



Universidade do Minho

Departamento de Engenharia Mecânica

Departamento de Eletrónica Industrial

Ricardo José da Cunha Fernandes

Máquina de medir por coordenadas de baixo custo

Tese de Mestrado

Metrologia e Modelação

Trabalho efetuado sob a orientação dos professores

António Alberto Caetano Monteiro

Agostinho Gil Teixeira Lopes

Outubro de 2012

Declaração

Nome:

Ricardo José da Cunha Fernandes

Endereço de correio eletrónico: rrccff@gmail.com

Telefone: 911019091

Número do cartão do cidadão: 10967576

Título de dissertação

Máquina de medir por coordenadas de baixo custo

Orientadores:

Professor António Alberto Caetano Monteiro

Professor Agostinho Gil Teixeira Lopes

Ano de conclusão: 2012

Designação do Mestrado: Mestrado de 2º Ciclo em Engenharia Mecatrónica

É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA TESE/TRABALHO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE.

Universidade do Minho, ____/____/____

Assinatura: _____

Agradecimentos

Apesar do trabalho solitário que implica muitas vezes uma dissertação tenho a realçar e agradecer os contributos de quem me apoiou neste processo.

Ao professor António Alberto Caetano Monteiro, pela grande fonte de motivação que alimentava a minha vontade de vencer nesta dissertação, pela competência científica e transversal e pelo apoio que sempre demonstrou, críticas e sugestões sempre muito pertinentes.

Ao professor Agostinho Gil Teixeira Lopes, pela competência científica, pela visão aberta na procura de soluções, pelo apoio, críticas e sugestões.

À minha adorável esposa, professora Marta Teixeira Ferreira da Silva Fernandes, pelo apoio e segurança inculcada durante todo este processo tendo sido sempre paciente e compreensiva.

Ao engenheiro Paulo José Abreu Martins pela disponibilidade que teve para a troca de ideias sobre este assunto.

Ao meu tio João Rocha da Cunha que pela sua experiência mecânica me ajudou a dar os primeiros passos experimentais nesta área.

Título

MMC de baixo custo

Resumo

A metrologia e a modelação têm um papel fundamental numa sociedade onde a competitividade está francamente relacionada com a qualidade. Uma indústria de ponta não pode estar desinteressada desta área e a busca de soluções metrológicas de baixo custo vem permitir o aumento da qualidade dos produtos e ao mesmo tempo da competitividade das empresas. A ampla gama de conhecimentos gerados na mecânica, eletrónica e na informática potencializam os conhecimentos das áreas da ciência existentes no currículo de ensino nacional, assim como se tornam num veículo de motivação e experenciação dos alunos. O desenvolvimento desta área deve irromper nas bases de estudos dos futuros cidadãos potencializando a modernidade da sociedade. Ao dissertar sobre o desenvolvimento de uma máquina de medir por coordenadas desde os domínios da sua conceção, irrompendo pelo domínio pedagógico é possível formar um juízo mais claro sobre as suas potencialidades no desenvolvimento de uma nação. Esta investigação abrange diversos domínios científicos e pedagógicos numa pluralidade interdisciplinar.

Palavras chave

Mecânica, Eletrónica, Informática, Metrologia, MMC, Digitalizador 3D

Title

Low cost CMM

Abstract

Metrology and modeling play an important role within a society where competitiveness is directly related to quality. A cutting-edge industry cannot be indifference to this area whereas the search for low cost metrological solutions raises the product's quality as well as the competitiveness of the enterprises. The big range of knowledge related to mechanics, electronics and computation empower the urge in science study inside the national education curriculum being, besides this, a greater factor of motivation and experimentation for most students. The development in this area should burst into the bases of the education of all future citizens and this way help leveraging the modern society. When we speak about development of a coordinate measurement machine from the domains of its conception, breaking out the pedagogical domain it is possible to form a clear judgment about its potential within the development of a nation. This research covers various scientific fields in a plurality interdisciplinary teaching.

Key words

Mechanics, Electronics, Computing, Metrology, MMC, 3D Scanner

Índice

Agradecimentos	v
Título	vii
Resumo	vii
Palavras chave.....	vii
Title	ix
Abstract.....	ix
Key words.....	ix
Abreviaturas e siglas	xv
Figuras	xvii
Tabelas.....	xix
Equações.....	xxi
Diagramas.....	xxiii
1. Introdução.....	1 -
1.1. Objeto do estudo.....	1 -
1.2. Potencialidades económicas.....	2 -
1.3. Potencialidades pedagógicas	4 -
1.4. Estrutura adotada.....	5 -
1.5. Objetivos a atingir.....	6 -
2. Estado da arte	7 -
2.1. Introdução	7 -
2.2. MMC baseada em sistemas de 3 eixos	7 -
2.3. MMC baseada em sistemas de varrimento.....	10 -
2.4. MMC baseada em sistemas laser	11 -
2.5. MMC baseada em sistemas de luz estruturada	12 -
2.6. MMC baseada em sistemas de ondas estacionárias.....	12 -
2.7. Análise comparativa	13 -

2.8.	Controlo eletrónico	14 -
2.8.1.	Acionamento direto por computador.....	14 -
2.8.2.	Acionamento com recurso a um sistema com microcontrolador.....	15 -
2.8.3.	Acionamento com recurso a um sistema embebido.....	16 -
2.8.4.	Utilização de motores de passo para o deslocamento dos eixos	17 -
2.8.5.	Utilização de servomotores ou motores DC.....	18 -
2.9.	Potencialidades dos sistemas CAD	18 -
2.10.	Aplicação de visão por computador à MMC	19 -
2.10.1.	Modelos de visão por computador	20 -
2.10.2.	Visão por infravermelhos.....	20 -
2.10.3.	Visão estéreo	21 -
2.11.	Algoritmos de medição.....	21 -
3.	Desenvolvimento de uma MMC.....	23 -
3.1.	Componentes estruturais.....	24 -
3.1.1.	Base e Pórtico	26 -
3.1.2.	Eixos.....	27 -
3.1.3.	Apalpador	29 -
3.2.	Cinemática direta	30 -
3.3.	Sistema de microcontrolador utilizado	33 -
3.3.1.	Controlo do motor.....	35 -
3.3.2.	Controlo do mecanismo de medição	36 -
3.3.3.	Controlo do fim de curso dos eixos.....	37 -
3.3.4.	Visão geral do sistema eletrónico implementado.....	39 -
3.4.	Programação	40 -
3.4.1.	Programação do microcontrolador.....	40 -
3.4.2.	Programação da aplicação de visão por computador	44 -
3.4.3.	Programação da leitura de coordenadas.....	46 -

4. Resultados	- 49 -
5. Conclusões e trabalho futuro	- 51 -
Bibliografia.....	- 53 -

Abreviaturas e siglas

CAD – Computer-aided Design (Desenho assistido por computador)

