوي على الخلاصة والتوصيات .

الفصل الثاني
(الاطار النظري للبحث)

الفصل الثاني

(الاطار النظري للبحث)

1-2 المقدمة :

التحكم في السرعة المحرك التيار المستمر يعني التغير المعتمد في سرعة الادارة للقيمة المطلوبة لاداء عملية تشغيل محددة والتغير ليس التغير الطبيعي في السرعة الذي يحدث بسبب التغير في الحمل علي عمود الادارة التغير المرغوب في السرعة يكون مصحوب بالتاثير علي محرك الادارة او علي الناقل الذي يصلها للوحدة التي تخدم ادارتها . ويتميز محرك التيار المستمر علي كل الانواع الاخري بحيث يكون ذلك من سهولة النسبة التي يمكن ان يفقد بها التحكم في السرعة , بحيث يتم التحكم في السرعة المحركات عن طريق الجهد المصدر (v) , المقاومة الخارجية في دائرة عضو الانتاج (R) , الفيض المغناطيسي (Φ) . ويتم التحكم في السرعة وذلك بعمل شغل جيد ولا يكون هنالك خلل في الماكينة التي يتم فية الشغل.

2-2 معادلة السرعة لمحرك التيار المستمر:

من معادلة القوة الدافعة الكهربائية المتولدة في عضو الانتاج لمحرك التيار المستمر تعطي العلاقة كالاتي :

$$E_b = (\Phi Z N / 60) * P / A \dots \dots \dots (2-1)$$

$$E_b = V - I_a R_a \dots \dots \dots (2-2)$$

المقارنة بين المعادلتين (2-1) و(2-2) نحصل علي

$$(\Phi Z N / 60) * P / A = V - I_a R_a =$$

$$N = (V - I_a R_a / \Phi Z) * (60 A / P) =$$

$$(K V - I_a R_a / \Phi Z) = K (E_b / \Phi) =$$

$$N = (V - I_a R_a / K \Phi) \dots \dots \dots (2-3)$$

المعادلة (2-3) هي معادلة السرعة لمحرك التيار المستمر ويمكن التحكم في السرعة بتغير في الجهد أو الهبوط في الجهد أو الفيض .

حيث ان :

(Z, A, P) هي ثوابت لالة خاصة لمحرك تيار مستمر, إذا كان القيمة الابتدائية للسرعة تيار عضو الانتاج القوة الدافعة الكهربائية المضادة الفيض القطب هي (N1, Ia1, Eb1, Φ1) علي الترتيب والقيمة النهائية المقابلة هي:

(N2, Ia2, Eb2 ,Φ2) علي الترتيب إذن نحصل علي المعادلات كالاتي :

$$E_{b1} = V - I_{a1} R_a \quad N_1 \alpha E_{b1} / \Phi_1$$

$$E_{b2} = V - I_{a2}R_a \quad N_2 \propto E_{b2}/\Phi_2$$

$$N_2/N_1 = (E_{b2}/\Phi_2)/(E_{b1}/\Phi_1) = (E_{b2}/E_{b1}) * (\Phi_1/\Phi_2) \dots (2-4)$$

وفي حالة المحرك باثارة توازي فإن الفيض يكون ذو قيمة ثابتة الا اذا تم تغيير في مقاومة لفيفة المجال وبالتالي $\Phi_1 = \Phi_2$

$$N_2/N_1 = (E_{b2}/E_{b1}) = (V - I_{a2}R_a / V - I_{a1}R_a) \dots (2-5)$$

وفي حالة التوالي لمحرك التيار المستمر قبل التشبع فإن $I_{se} \propto I_a \Phi$

$$\Phi_1/\Phi_2 = I_{a1}/I_{a2}$$

$$N_2/N_1 = (E_{b2}/E_{b1}) * (I_{a1}/I_{a2}) \dots (2-6)$$

(ELECTRIAL TECHNOLOGY VOL II سراجا)

2-3 التحكم في سرعة المحركات التيار المستمر:

ويتم التحكم في سرعة محركات التيار المستمر يتم بعدة طرق هي :

2-3-1 التحكم في السرعة عن طريق مصدر الجهد:

كل المحركات تشترك في أنه يمكن التحكم في سرعتها عن طريق التغيير في جهد المصدر حسب العلاقة الآتية:

$$N = (V - I_a R_a) / k\Phi \dots (2-7)$$

حيث أن :

V : الجهد المصدر

N : سرعة المحرك

I_a : تيار عضو الانتاج

R_a : مقاومة ملفات عضو الانتاج

K : ثابت

Φ : الفيض المغناطيسي.

كما يجب أن تأخذ في إعتبار مقاومة ملفات المجال في حالة محركات التوالي والمركبة لتصبح المعادلة:

$$n = (V - I_a R_a - I_a R_s) / k\Phi \dots (2-8)$$

Rs : مقاومة ملفات المجال التوالي (هذا بإهمال الهبوط في الجهد على الفرش الكربونية)

2-3-2 التحكم في السرعة عن طريق ربط مقاومة على التوالي مع عضو الانتاج:

من نفس المعادلة السابقة يمكن ملاحظة أنه يمكن التأثير على سرعة محركات التيار المستمر بربط مقاومة على التوالي مع عضو الإنتاج وبالتالي فإن الهبوط في الجهد سيرتفع مما يجعل من القوة الدافعة الهربائية تنخفض فتتخفض سرعة المحرك، أي أن العلاقة عكسية بين هذه المقاومة وسرعة المحرك. لهذه المقاومة وظيفة أخرى وهي التخفيض من تيار الإقلاع الذي يكون مرتفعاً جداً في حالة البدء مما يسبب إحتراق ملفات عضو الانتاج حتى ولو مر فيها لثواني بسيطة.

2-3-3 التحكم في السرعة عن طريق تغيير الفيض المغناطيسي:

ربط مقاومة على التوالي مع ملفات المجال في محركات التوازي عند ربط هذه المقاومة فإن تيار المجال سوف يقل مما يسبب في خفض قيمة الفيض المغناطيسي ، وإذا لاحظت من المعادلة السابقة فإن العلاقة عكسية بين الفيض والسرعة وبالتالي فإن التخفيض في الفيض سيجعل من سرعة المحرك ترتفع.