DC – Direct Current (Corrente contínua)

DLL – Dynamic-link Library (Livraria dinâmica)

DWF – Design Web Format (Formato de desenho para a Internet)

GND – Ground (Ligação equipotencial)

MMC – Máquina de Medir por Coordenadas

PIC – Peripheral Interface Controller (Controlador de interface periférica)

TTL – Transistor – Transistor Logic (Lógica a transístor)

USB – Universal Serial Bus (Barramento série universal)

VBA – Visual Basic for Applications (Visual Basic para aplicações)

Figuras

Fig. 1 Modelo Mitutoyo - Legex® 1200 Series (Mitutoyo America Corporation)	7 -
Fig. 2 Componentes de um apalpador (Fernandes, 2010)	8 -
Fig. 3 Comparação da rigidez de flexão dos materiais (Renishaw, 2011)	9 -
Fig. 4 Comparação de materiais - coeficiente de expansão/massa (Renishaw, 2011)	9 -
Fig. 5 Apalpador de ponta direita (Fernandes, 2010)	9 -
Fig. 6 Seleção da ponta. A – comprimento total, B – comprimento efetivo, C – diâmetro da haste, D – diâmetro da esfera (Fernandes, 2010)	10 -
Fig. 7 Sistema de varrimento para obtenção de geometrias de superfícies (Park Systems, 2008)	11 -
Fig. 8 Digitalização por laser (Nikon, 2011)	11 -
Fig. 9 Digitalização com recurso a sistemas de luz estruturada (APM Technologies)	12 -
Fig. 10 Digitalizador por ultrassons (The London Podiatry Centre - Scanning Technology, 2009)	13 -
Fig. 11 Placas de desenvolvimento de sistemas embebidos (Foundation, 2012) (Banzi, Cuartielles, Igoe, Martino, & Mellis, 2005)	16 -
Fig. 12 Motor de passo CKD existente numa MMC Roland do dep. de mecânica	17 -
Fig. 13 Método de obtenção da circularidade de uma peça (Mitutoyo America Corporation)	21 -
Fig. 14 Sequência cinemática de uma MMC de 3 eixos (Fernandes, 2010)	24 -
Fig. 15 MMC desenvolvida em alumínio com base e pórtico móvel	26 -
Fig. 16 Sistema de deslocamento dos eixos	27 -
Fig. 17 Aplicação da resina epóxi	28 -
Fig. 18 Fixação com parafuso dos eixos de deslocamento	28 -
Fig. 19 Apalpador de ponta em bico (Fernandes, 2010)	29 -
Fig. 20 Apalpador construído com recurso a uma agulha em aço	29 -
Fig. 21 Eixos do cabeçote de uma possível configuração (Renishaw, 2011)	30 -
Fig. 22 Objeto de aplicação do método de Denavit-Hartenberg e respetiva estrutura das juntas (Renishaw, 2011)	30 -
Fig. 23 Introdução do referencial do objeto (Renishaw, 2011)	32 -
Fig. 24 Placa de desenvolvimento PIC-Ready1 (MikroElektronika, 1998)	33 -
Fig. 25 Placa de desenvolvimento Arduino Uno (Banzi, Cuartielles, Igoe, Martino, & Mellis, 2005)	34 -
Fig. 26 Ambiente de programação Arduino (Banzi, Cuartielles, Igoe, Martino, & Mellis, 2005)	34 -
Fig. 27 Placa de controlo de motores EasyDrive (Schmalz, nd.)	35 -

Fig. 28 Regulador de corrente - EasyDrive (Schmalz, nd.)	35 -
Fig. 29 Ligação do sensor piezoelétrico ao Arduino Uno.....	36 -
Fig. 30 Ligação da agulha ao Arduino Uno.....	37 -
Fig. 31 Efeito de Ripple nos fim de curso.....	38 -
Fig. 32 Circuito de controlo dos fins de curso	38 -
Fig. 33 Remoção do efeito de Ripple	38 -
Fig. 34 Circuito completo do sistema implementado.....	39 -
Fig. 35 Comunicação série com o Arduino	40 -
Fig. 36 Folha de cálculo preenchida com dados do Opencv	45 -
Fig. 37 Formulário de controlo manual da MMC	46 -
Fig. 38 Verificação do tamanho da peça com Opencv.....	50 -

Tabelas

Tabela 1 Produção europeia de máquinas em milhões de euros no ano de 2008 (Eurostat, 2012).	2 -
Tabela 2 Tabela comparativa de tecnologias metrológicas	14 -
Tabela 3 Parâmetros de Denavit-Hartenberg.....	31 -
Tabela 4 Ajustamento do <i>microstepping</i> na placa EasyDrive	36 -
Tabela 5 Excerto de leitura da ADC pelo apalpador baseado no piezoelétrico	49 -
Tabela 6 Dados intrínsecos da MMC	50 -

Equações

Equação 1 Matrizes de transformação homogéneas	- 31 -
Equação 2 Matriz de transformação resultante relativa à MMC	- 32 -
Equação 3 Matriz de transformação resultante relativa ao objeto a medir.....	- 32 -

Diagramas

Diagrama 1 Sistema eletrónico com utilização da porta paralela	15 -
Diagrama 2 Sistema eletrónico com utilização de um microcontrolador.	15 -
Diagrama 3 Estrutura da MMC.....	25 -
Diagrama 4 Algoritmo principal de funcionamento do Arduino	42 -
Diagrama 5 Rotina do Arduino de movimentação de um eixo.....	43 -
Diagrama 6 Algoritmo da biblioteca DLL com Opencv.....	44 -
Diagrama 7 Algoritmo do código VBA no Inventor	45 -
Diagrama 8 Algoritmo de varrimento	47 -
Diagrama 9 Algoritmo de apuramento das coordenadas obtidas na visão por computador	48 -

1. Introdução

O mundo atravessa um período de grandes desafios para todas as nações. A Europa vive uma crise que está a afetar a sua população expondo as fragilidades económicas e sociais. Esta situação levou a Comissão Europeia a elaborar um plano estratégico, "Europa 2020". Nesta perspetiva encontra-se a necessidade de melhoria da competitividade industrial, otimização de recursos e inovação. A Comissão Europeia aponta para o alargamento dos horizontes das empresas para além dos seus setores tradicionais salientando a necessidade de garantir a uma "fertilização cruzada de ideias" e de modelos de negócios (Comissão Europeia, 2012). Mas não se pode falar de inovação sem incluir a educação como um dos principais motores. Esta dissertação vem criar um equipamento com o qual se pretende incutir na cultura educativa as bases de estudo para uma sociedade produtiva e tecnológica.

1.1. Objeto do estudo

A tecnologia de obtenção de coordenadas metrológicas de objetos estabeleceu-se durante muitos anos pelo desenvolvimento de equipamentos mecânicos capazes de determinar as dimensões de objetos. A necessidade de determinar as formas dos objetos levou à criação de equipamentos mecânicos de três eixos: as máquinas de medir por coordenadas (MMC). Esta tecnologia apareceu em meados do século XX. A empresa "Ferranti Ltd" com o desenvolvimento de equipamento de controlo numérico impulsionado pelo governo Inglês (Museum, 2008), automatizou a MMC com o aparecimento dos sistemas microeletrónicos nos anos 60, que permitiu elevar a classe de exatidão destes equipamentos. A necessidade de investigação e desenvolvimento rápido de produtos fez surgir novas tecnologias para a medição tais como o recurso a sistemas laser. Hoje em dia existem poderosos equipamentos de medição

dimensional, tanto em exatidão como em rapidez de funcionamento de acordo com a aplicação que se pretende.

A potencialidade destes equipamentos é fulcral, tanto no desenvolvimento de produtos como nos processos de avaliação da qualidade. As empresas de fabrico de produtos de alta qualidade têm incutido na sua organização um plano de qualidade que está diretamente ligado ao planeamento e realização do produto.

1.2. Potencialidades económicas

Ao analisar os países mais desenvolvidos da Europa, não querendo criar uma visão redutora da realidade mundial mas focalizando a comparação no mercado europeu, verifica-se uma clara ligação ao facto de serem estes os que demonstram maior movimentação económica no fabrico de máquinas-ferramentas. Esta realidade vem apoiar a ideia de que a existência de uma cultura de desenvolvimento de máquinas-ferramentas é altamente benéfica para o desenvolvimento de um país (Pruvot, 1993).

Tabela 1 Produção europeia de máquinas em milhões de euros no ano de 2008 (Eurostat, 2012).

	Países	Fabrico de máquinas de uso geral	Fabrico de máquinas e equipamento não especificado	Fabrico de outras máquinas de uso geral	Total
1	DE - Germany (including former GDR from 1991)	65387,1	204511	56618	326516,1
2	IT - Italy	25691,3	103128,1	32370	161189,4
3	FR - France	14903,2	45136,5	15384,2	75423,9
4	UK - United Kingdom	9992,8	34957,1	12788,8	57738,7
5	ES - Spain	4671,8	22798,6	9753,8	37224,2
6	DK - Denmark	11124,1	17814,4	3067,4	32005,9
7	SE - Sweden	4446,1	19831,2	6917,1	31194,4
8	AT - Austria	3155,8	17364,4	5201,3	25721,5
	...				
16	PT - Portugal	846,4	2716,3	928,7	4491,4

Segundo a base de dados do Eurostat (Tabela 1), a produção de máquinas na Europa, segundo os últimos dados que são referentes ao ano de 2008, é dominada pelos países mais desenvolvidos onde se destacam a Alemanha, Itália, França, Reino Unido e Espanha. Portugal encontra-se nos últimos lugares no desenvolvimento de máquinas de produção. Apesar da integração de Portugal na União Europeia, as políticas de investimento viradas para a aquisição de máquinas e para a produção de baixo valor acrescentado com mão-de-obra barata, traduziu-se numa economia que não foi capaz de fazer frente à abertura do comércio mundial. Na fase de adesão, a modernização das linhas de produção à custa da aquisição de máquinas desenvolvidas por países terceiros, colocou Portugal perante a dependência desses países, não permitindo ter tecnologia capaz de os igualar na produção de bens de consumo e acima de tudo de os superar. Nesta linha pode-se ainda apontar que a falta de tradição na produção de máquinas-ferramentas, leva à dificuldade de garantir a "fertilização cruzada de ideias" prevista no plano "Europa 2020". Se existir um forte investimento numa cultura de produção de máquinas-ferramentas e com essa mesma cultura "fertilizar" a produção industrial nacional é possível atingir uma produção virada para o desenvolvimento de produtos e processos que até então estavam aprisionados pela tecnologia estrangeira (Pruvot, 1993). Estamos ainda perante uma situação potencialmente desejada quando a palavra "inovação" consta em todos os discursos político-económicos. O incentivo ao desenvolvimento de máquinas-ferramentas numa cultura alheia ao mesmo, pode a curto prazo ver processos produtivos culturalmente estabilizados serem alavancados para um novo paradigma e com isso para produtos inovados e processos produtivos mais eficientes. Ao mesmo tempo criam-se postos de trabalho altamente qualificados capazes de absorver cidadãos que saem do sistema de ensino nacional. Com o objetivo geral traçado nesta dissertação de desenvolver um equipamento pedagógico de medição por coordenadas que em muitos aspetos se relaciona com o fabrico e desenvolvimento das máquinas-ferramentas, tais como, a mecânica, o controlo eletrónico, a programação, o controlo de qualidade, e toda a área comercial e de estudo do produto, pretende-se dar um contributo para o desenvolvimento pedagógico e económico do país.

1.3. Potencialidades pedagógicas

O desenvolvimento de uma máquina-ferramenta pode ser por excelência um grande apoio pedagógico nas várias áreas disciplinares (Pruvot, 1993). À primeira vista uma máquina-ferramenta pode ser pouco atrativa já que está limitada a um desempenho perceptível, mas o desenvolvimento destas tecnologias infere no domínio industrial e científico. Abrangendo áreas desde a investigação, produção, fabricação, gestão e comercialização, este é um perfeito recomeço para uma sociedade que se quer assente na inovação e desenvolvimento. A utilização de equipamentos como uma MMC durante o ensino, permite que os alunos construam e interajam com máquinas. O relacionamento com equipamentos ligados ao sistema produtivo, realizado desde primeiros ciclos de ensino, ajuda a criar uma cultura direcionada para o mesmo. Este contacto proporciona aos alunos a possibilidade de escolha de outras vias de ensino, vocacionais e profissionais, que de outro modo não estariam no seu horizonte quer por acharem complicadas ou até mesmo por desconhecimento da sua existência. Mesmo sem entrar nas vias vocacionais do ensino básico e profissionais do ensino secundário, que têm claramente conteúdos relacionados com esta área de estudo, pela sua relação direta com o saber fazer das profissões, ao analisar os documentos clarificadores das prioridades nos conteúdos fundamentais do currículo nacional do ensino básico (Crato, 2011), facilmente se encontram áreas de transversalidade entre o conhecimento gerado pelo desenvolvimento e utilização de uma MMC e esses mesmos documentos. Para clarificação desta situação e a título de exemplo, um dos conteúdos fundamentais da matemática para o 7º ano abrange o estudo das figuras geométricas, a noção de paralelismo, congruência, semelhança e medida; em português e para o 8º ano encontra-se na escrita a necessidade de escrever para expressar conhecimentos, escrever textos informativos e argumentativos; nas tecnologias de informação e comunicação existe a necessidade no 7º ano da produção e edição de documentos; na educação tecnológica existe já no 5º ano a necessidade de discriminar a relevância do objeto técnico, dominar a aquisição de conhecimento técnico, reconhecer tipos de grandeza e respetivos instrumentos de medição; na educação visual e no âmbito da representação para o 7º ano há a necessidade de dominar tipologias de representação bi e tridimensional e no âmbito do discurso a necessidade de compreender a noção de superfície e de sólido. A utilização de máquinas-ferramenta,

nomeadamente, a MMC, num projeto curricular de turma transversal a todas as disciplinas vêm permitir aos alunos compreender melhor e muitas vezes aceitar a necessidade revelada por cada disciplina quanto aos seus conteúdos e objetivos. Ao mesmo tempo, para além de ser um meio integrador de grande parte dos conteúdos a serem estudados nos ciclos básicos de ensino, cria-se uma cultura virada para o desenvolvimento tecnológico de grande relevância para uma sociedade que se pretende mais avançada.