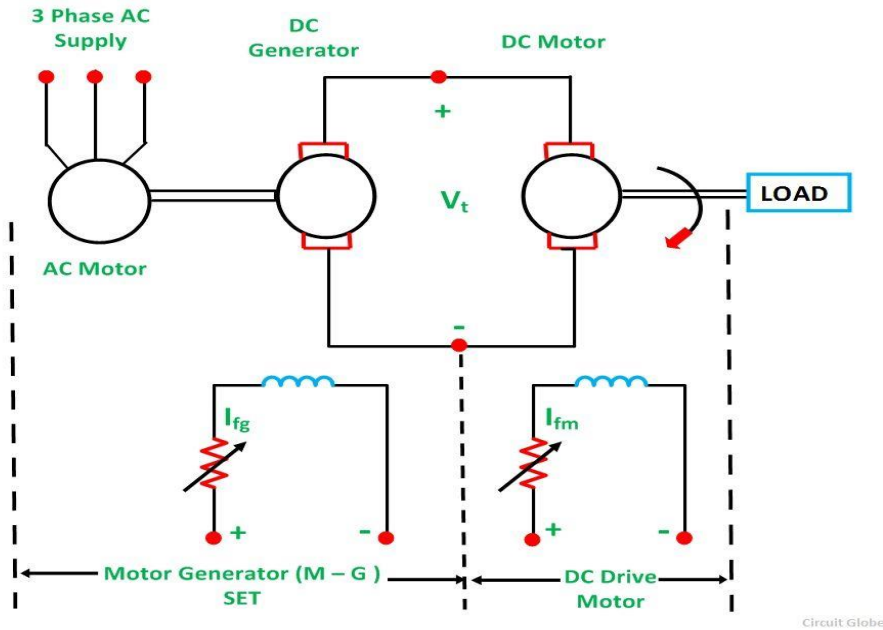
ربط مقاومة على التوازي مع ملفات المجال في حالة محرك التوالي كلما زدنا من قيمة هذه المقاومة سيرتفع التيار المار في ملفات المجال مما يسبب في رفع الفيض وبالتالي التخفيض في سرعة المحرك.

التغيير في جهد المجال في حالة المحرك المنفصل الإثارة بما أن هناك مصدري جهد في هذا النوع من المحركات فإنه يمكننا أن نغير من قيمة الجهد بالتغيير في جهد المجال المتصل مع ملفات المجال وبالتالي سيتغير تيار المجال الذي سيغير بدوره سيغير الفيض فتتغير سرعة المحرك.

هذه هي إجمالاً طرق التحكم في سرعة محركات التيار المستمر وقد حاولت قدر الإمكان الإختصار لإن هذا الموضوع طويل ويحتاج للرسمات التوضيحية (الدوائر المكافئة) .

2-3-4 طريقة وارد – ليوناردو للتحكم في السرعة:

طريقة التحكم في السرعة بالتحكم المضبوط في جهد عضو الانتاج الاساسية مصحوب بوسيلة مولد الجهد المضبوط تسمى (نظام وارد - ليونارد) هذا النظام يتكون بتشغيل المحرك بآثارة ثابتة والتاثير بجهد متغير علي عضو انتاجه ليعطي السرعة المطلوبة . مصدر الجهد المتغير يتم الحصول عليه من مولد – محرك أو مجموعة مغير في الشكل (1-3) حيث M1 محرك العمل موصل بالمولد G يتم الحصول عليه من المسستثير E المركب علي نفس عمود المولد مجموعة وارد ليونارد تبدأ ببدء محرك الادارة ريوستات المجال R للمولد يخرج تدريجياً من الدائرة بمجرد أن يلتقط المولد السرعة ويبدأ محرك العمل في الدوران , الجهد المتغير عبر اطراف المولد او عبر المحرك يتم الحصول عليها بتغيير تيار الاثارة للمولد G بواسطة منظم التوازي R ,ويمكن عكس اتجاه دوران عضو إنتاج المحرك بعكس اتجاه تيار الاثارة للمولد G بمساعدة مفتاح العكس Rs مجموعة المغير تدور دائماً في نفس الاتجاه.



الشكل (1-2) يوضح نظام وراذ - ليونارد للتحكم في السرعة

فرملة المحرك M1 يمكن تنفيذها بتخفيض إثارة المولد حتي تكون ق.د.ك له أقل من ق.د.ك المضادة للمحرك M1 , وتحت هذه الشروط , يبدأ المحرك M1 في العمل مؤقتاً كمولد والمولد G كمحرك وألة الادارة ذات التيار المتردد M2 كمولد وكنتيجة فإن طاقة الحركة للمحرك M1 وحملة ترجع للمصدر الرئيسي ويأخذ تأثير الفرملة مكانة علي المحرك M1 .

ومن الميزات الاساسية في هذا النظام يكون التحكم في السرعة ناعم جداً عبر كل مدي من السرعة الصغيرة الي السرعة العادية ويمكن الحصول عليية في كلا الاتجاهين , وايضاً الحصول علي عجلة منتظمة وتنظيم سرعة جيد , وهذا النظام للتحكم في السرعة مناسب جداً حيث يكون مطلوب تحكم سرعة غير محدود تقريباً في كلا الاتجاهي الدوران مثل مفازز الصلب , الات الورق , المصاعد ولاناوش وروافع المناجم.

ومن عيوب ذها النظام هو مظلوب التين زيادة ولذلك فهي مكلفة وكفاءة عامة منخفضة لنظام وخاصة عند الاعمال الخفيفة.

5-3-2 التحكم بريوستات:

في هذه الطريقة تغيير السرعة يحقق بواسطة مقاومة متغيره داخله في التوالي مع مجال التوازي كما هو موضح في الشكل (2-3)اي زيادة في مقاومة التحكم يخفض تيار المجال مع التخفيض الناتج في الفيض وزيادة في السرعة . هذه الطريقة للتحكم في السرعة بسيطة جداً , مناسبة واكثر اقتصاداً , وتستخدم بكثرة في الادارة الكهربائية الحديثة . لكن هذه الطريقة لديها عيوب من عيوب هذه الطريقة بانه لا يمكن الحصول علي سرعات التدفق , يحصل غلي السرعات العالية جداً فقط عند العزم المنخفض بسبب المجال الضعيف جداً, تكون السرعات نهاية عظمي عند النهاية الصغري للفيض والتي تحكم بالتاثير إزالة المغنطة لرد فعل المنتج علي المجال .

1500W **120V AC**
15A



شكل (2-2) يوضح جهاز الريوستات للتحكم في سرعة محرك التيار المستمر

الفصل الثالث

(تصميم ماكينة ال CNC)

الفصل الثالث

(تصميم ماكينة ال CNC)

1-3 مقدمة :

بعد البحث حول ماكينات التحكم العددي المحوسب تم بناء ماكينة التحكم العددي المحوسب الخاصة بنا باستخدام مكونات متوفرة محلياً . هنالك الكثير من ماكينات التحكم العددي المحوسب في العالم ، بعضها يكون متطوراً ومعقداً في التكنولوجيا وحتى في التشغيل . من أجل هذا السبب تم تصميم وصنع ماكينة تخطيط بالتحكم العددي المحوسب ومبنية على أردينو ، والذي يعتبر الأبسط في الصنع ، في هذا المشروع تم صنع ماكينة تحكم عددي محوسب تقوم برسم معظم الأشكال الأساسية ، النصوص وحتى الكرتون . عملها مشابه لطريقة عمل اليد البشرية . وهي أسرع وأكثر دقة مقارنة باليد البشرية في الرسم والكتابة .

2-3 المكونات :

1-2-3 متحكم الاردوينو :

اردوينو أونو لوحة تحكم دقيقة مفتوحة المصدر تعتمد على وحدة التحكم الدقيقة Microchip ATmega328P وتم تطويرها بواسطة Arduino.cc تم تجهيز اللوحة بمجموعات من دبابيس الإدخال/الإخراج الرقمية والتناظرية (I/O) التي يمكن توصيلها بالوحدات التمديد المختلفة (الدروع) والدوائر الأخرى. تحتوي اللوحة على 14 منفذ إدخال/إخراج رقمي (ستة منها قادرة على إخراج PWM)، و 6 منافذ إدخال/إخراج تناظرية، وهي قابلة للبرمجة باستخدام Arduino IDE (بيئة التطوير المتكاملة)، عبر كبل USB من النوع B يمكن تشغيله عن طريق كبل USB أو بواسطة بطارية خارجية 9 فولت، على الرغم من أنه يقبل الفولتية بين 7 و 20 فولت. إنه مشابه لـ Arduino Nano و Leonardo يتم توزيع التصميم المرجعي للأجهزة بموجب ترخيص Creative Commons Attribution Share-Alike 2.5 وهو متاح على موقع Arduino الإلكتروني. تتوفر أيضاً ملفات التخطيط والإنتاج لبعض إصدارات الأجهزة ، والشكل (1-3) يوضح متحكم الاردوينو أونو .

2-2-3 الدرع L293D :

يوفر الدرع إجمالي 4 قناطر Bridges-H ويمكن لكل قنطرة H تقديم ما يصل إلى 0.6 A للمحرك الواحد .

يأتي الدرع أيضاً مع مسجل أزاحة HC59574 يمتد 4 دبابيس رقمية من Arduino إلى 8 دبابيس للتحكم في الاتجاه لرققتين L293D ، والشكل (2-3) يوضح شكل Shield L293D .

3-2-3 محرك السيرفو :

محرك السيرفو أو "المحرك المؤازر" مشغل إما دوار أو خطي يسمح بالتحكم الدقيق في كلا من الموضع سواء كان زاوي أو خطي، السرعة والتسارع. يتكون السيرفو من محرك مناسب مقترن بجهاز استشعار (حساس) لمراجعة الاشارات القادمة بخصوص الموضع. يتطلب الجهاز تحكم متطور نسبيا، يكون في أغلب الأحيان وحدة مخصصة مصممة للاستخدام مع نفس طراز السيرفو.

تعتبر أجهزة السيرفو فئة محددة من فئات المحركات على الرغم من أن المصطلح يستخدم في حالة ضرورة وجود محرك ذو دقة عالية للقيام بمهمة ما في نظام تحكم في دائرة مغلقة مثل صناعة الروبوتات، آلات التحكم الرقمي باستخدام الحاسوب، آلات التصنيع الآلي، خطوط التعبئة، خطوط التغليف وخطوط الحمل والنقل، والشكل (3-3) يوضح محرك السيرفو .

4-2-3 محرك اقراص ال DVD :

يستخدم محرك الاقراص للاستفادة من المحركات ال dc الموجودة بها , ايضا يستخدم الهيكل الداخلي كحامل للموتورات ، والشكل (4-3) يوضح محرك الاقراص .

5-2-3 محرك التيار المستمر :

محرك التيار المستمر (بالإنجليزية: DC Motor) هو عبارة عن آلة تحوّل الطاقة الكهربائية إلى طاقة ميكانيكية باستخدام التيار المستمر، وهو بالتالي يعمل فقط على أنظمتها، وتنقسم أنواعه إلى نوعين بناءً على التركيب، واحدهما: هو محرك متمائل القطبين وهو أول اختراع يقوم بشغل فيزيائي بواسطة الحث الكهرومغناطيسي ، والشكل (5-3) يوضح محرك التيار المستمر .

6-2-3 برنامج InkScape :

إنكسكيب هو برنامج ذو جودة احترافية لرسمات المتجهات يعمل على أنظمة Windows و Mac و GNU و Linux ، يتم استخدامه من قبل محترفي التصميم والهواة في جميع أنحاء العالم ، لإنشاء مجموعة واسعة من الرسومات مثل الرسوم التوضيحية والرموز والشعارات والمخططات والخرائط ورسومات الويب، يستخدم البرنامج (SVG (Scalable Vector Graphics) ، وهو برنامج مجاني ومفتوح المصدر ، والشكل (6-3) يوضح الواجهة الرئيسية لبرنامج InkScape .