1.4. Estrutura adotada

Após a introdução à dissertação e a definição dos objetivos a atingir no capítulo 1 e iniciou-se uma verificação do estado da arte retratada durante o capítulo 2. Neste, realizou-se uma análise comparativa das tecnologias existentes para a obtenção de coordenadas metrológicas, nomeadamente, os sistemas mecânicos de três eixos, sistemas de varrimento, sistemas laser, sistemas de luz estruturada e sistemas de ondas estacionárias. Foram ainda analisadas no capítulo 2 as tecnologias eletrónicas associadas aos sistemas mecânicos de três eixos, a potencialidade do uso dos sistemas de CAD em conjunto com a visão por computador nos processos de medição, assim como a algoritmia dos mesmos.

No capítulo 3 fica relatado todo o trabalho realizado para a obtenção de um equipamento simples, capaz de integrar num sistema básico de medição com 3 eixos a visão por computador através de um sistema CAD. Iniciou-se o capítulo com a construção mecânica e respetiva cinemática direta, seguido de todo o controlo eletrónico implementado e terminando na programação do sistema eletrónico e da visão por computador associada ao sistema CAD.

No capítulo 4 são relatados os resultados obtidos. Todos os resultados são analisados no capítulo 5 e são retiradas as respetivas conclusões procurando apontar para os campos de possível trabalho futuro.

1.5. Objetivos a atingir

Nesta dissertação pretende-se a preparação de um dispositivo para o mercado de educação, capaz de incentivar e motivar os jovens desde os ciclos iniciais de formação. Para atingir esse objetivo, ao contrário da génese de uma dissertação normal, serão abordados todos os assuntos desde a sua forma mais básica apontando sempre para os caminhos de desenvolvimento mais audazes, capazes de cativar também os alunos dos ciclos de formação superior. Será um equipamento capaz de ser construído pelos alunos e cuja integração das áreas vocacionais, profissionais e de projeto existentes no atual sistema de ensino, será uma mais-valia.

O resultado final deverá demonstrar a possibilidade de poder ser construído um dispositivo educativo económico capaz de aferir, com maior ou menor exatidão as medidas de uma peça. Será construído um sistema de 3 eixos atuados por motores, controlado por um sistema eletrónico e informático. Os sistemas, mecânico e eletrónico, deverão pela sua simplicidade, serem capazes de serem montados por alunos do 3º ciclo de ensino, estando estes também aptos de interagir com essa montagem através do sistema informático. Todo o sistema poderá ser criado, montado e instalado pelos alunos dos cursos profissionais de mecatrónica, eletrónica e computadores assim como poderão ser realizadas pequenas melhorias pelos próprios. Os alunos do ensino superior terão nesta dissertação uma base potencial para o desenvolvimento de sistemas de medição por coordenadas. Estes poderão aperfeiçoar todas as áreas em que assenta o desenvolvimento deste equipamento, que se pretende autónomo, dotado de visão por computador e integrado em sistemas de grande potencialidade da área da modelação, como o programa *Inventor*.

2. Estado da arte

2.1. Introdução

As tecnologias abordadas neste texto estão enquadradas no campo da metrologia aplicada ou industrial. Na era industrial, a aplicação da ciência de medição ao processo de fabrico de equipamentos veio assegurar o desenvolvimento de instrumentos de medição capazes de garantir o fabrico com processos de calibração e controlo de qualidade do mesmo. Com o desenvolvimento da eletrónica associada a novas tecnologias foram aparecendo novos sistemas de medição capazes tanto de diminuir o tempo de medição como aumentar a sua qualidade.

2.2. MMC baseada em sistemas de 3 eixos

Os sistemas de medição dimensional mais comuns, baseados em 3 eixos, recorrem a equipamentos baseados em pórticos. São sistemas que permitem atingir exatidões na ordem de micrómetro (Fig. 1). As mesas são fabricadas geralmente em granito ou ferro fundido com sistemas de molas isoladoras por amortecimento a ar, usados na eliminação das vibrações.



Fig. 1 Modelo Mitutoyo - Legex® 1200 Series (Mitutoyo America Corporation)

A configuração da MMC vai definir a estrutura da mesma o que tem influência no seu desempenho. Esta deve contribuir para a estabilidade dimensional, a rigidez, baixar a inércia de funcionamento, ter baixo coeficiente de expansão térmica, grande condutividade térmica, entre outros (Bosh, 1995). Desta forma, o granito e o alumínio são dois dos materiais mais usados na construção das MMC. O alumínio é por excelência o elemento de grande leveza e ao mesmo tempo de rigidez, o que permite ser usado principalmente para as partes móveis como o pórtico. O facto de ter um elevado coeficiente de expansão térmica comparado por exemplo com o granito, este obtém mais rapidamente a estabilidade térmica quando existem perturbações de temperatura. Deste modo, apesar do seu fraco desempenho térmico, o granito é usado como superfície de trabalho pela sua durabilidade e também por servir como elemento de amortecimento de vibrações que têm um efeito negativo na exatidão e repetibilidade. A sua característica térmica poder ser compensada pelo controlo de temperatura ou pela correção das medições ao nível dos algoritmos de leitura (Bennett & Weekers, 2010).

Esta tecnologia recorre a sondas de contacto para a determinação geométrica das peças a medir. Um dos componentes com especial contributo para uma maior ou menor qualidade de medição é o apalpador. Este elemento da MMC é fixo ao cabeçote que normalmente faz parte do braço no eixo OZ. O apalpador é composto por dois elementos: um corpo e um módulo removível, onde se fixam as pontas (Fig. 2). As pontas são um elemento chave já que fazem o contacto com a peça (Fernandes, 2010).

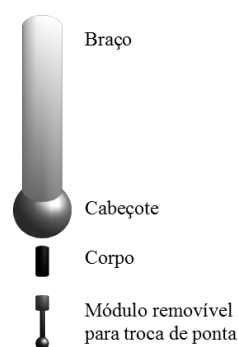


Fig. 2 Componentes de um apalpador (Fernandes, 2010)

Vários materiais poderão ser utilizados na sua construção, nomeadamente, o rubi, nitreto de silicone, zircónio, cerâmica ou carboneto de tungsténio (Fig. 3).