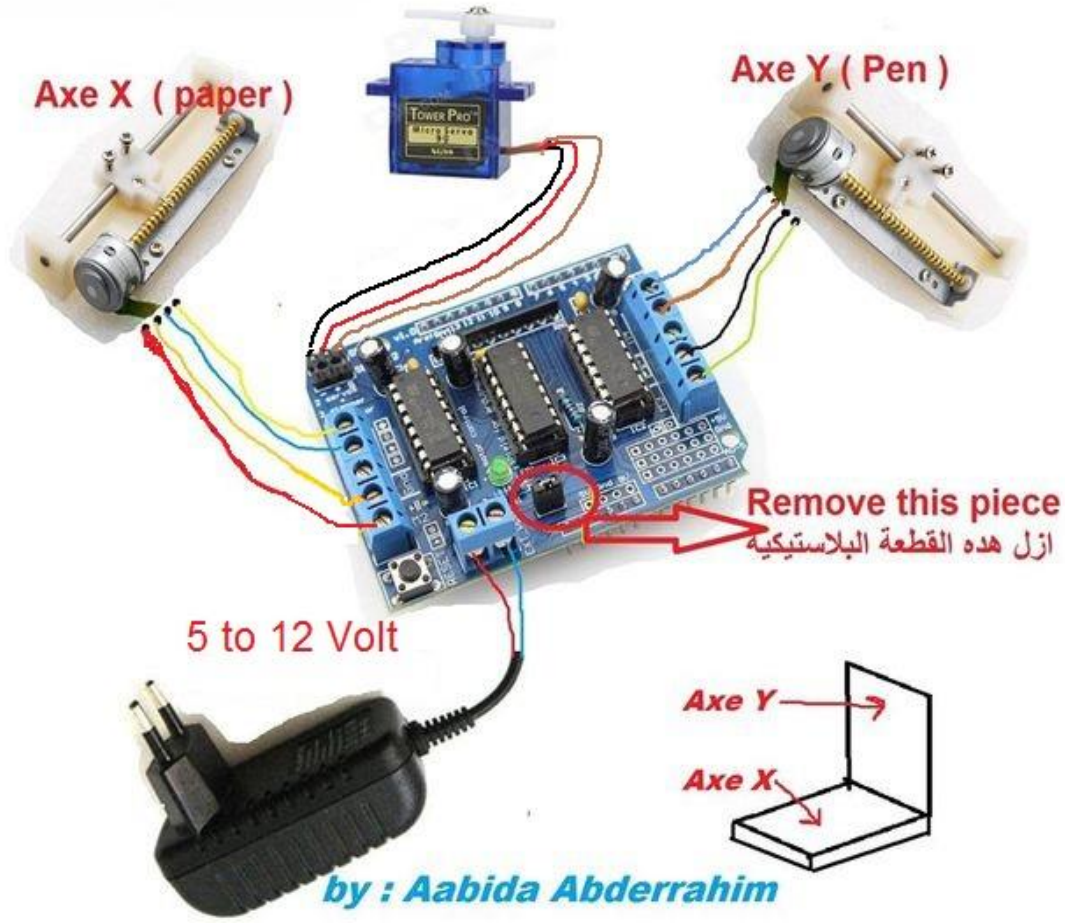
7-2-3 برنامج Pronterface :

يتم توصيل الماكينة بالحاسوب، ونقل ملف الجي كود عن طريق برنامج برونترفيس-Pronterfa وهو مسؤول عن التحكم في الماكينة أثناء التشغيل ، والشكل (7-3) يوضح الواجهة الرئيسية لبرنامج . Pronterface

8-2-3 مصدر التغذية :

يقوم بتحويل التيار المتردد 220V الى تيار مستمر لتغذية الماكينة بطاقة قدرها 9V ، والشكل (7-3) يوضح مصدر التغذية .

3-3 الدائرة العملية :



شكل (1-3) يوضح الدائرة العملية لماكينة ال CNC

4-3 طريقة العمل :

تتكون ماكينة CNC التي تم تصميمها من جزئين:

1-4-3 المكونات المادية :

وهي كل الأجزاء التي نراها ونستطيع لمسها .

يتم احضار قطع من الخشب وتثبيتها بحيث تكون القطعتين متعامدتين فوق بعضهما لتمثل احدهما محور X والقطعة الاخرى تمثل محور Y ويتم تثبيت اقراص ال DVD التي تحتوي على عدد 2 موتور احدهما على المحور X والاخر على المحور Y حيث يتحرك كل موتور حسب امر استلمه من الاردوينو الذي بدوره قم بتفسير كل حركه قام بها برنامج pronterface الذي بدوره استلم الرسم او التوقيع من برنامج Inkspace الذي قام بتحويلهما الى صيغة Gcode ويتم تركيب الاردوينو والشيلد على القاعدة الخشبية .

وايضا يتم تركيب موتور سيرفو على محور Y متعامدا معه بحيث يعمل كمحور Z .

2-4-3 البرامج المستخدمة :

تم استخدام برنامج Inkspace لتحويل الصورة من صيغة PNG الى صيغة gcode ونقلها برنامج pronterface ليقوم بارسال الصورة الى الاردوينو ليقوم الاردوينو بمعالجتها ورسماها .



شكل (3-3) يوضح الدرع L293D



شكل (2-3) يوضح متحكم الاردوينو اونو



شكل (5-3) يوضح محرك اقراص ال DVD



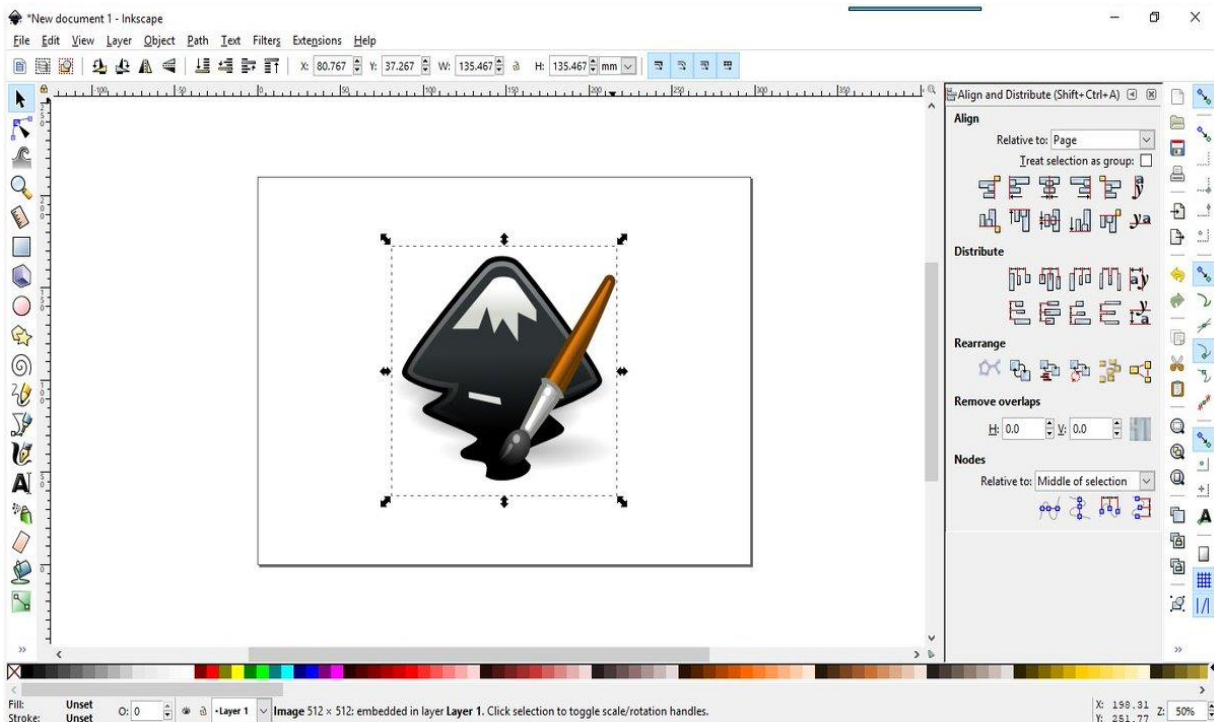
شكل (4-3) يوضح محرك السيرفو



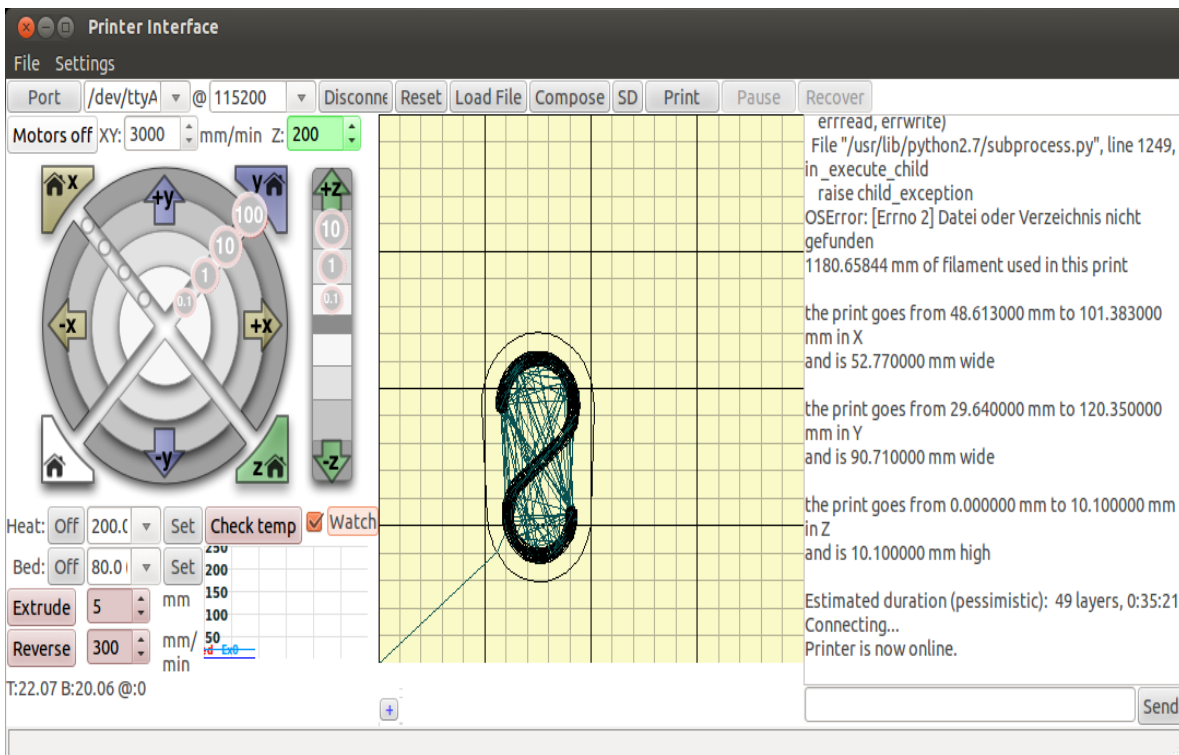
شكل (7-3) يوضح مصدر التغذية



شكل (6-3) يوضح محرك التيار المستمر



شكل (7-3) يوضح الواجهة الرئيسية لبرنامج Inkscape



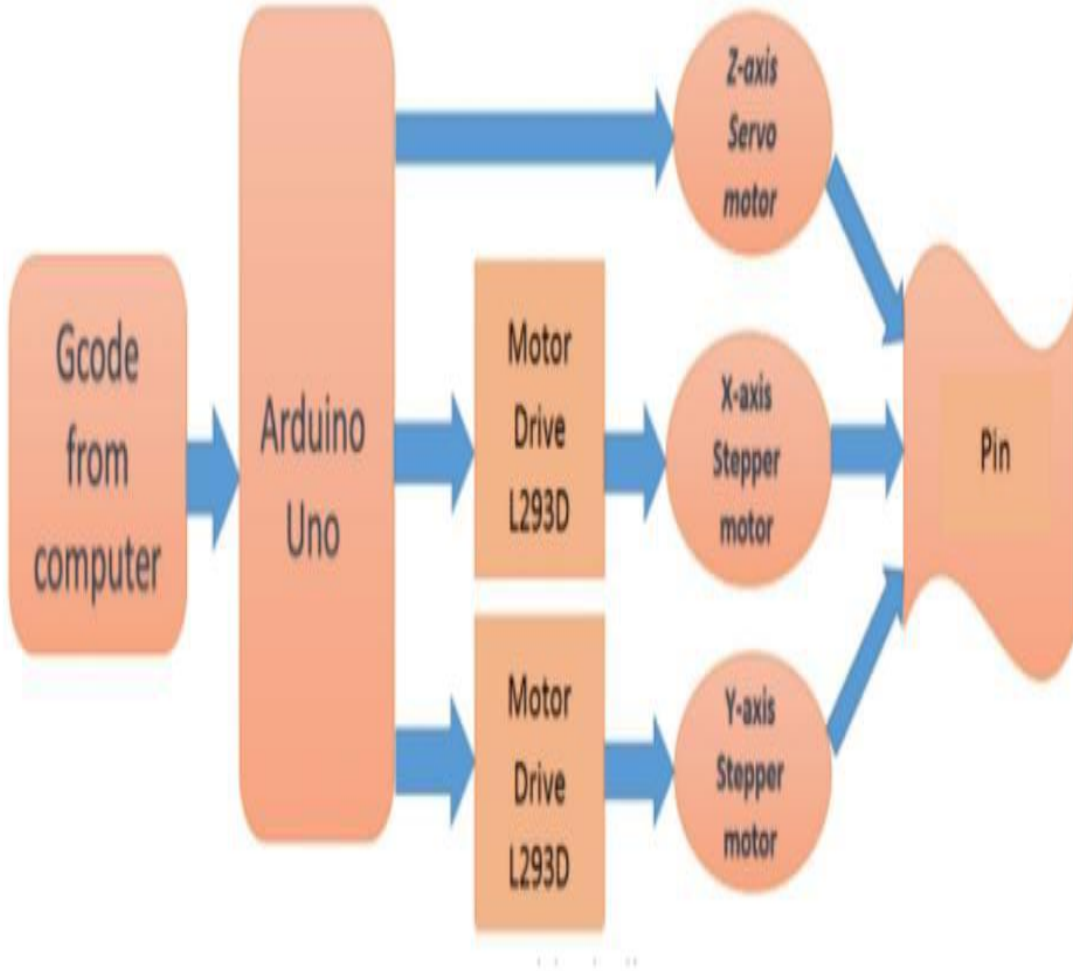
شكل (8-3) يوضح الواجهة الرئيسية لبرنامج Pronterface

الفصل الرابع (النتائج والمناقشة)

الفصل الرابع (النتائج والمناقشة)

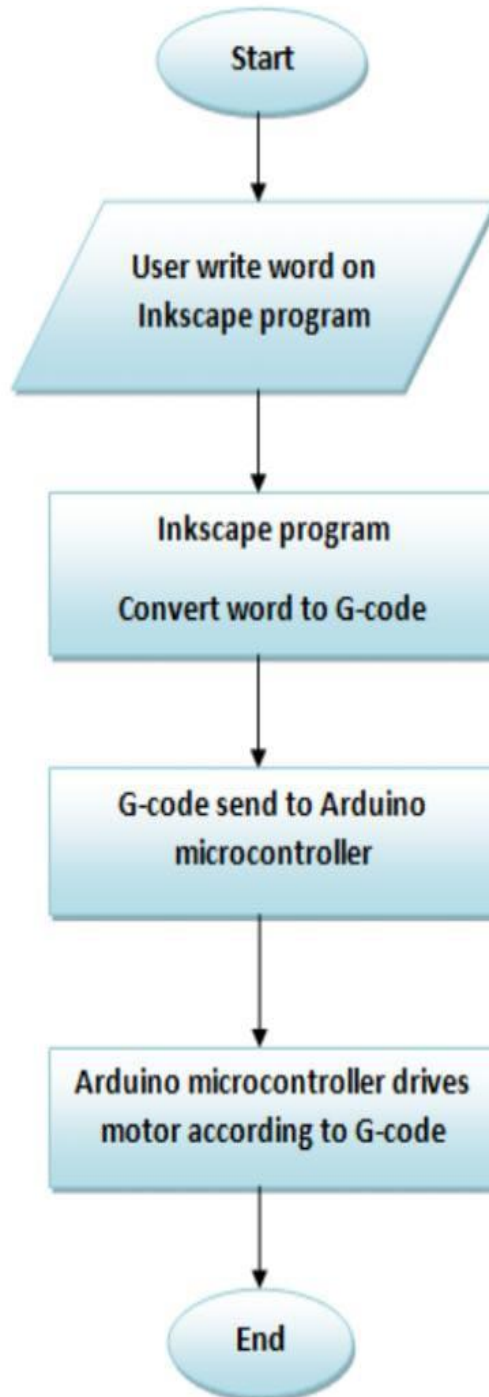
1-4 المخطط الصندوقي :

هو المخطط الذي يوضح اجزاء المشروع وطريقة عملها وكذلك توصيلها .



شكل (1-4) يوضح المخطط الصندوقي لماكينة ال CNC

2-4 المخطط الانسيابي :



شكل (2-4) يوضح المخطط الانسيابي لماكينة ال CNC

1-2-4 شرح المخطط الانسيابي :

يوضح المخطط الانسيابي طريقة عمل المنظومة من بداية البرنامج وحتى نهايته ففي بداية البرنامج يتم ادخال الصورة في برنامج inkspace بصيغة png بحيث يقوم برنامج inkspace الى صيغة gcode ثم بعد ذلك يتم ارسال الصورة الى الاردوينو بحيث يقوم الاردوينو بتحريك الموتورات وفقا لصيغة ال gcode .