Material	E modulus in kN/mm ²
Tungsten carbide	620
Steel	200
Aluminium	70
Titanium	150
Ceramic	300–400
Carbon fibre	≥450

Fig. 3 Comparação da rigidez de flexão dos materiais (Renishaw, 2011)

Os suportes e as hastes das pontas são em geral construídos a partir de titânio, carboneto de tungstênio, aço inoxidável, cerâmica e fibra de carbono (Fig. 3 e Fig. 4). Estas devem evitar o dobrar excessivo pois tem impacto direto na incerteza da medição.

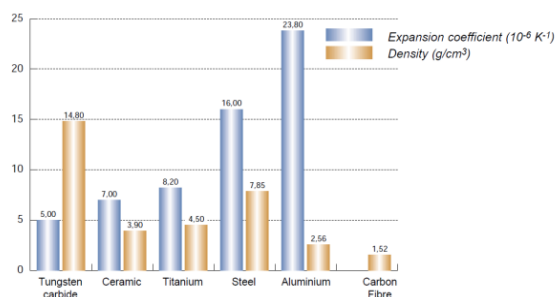


Fig. 4 Comparação de materiais - coeficiente de expansão/massa (Renishaw, 2011)

Existem ainda diversos tipos de pontas para apalpadores de acordo com a geometria da peça a ser medida (Renishaw, 2011). O tipo de pontas mais comum é a direita (Fig. 5) e é adequada para a maioria das medições. Este tipo de ponta incorpora uma bola de um material extremamente duro (por exemplo o rubi ou diamante), que também deve ter o mínimo de inércia para que durante movimento da máquina não existam falsas medições e sejam minorados os erros de medição (Fernandes, 2010).

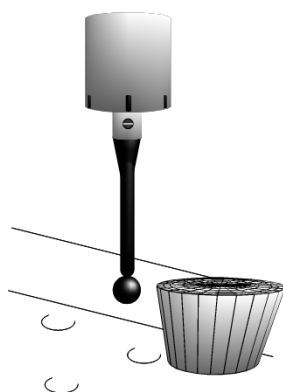


Fig. 5 Apalpador de ponta direita (Fernandes, 2010)

É possível melhorar a exatidão do apalpador se for bem escolhida a ponta de medição. Deve-se manter a ponta curta, pois vai defletir menos melhorando a medição. A utilização de muitas extensões que unem as pontas, introduzem pontos de flexão e deflexão potenciais, logo deve-se recorrer ao mínimo de extensões possível (Fig. 6).

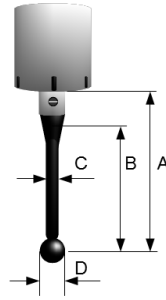


Fig. 6 Seleção da ponta. A – comprimento total, B – comprimento efetivo, C – diâmetro da haste, D – diâmetro da esfera (Fernandes, 2010)

A esfera deve ser escolhida tendo em conta que quanto maior o seu tamanho, maior é a distância entre a esfera e a haste reduzindo a possibilidade de falsa deteção causada pelo desvio da haste, aumentando ainda o comprimento efetivo. Uma esfera maior vai permitir também ter uma haste mais robusta aumentando a rigidez da ponta e ainda minimiza o efeito do acabamento superficial da peça a medir (Fernandes, 2010).

As áreas de trabalho podem variar de pouco milímetros a vários metros e a resolução destes equipamentos podem atingir os $0,01 \mu\text{m}$ (Mitutoyo America Corporation).

2.3. MMC baseada em sistemas de varrimento

As MMC de varrimento, utilizam a tecnologia dos microscópios de varrimento, que através da deflexão de um feixe de luz, permite determinar a geometria de superfícies. Esta tecnologia é usada, por exemplo, para a medição de rugosidades nas superfícies das peças (Fig. 7).

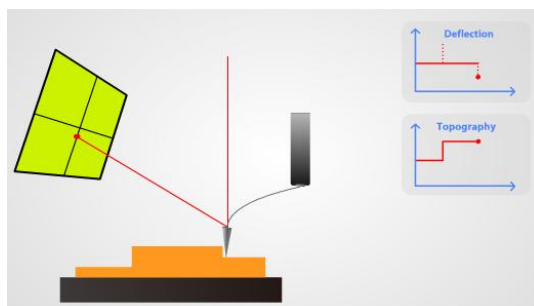


Fig. 7 Sistema de varrimento para obtenção de geometrias de superficies (Park Systems, 2008)

Na variante mecânica destes sistemas podemos atingir resoluções na ordem dos 0.1 nm, uma linearidade do eixo de varrimento de 0,015 % com uma área de trabalho 100 μm x 100 μm (Park Systems, 2008). Não existem informações relativamente a outros dados metrológicos comparativos.

2.4. MMC baseada em sistemas laser

Uma das tecnologias mais recentes é o recurso à digitalização por laser (Fig. 8). Esta solução permite uma rápida captura da geometria de qualquer tipo de objeto e sem necessidade de contacto. A exatidão dos dados obtidos ainda é reduzida em comparação com as possibilidades permitidas na verificação metrológica por sondas de contacto. Mas vai de facto acelerar a engenharia inversa, pois neste caso, frequentemente, não serão necessários dados metrológicos de grande precisão.



Fig. 8 Digitalização por laser (Nikon, 2011)

A área de trabalho destes sistemas pode variar entre os 3 mm por 65 mm e 300 mm por 200 mm com uma resolução de 5 μm (Nikon, 2011).

2.5. MMC baseada em sistemas de luz estruturada

Estes sistemas permitem obter a medição através da projeção de padrões de luz estruturada (Fig. 9). Esta projeção de padrões sobre a peça vai permitir a triangulação e obtenção de coordenadas XYZ através de algoritmos de fotogrametria. Esta ciência tem por base dois tipos de algoritmos, aéreos para cartografia ou de proximidade em aplicações de metrologia.