الشكل (3-4) تم رسمه بواسطة المنظومة التي تم تصميمها.



الشكل (3-4) تم رسمه بواسطة المنظومة التي تم تصميمه

الشكل (4-4) يوضح المنظومة التي تم تصميمها.



شكل (4-4) يوضح المنظومة التي تم تصميمها

الرابط التالي يوضح فيديو للمنظومة التي تم تصميمها أثناء التجارب:

<https://t.me/hassangaffer97>

الفصل الخامس

(الخلاصة و التوصيات)

الفصل الخامس

(الخلاصة و التوصيات)

1-5 الخلاصة :

تم تصميم ماكينة CNC تقوم بالرسم بسرعه ودقة عالية عن طريق متحكمة اردوينو ومساعدة بعض البرامج من اجل تشغيل واطفاء الماكينه .

2-5 التوصيات :

- تطوير الماكينة لكي تقوم بالرسم ثلاثي الابعاد .
- تطوير الماكينه للرسم على غير الورق .

(المراجع)

(المراجع)

1. المهندس عبد الله علي - اردوينو ببساطه- مكتبة الكتب - الطبعة الاولى - 2015.
2. محركات السيرفو - المكتبة الهندسية
3. <http://electrohakim.blogspot.com/2019/07/servo-motors.html?m=1>

(الملاحق)

الملاحق

البرنامج :

```
#include <Servo.h>

#include <AFMotor.h>

#define LINE_BUFFER_LENGTH 512

char STEP = MICROSTEP ;

// Servo position for Up and Down

const int penZUp = 95;

const int penZDown = 83;

// Servo on PWM pin 10

const int penServoPin =10 ;

// Should be right for DVD steppers, but is not too important here

const int stepsPerRevolution = 48;

// create servo object to control a servo

Servo penServo;

// Initialize steppers for X- and Y-axis using this Arduino pins for the L293D H-bridge

AF_Stepper myStepperY(stepsPerRevolution,1);

AF_Stepper myStepperX(stepsPerRevolution,2);

/* Structures, global variables */

struct point {

    float x;

    float y;
```

```

float z;
};
// Current position of plothead
struct point actuatorPos;
// Drawing settings, should be OK
float StepInc = 1;
int StepDelay = 0;
int LineDelay =0;
int penDelay = 50;
// Motor steps to go 1 millimeter.
// Use test sketch to go 100 steps. Measure the length of line.
// Calculate steps per mm. Enter here.
float StepsPerMillimeterX = 100.0;
float StepsPerMillimeterY = 100.0;
// Drawing robot limits, in mm
// OK to start with. Could go up to 50 mm if calibrated well.
float Xmin = 0;
float Xmax = 40;
float Ymin = 0;
float Ymax = 40;
float Zmin = 0;
float Zmax = 1;
float Xpos = Xmin;
float Ypos = Ymin;

```

```

float Zpos = Zmax;
// Set to true to get debug output.
boolean verbose = false;
// Needs to interpret
// G1 for moving
// G4 P300 (wait 150ms)
// M300 S30 (pen down)
// M300 S50 (pen up)
// Discard anything with a (
// Discard any other command!

/*****
 * void setup() - Initialisations
 *****/
void setup() {
  // Setup
  Serial.begin( 9600 );

  penServo.attach(penServoPin);
  penServo.write(penZUp);
  delay(100);
  // Decrease if necessary
  myStepperX.setSpeed(600);
  myStepperY.setSpeed(600);

```

```

// Set & move to initial default position
// TBD
// Notifications!!!
Serial.println("Mini CNC Plotter alive and kicking!");
Serial.print("X range is from ");
Serial.print(Xmin);
Serial.print(" to ");
Serial.print(Xmax);
Serial.println(" mm.");
Serial.print("Y range is from ");
Serial.print(Ymin);
Serial.print(" to ");
Serial.print(Ymax);
Serial.println(" mm.");
}

```

```

/*****

```

```

* void loop() - Main loop

```

```

*****/

```

```

void loop()

```

```

{

```

```

  delay(100);

```

```

  char line[ LINE_BUFFER_LENGTH ];

```

```

  char c;

```

```

int lineIndex;

bool lineIsComment, lineSemiColon;

lineIndex = 0;

lineSemiColon = false;

lineIsComment = false;

while (1) {

    // Serial reception - Mostly from Grbl, added semicolon support

    while ( Serial.available()>0 ) {

        c = Serial.read();

        if (( c == '\n' ) || ( c == '\r' ) ) {           // End of line reached

            if ( lineIndex > 0 ) {                     // Line is complete. Then execute!

                line[ lineIndex ] = '\0';             // Terminate string

                if (verbose) {

                    Serial.print( "Received : ");

                    Serial.println( line );

                }

                processIncomingLine( line, lineIndex );

                lineIndex = 0;

            }

            else {

                // Empty or comment line. Skip block.

            }

            lineIsComment = false;

            lineSemiColon = false;

        }

    }

}

```

```

Serial.println("ok");
}
else {
    if ( (lineIsComment) || (lineSemiColon) ) { // Throw away all comment
characters
        if ( c == ')' ) lineIsComment = false; // End of comment. Resume line.
        }
    else {
        if ( c <= ' ' ) { // Throw away whitespace and control
characters
            }
        else if ( c == '/' ) { // Block delete not supported. Ignore
character.
            }
        else if ( c == '(' ) { // Enable comments flag and ignore all
characters until ')' or EOL.
            lineIsComment = true;
            }
        else if ( c == ';' ) {
            lineSemiColon = true;
            }
        }
    else if ( lineIndex >= LINE_BUFFER_LENGTH-1 ) {
        Serial.println( "ERROR - lineBuffer overflow" );
        lineIsComment = false;
        lineSemiColon = false;
        }
}