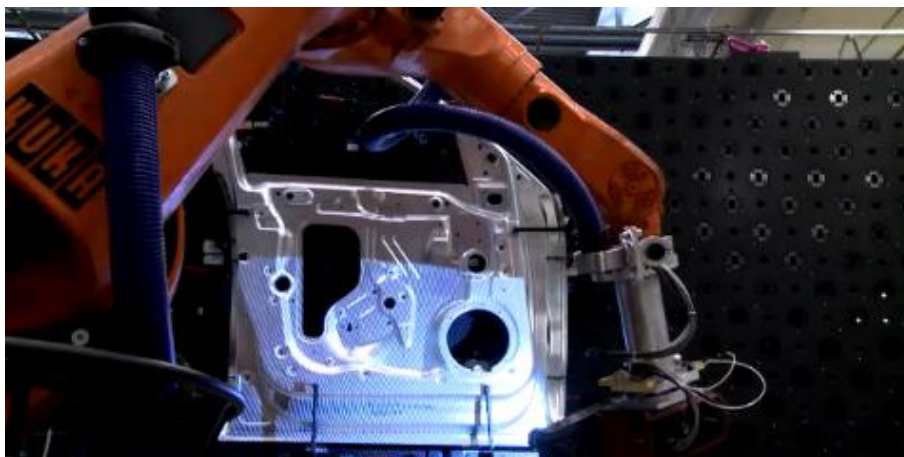


Fig. 9 Digitalização com recurso a sistemas de luz estruturada (APM Technologies)

Nas aplicações metrológicas, a área de trabalho pode variar entre 38 mm por 29 mm e os 2000 mm por 1500 mm com resoluções de 0,02 mm a 0,79 mm (APM Technologies).

2.6. MMC baseada em sistemas de ondas estacionárias

Tratam-se de sistemas que permitem a digitalização 3D recorrendo a ultrassons (Fig. 10). É uma técnica muito usada em procedimentos médicos, permitindo, por exemplo, a visualização de

fetos ou a visualização de estruturas biológicas como os vasos sanguíneos de grande calibre. Nesta técnica são projetados sons de elevada frequência, de vários ângulos e posteriormente são processados os seus ecos para construção 3D das estruturas.



Fig. 10 Digitalizador por ultrassons (The London Podiatry Centre - Scanning Technology, 2009)

Estes tipos de equipamentos não apresentam dados metrológicos nas suas especificações mas pelas suas características apresentadas, apontam para uma área de trabalho até aos 20 cm e resoluções das sondas de cerca de 0,2 mm (Healthcare, 2009).

2.7. Análise comparativa

As propriedades metrológicas de uma máquina de medir por coordenadas de três eixos objeto desta dissertação, caracterizam-se pela determinação exigente de coordenadas dos pontos tridimensionais, de contacto com a peça a medir, para que se possa mensurar a dimensão, forma e posição (Guedes, 2011). Apesar da utilização de sondas de contacto possibilitarem a deformação da peça no momento do contacto, assim como condicionar a medição de elementos de difícil acesso, tem uma grande versatilidade, exatidão e fiabilidade. Os sistemas óticos revelam-se vantajosos no que concerne a manter a integridade da peça a medir e na velocidade de aquisição metrológica, no entanto, revelam limitações na exatidão e fiabilidade da medição. Ao realizar uma comparação ao nível da resolução das várias tecnologias, os sistemas de

varrimento permitem uma maior resolução atingindo a ordem do nanómetro embora sejam mais adequados ao mapeamento do relevo dos materiais. A tecnologia de 3 eixos é a que permite obter melhor resolução para mapeamento de peças logo a seguir pelos sistemas de laser. Quanto às áreas de trabalho, os sistemas estudados são muito distintos e vão depender do objeto de medição.

Tabela 2 Tabela comparativa de tecnologias metrológicas

	Resolução	Área de trabalho
sistemas de 3 eixos	0,01 μm	poucos mm a vários metros
sistemas de varrimento	0.1 nm	alguns micrómetros
sistemas laser	5 μm	poucos mm a alguns centímetros
sistemas de luz estruturada	0,02 mm	poucos cm a alguns metros
sistemas de ondas estacionárias	0,2 mm	alguns centímetros

2.8. Controlo eletrónico

O acionamento eletrónico destes equipamentos pode ser realizado recorrendo a vários métodos. Podem-se reconhecer, soluções distintas para o controlo deste acionamento, diretamente por computador, através de um sistema baseado em microcontrolador ou utilizando um sistema embebido dedicado.

2.8.1. Acionamento direto por computador

O recurso a uma plataforma onde o computador controle diretamente os motores implica uma vantagem clara de simplificação do desenvolvimento do circuito eletrónico utilizado (Diagrama 1). Embora as portas paralelas estejam em desuso, o recurso à mesma permite o envio e receção paralela de sinais de 16 bits, sendo o suficiente para controlar vários motores em simultâneo e ao mesmo tempo receber os sinais do apalpador. Esta solução pode-se tornar inviável devido ao atraso na leitura da porta paralela pelo processador do computador. O facto de

num sistema operativo estarem a concorrer várias aplicações pela utilização do processador pode levar a uma situação em que haja perda do controlo do posicionamento.

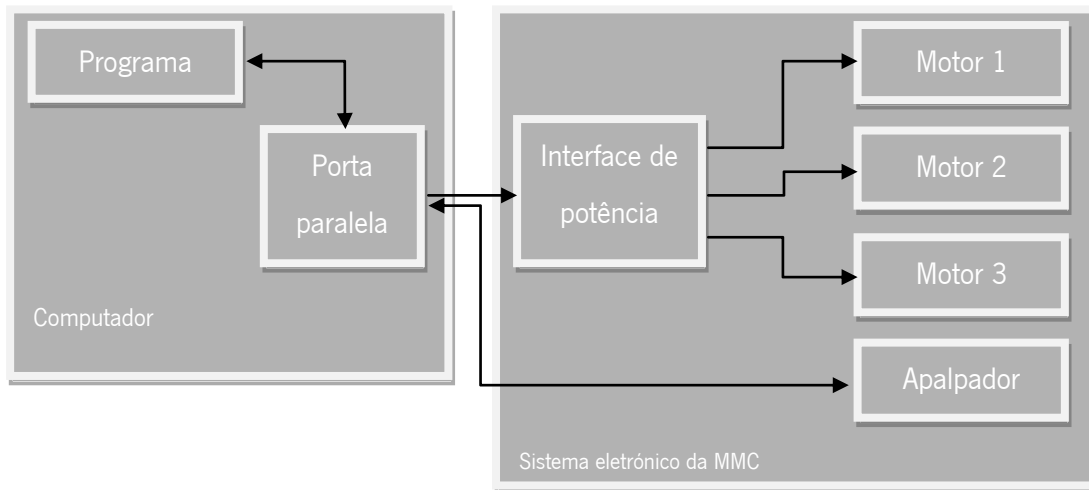


Diagrama 1 Sistema eletrônico com utilização da porta paralela.

2.8.2. Acionamento com recurso a um sistema com microcontrolador

Uma vantagem é identificada de imediato em sistemas baseados em microcontroladores (Diagrama 2), o facto de existir um equipamento dedicado ao controlo de toda a eletrónica de posicionamento. Recorrendo à programação de um microcontrolador, é possível ter um programa de computador que comunica com o microcontrolador, por exemplo, através de uma porta USB, dando instruções de movimentação gerados por algoritmos complexos de leitura e ao mesmo tempo receber um retorno do sucesso dessas mesmas instruções.