```



```

// G1 X30 Y50
// M300 S30 (pen down)
// M300 S50 (pen up)
// Discard anything with a (
// Discard any other command!

while( currentIndex < charNB ) {
    switch ( line[ currentIndex++ ] ) {          // Select command, if any
    case 'U':
        penUp();
        break;
    case 'D':
        penDown();
        break;
    case 'G':
        buffer[0] = line[ currentIndex++ ];    // !\ Dirty - Only works with 2 digit
commands
        //   buffer[1] = line[ currentIndex++ ];
        //   buffer[2] = '\0';
        buffer[1] = '\0';

        switch ( atoi( buffer ) ){            // Select G command
        case 0:                               // G00 & G01 - Movement or fast movement.
Same here
        case 1:

```

```

// !\ Dirty - Suppose that X is before Y

char* indexX = strchr( line+currentIndex, 'X' ); // Get X/Y position in the
string (if any)

char* indexY = strchr( line+currentIndex, 'Y' );

if ( indexY <= 0 ) {

    newPos.x = atof( indexX + 1);

    newPos.y = actuatorPos.y;

}

else if ( indexX <= 0 ) {

    newPos.y = atof( indexY + 1);

    newPos.x = actuatorPos.x;

}

else {

    newPos.y = atof( indexY + 1);

    indexY = '\0';

    newPos.x = atof( indexX + 1);

}

drawLine(newPos.x, newPos.y );

//    Serial.println("ok");

actuatorPos.x = newPos.x;

actuatorPos.y = newPos.y;

break;

}

break;

case 'M':

```

```
buffer[0] = line[ currentIndex++ ];    // !\ Dirty - Only works with 3 digit
commands
```

```
buffer[1] = line[ currentIndex++ ];
```

```
buffer[2] = line[ currentIndex++ ];
```

```
buffer[3] = '\0';
```

```
switch ( atoi( buffer ) ){
```

```
case 300:
```

```
{
```

```
char* indexS = strchr( line+currentIndex, 'S' );
```

```
float Spos = atof( indexS + 1);
```

```
// Serial.println("ok");
```

```
if (Spos == 30) {
```

```
penDown();
```

```
}
```

```
if (Spos == 50) {
```

```
penUp();
```

```
}
```

```
break;
```

```
}
```

```
case 114:                // M114 - Report position
```

```
Serial.print( "Absolute position : X = " );
```

```
Serial.print( actuatorPos.x );
```

```
Serial.print( " - Y = " );
```

```
Serial.println( actuatorPos.y );
```

```
break;
```

default:

```
Serial.print( "Command not recognized : M");
```

```
Serial.println( buffer );
```

```
}
```

```
}
```

```
}
```

```
}
```

```
/**
```

```
* Draw a line from (x0;y0) to (x1;y1).
```

```
* Bresenham algo from https://www.marginallyclever.com/blog/2013/08/how-to-build-an-2-axis-arduino-cnc-gcode-interpreter/
```

```
* int (x1;y1) : Starting coordinates
```

```
* int (x2;y2) : Ending coordinates
```

```
*/
```

```
void drawLine(float x1, float y1) {
```

```
if (verbose)
```

```
{
```

```
Serial.print("fx1, fy1: ");
```

```
Serial.print(x1);
```

```
Serial.print(",");
```

```
Serial.print(y1);
```

```
Serial.println("");
```

```
}
```

```
// Bring instructions within limits
```

```
if (x1 >= Xmax) {
    x1 = Xmax;
}
if (x1 <= Xmin) {
    x1 = Xmin;
}
if (y1 >= Ymax) {
    y1 = Ymax;
}
if (y1 <= Ymin) {
    y1 = Ymin;
}

if (verbose)
{
    Serial.print("Xpos, Ypos: ");
    Serial.print(Xpos);
    Serial.print(",");
    Serial.print(Ypos);
    Serial.println("");
}

if (verbose)
{
```

```

Serial.print("x1, y1: ");
Serial.print(x1);
Serial.print(",");
Serial.print(y1);
Serial.println("");
}

// Convert coordinates to steps
x1 = (int)(x1*StepsPerMillimeterX);
y1 = (int)(y1*StepsPerMillimeterY);
float x0 = Xpos;
float y0 = Ypos;

// Let's find out the change for the coordinates
long dx = abs(x1-x0);
long dy = abs(y1-y0);
int sx = x0<x1 ? StepInc : -StepInc;
int sy = y0<y1 ? StepInc : -StepInc;

long i;
long over = 0;

if (dx > dy) {
    for (i=0; i<dx; ++i) {

```

```

myStepperX.onestep(sx,STEP);
over+=dy;
if (over>=dx) {
    over-=dx;
    myStepperY.onestep(sy,STEP);
}
delay(StepDelay);
}
}
else {
    for (i=0; i<dy; ++i) {
        myStepperY.onestep(sy,STEP);
        over+=dx;
        if (over>=dy) {
            over-=dy;
            myStepperX.onestep(sx,STEP);
        }
        delay(StepDelay);
    }
}

if (verbose)
{
    Serial.print("dx, dy:");

```

```
Serial.print(dx);  
Serial.print(",");  
Serial.print(dy);  
Serial.println("");  
}
```

```
if (verbose)
```

```
{  
  Serial.print("Going to (");  
  Serial.print(x0);  
  Serial.print(",");  
  Serial.print(y0);  
  Serial.println(")");  
}
```

```
// Delay before any next lines are submitted
```

```
delay(LineDelay);
```

```
// Update the positions
```

```
Xpos = x1;
```

```
Ypos = y1;
```

```
}
```

```
// Raises pen
```

```
void penUp() {
```

```
penServo.write(penZUp);
delay(penDelay);
Zpos=Zmax;
digitalWrite(15, LOW);
  digitalWrite(16, HIGH);
if (verbose) {
  Serial.println("Pen up!");

}
}
// Lowers pen
void penDown() {
  penServo.write(penZDown);
  delay(penDelay);
  Zpos=Zmin;
  digitalWrite(15, HIGH);
  digitalWrite(16, LOW);
  if (verbose) {
    Serial.println("Pen down.");

  }
}
```