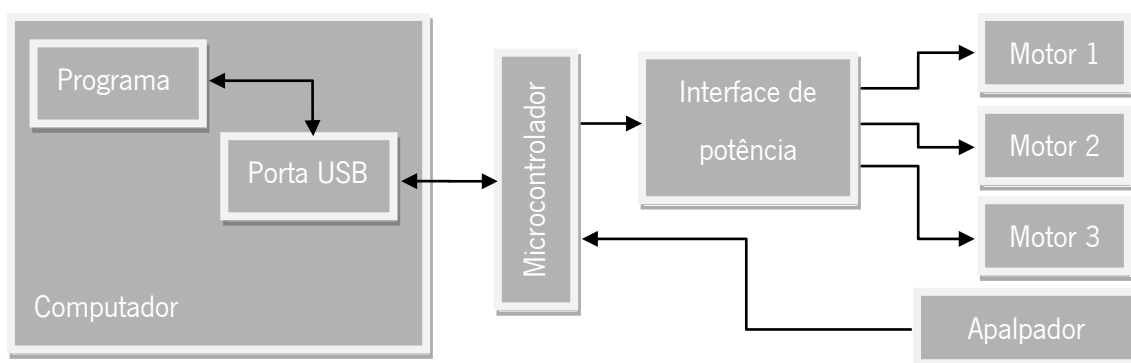


Diagrama 2 Sistema eletrônico com utilização de um microcontrolador.

2.8.3. Acionamento com recurso a um sistema embebido

A utilização de sistemas embebidos de baixo custo pode permitir a autonomia da MMC para executar algoritmos pré-programados de medição. A tecnologia existente já permite dotar uma MMC de processamento relativamente elevado e autónomo. Esta solução pode dotar a MMC de todo um sistema autónomo capaz de processar imagem, realizar medições, comparar com modelos pré-determinados e, por exemplo, numa linha de produção aferir a qualidade de um produto pelo desvio relativo a um modelo computacional. Tendo em conta a propriedade económica que se pretende para a MMC, existem disponíveis, por exemplo, dois sistemas capazes de serem utilizados para este efeito, o Raspberri Pi e Arduino DUE (Fig. 11).



Fig. 11 Placas de desenvolvimento de sistemas embebidos (Foundation, 2012) (Banzi, Cuartielles, Igoe, Martino, & Mellis, 2005)

Apesar de ser possível aplicar um sistema embebido nestas placas disponíveis por menos de 50€, estas têm características muito distintas. Na Fig. 11, à esquerda, temos a placa Raspberry Pi que comporta um processador ARM11 a 700MHz, interfaces de rede, USB, vídeo, HDMI e 17 pinos digitais e, à direita, está a placa Arduino DUE que tem um processador ARM Cortex-M3 a 84MHz com interface USB e com 54 pinos digitais e 14 analógicos.

2.8.4. Utilização de motores de passo para o deslocamento dos eixos

Uma das tecnologias mais usadas para o acionamento dos eixos das MMC baseia-se na utilização de motores de passo (Fig. 12).



Fig. 12 Motor de passo CKD existente numa MMC Roland do dep. de mecânica

Este tipo de motor tem um número específico de enrolamentos que são acionados com uma sequência própria de acordo com as necessidades. Existem dois géneros de motores, unipolares ou bipolares, onde a principal diferença se prende com o facto de nos motores bipolares se alternarem as tensões aplicadas aos enrolamentos o que vai implicar uma corrente que o atravessará em dois sentidos, enquanto nos motores unipolares esta situação não se verifica. É possível executar a excitação dos motores em modo “wavedrive”, excitando um enrolamento de cada vez, em “full step” atuando em dois enrolamentos em simultâneo, em “half-step”, que para além de atuar dois enrolamentos em simultâneo aumenta o número de passos por revolução permitindo aumentar a exatidão do movimento ou em “microstepping” que leva a um aumento da exatidão mas também à complexidade do circuito eletrónico de acionamento. Nos modos “wavedrive”, “full step” e “half-step” o acionamento recorre a uma lógica binária simples e a circuitos de elevação de sinais, por exemplo, de níveis TTL para outros níveis de tensão e corrente, como o caso da utilização do integrado UNL2003 que recorre a um conjunto de transistores *darlington* em cascata para o controlo de motores unipolares. O modo “microstepping” exige um circuito de maior complexidade, pois ao invés de existirem apenas dois níveis de tensão nos enrolamentos, podem existir vários níveis de tensão de acordo com o

movimento pretendido. Claramente se depreende que o modo “microstepping” está mais sujeito a interferências externas podendo-se desta forma afetar a repetibilidade do equipamento.

Identifica-se nestes equipamentos uma vantagem imediata, o facto de simplificar o controlo do posicionamento. Apesar de este facto beneficiar o desenvolvimento do controlo, como não existe uma forma de monitorização da sua movimentação, sem outro equipamento auxiliar, esta situação pode levar à perda de informação relativamente ao posicionamento. Este problema é intrínseco do controlo de sistemas em malha aberta. Esta perda pode ser minorada desde que se configure devidamente o binário dos motores.

2.8.5. Utilização de servomotores ou motores DC

Existem no mercado tecnologias de motores mais baratas, como o caso dos servomotores e dos motores DC. É possível recorrer a estes motores se forem incorporados nos eixos codificadores capazes de determinar o ângulo de rotação dos mesmos. Existe uma desvantagem clara relativamente aos motores de passo no caso de se pretender parar o sistema, onde se mantêm os motores sob carga. Os motores DC ao contrário dos motores de passo apesar de poderem ser parados numa determinada posição em carga, isso requer o desenvolvimento de um sistema de controlo mais complexo. Como as vibrações podem afetar mais facilmente sistemas que recorrem a estas tecnologias, será necessário também incluir esta situação no controlo eletrónico.

2.9. Potencialidades dos sistemas CAD

A complexidade da aplicação de leitura estipula o tipo de tecnologia a utilizar. Tanto se pode desenvolver uma aplicação completa para a leitura das coordenadas de uma MMC como se pode utilizar programas CAD do mercado.

O recurso ao Visual Basic para aplicações (VBA), que já vem em diversos programas CAD existentes no mercado, tem uma grande vantagem, o acesso a inúmeras potencialidades destes, sendo possível com alguma facilidade visualizar as coordenadas que se estão a obter da MMC em tempo real, fazer a aproximação dos pontos obtidos a planos, círculos e outras geometrias, entre outras capacidades típicas destes programas. No entanto, existem algumas limitações quando se deseja utilizar a visão por computador. Embora se possa desenvolver todo o programa de gestão do posicionamento em VBA, para utilizar a visão por computador é possível criar bibliotecas DLL para incorporar as plataformas de visão com os programas CAD. Desta situação advém uma dificuldade no desenvolvimento da biblioteca, o *debugging*. No entanto, é uma situação passível de ser explorada.

A criação de uma aplicação dedicada pode, por questões de simplificação, incidir apenas na aquisição prévia da geometria e posicionamento da peça e na gestão da leitura física das coordenadas da mesma pela MMC. Através da obtenção de um ficheiro com as coordenadas da peça, por exemplo em formato DWF, é possível a sua abertura pelos programas de CAD e aí explorar as suas capacidades de processamento.

2.10. Aplicação de visão por computador à MMC

O recurso à visão por computador pode permitir, por exemplo, determinar a posição e orientação da peça a medir, trazendo de imediato vantagem pela automatização do processo e ao mesmo tempo pela possibilidade de obter uma aproximação imediata da geometria da peça. Existem no mercado várias plataformas de processamento de imagem. Uma de grande relevo na comunidade científica é a OPENCV. Trata-se de uma biblioteca livre de visão por computador utilizada maioritariamente para aplicações em tempo real. Uma das características mais importantes é a sua flexibilidade pela pluralidade de sistemas que suporta. Esta plataforma reúne um conjunto vasto de algoritmos de fácil utilização que permitem realizar desde simples operações nas imagens como a binarização até operações mais complexas como a visão *stereo*. Esta plataforma permite ainda ser utilizada por várias ferramentas de várias linguagens de programação.

2.10.1. Modelos de visão por computador

Antevêem-se duas situações distintas, quando se conhece a geometria da peça ou quando essa informação não existe. Em sistemas de controlo dimensional, nomeadamente, no fabrico de peças, a geometria da peça é conhecida podendo ser usado um dos sistemas mais comuns reconhecimento de objetos, o sistema de reconhecimento baseado em modelo.

Este sistema baseia-se na utilização de um sistema sensorial (câmara laser, ultra sónica, infravermelhos, ccd, etc.) utilizado para obter informações de uma zona de interesse. Estas imagens captadas são posteriormente processadas para a obtenção de características das peças tais como vértices, cantos, linhas ou texturas. De acordo com as características existentes numa base de dados de desenhos geométricos, obtém-se um grupo de correspondência. Através de um algoritmo é feita a renderização do desenho geométrico e a comparação com a imagem captada procurando obter a melhor correspondência. A partir deste momento é possível determinar para a peça, a posição, orientação, área, perímetro, orifícios (número, tamanho e posição), circularidade, planicidade, entre outras especificações (Campos, 2010).

Quando se pretende realizar engenharia inversa, nomeadamente, através de um objeto, determinar a sua geometria e possíveis restrições geométricas, perfilam-se duas opções, a visão estéreo ou a utilização de um sistema de deteção geométrica por profundidade.

2.10.2. Visão por infravermelhos

Uma das possibilidades para a aquisição prévia da geometria das peças através da visão por computador pode estar nas novas tecnologias das consolas de vídeo como a *Kinetic* da *Xbox360* da empresa Microsoft. Trata-se de um integrado criado pela empresa CANESTA baseado numa tecnologia que recorre à medição do tempo que a luz infravermelha leva do objeto ao sensor obtendo a distância exata do mesmo. Esta solução pode ser utilizada para a aquisição prévia da geometria da peça. Através de uma iluminação infravermelha aplicada à zona de enquadramento da peça, é possível adquirir a imagem por uma *webcam* para processamento. Existem diversas câmaras de baixo custo que fazem a aquisição de imagens também neste espectro de luz,

sendo apenas necessário recorrer a um filtro. Este filtro pode ser realizado recorrendo aos rolos fotográficos de tecnologias antigas de fotografia. Desta forma, este filtro ao permitir a passagem apenas da luz infravermelha é obtida uma imagem em tons de cinzento cujo tom indica a distância à peça. Com algum processamento de imagem adquire-se a geometria da peça.

2.10.3. Visão estéreo

Um dos campos de exploração é a visão estéreo por computador. Neste campo, é necessário o recurso a duas câmaras que visualizam a peça de ângulos diferentes e que iram permitir obter profundidade de imagem como resultado final do sistema. Através de algoritmos de tratamento de imagem é possível obter a geometria das peças e o seu posicionamento na MMC.

2.11. Algoritmos de medição

Existem formas distintas de abordagem na medição, conforme o que se pretende medir. Quando se utiliza a visão por computador para determinar a localização da peça, posteriormente podem-se aplicar algoritmos conforme a geometria previamente lida. Por exemplo, no caso de se detetar um furo numa peça e se pretenda medir a circularidade do mesmo (Fig. 13), a distância entre dois círculos concêntricos que enquadram a linha circunferencial extraída, após a obtenção das coordenadas é necessário recorrer a métodos numéricos para aferir o resultado.



Fig. 13 Método de obtenção da circularidade de uma peça (Mitutoyo America Corporation)

O método usado tem uma influência direta no resultado obtido. Têm sido adotados vários métodos para a obtenção da mensuranda, como o método dos mínimos quadrados ou Gaussiano, utilizado em qualquer geometria (Barczak, 1996). A norma de especificações geométricas de produtos ISO 12181-1:2011 define os termos e os conceitos relacionados com as características integrais individuais de circularidade e todos os seus conceitos. É então determinante para cada especificidade tais como a planeza, circularidade, cilindridade, perpendicularidade, concentricidade e outras que se pretenda obter, verificar o método especificado pelas normas internacionais.

Outra das soluções é a obtenção dos dados metrológicos de uma peça através de uma nuvem de pontos obtida por varrimento. Esta solução permite até ser pré-programada para que autonomamente um microcontrolador possa obter os pontos de contacto no volume de ação da MMC e gravar em cartão de memória. Ao aplicar algoritmos específicos é possível através de nuvem de pontos obter os dados metrológicos pretendidos. Este método é muito lento requerendo muito tempo para obter a nuvem de pontos sendo ainda necessário fazer o seu processamento.

3. Desenvolvimento de uma MMC

Nesta fase procurou-se adotar uma metodologia de separação funcional, procurando também colocar todas as relações sob a forma de diagramas. Para além de tornar mais fácil a consciencialização à volta de uma MMC, torna este conhecimento mais duradouro já que fica independente da inovação tecnológica de determinados domínios (Pruvot, 1993).

Estabeleceu-se como principais requisitos para o desenvolvimento da MMC para o campo pedagógico:

- a utilização de materiais onde sejam fácil o corte, a furação, a limagem, etc, para que possa ser construída pelos ciclos básicos de educação;
- a facilidade de obtenção dos materiais recorrendo sempre que possível a materiais adaptados de equipamentos em fim de vida ou disponível num ponto de venda próximo;
- a facilidade da construção utilizando equipamentos básicos de uma oficina de mecânica e sem utilizar de processos de soldadura mecânica;
- a simplicidade técnica, utilizando a mesma forma de movimentação nos 3 eixos;
- a redução do número de peças utilizadas; a obtenção de uma resolução na ordem do décimo de milímetro;
- uma área de trabalho até algumas dezenas de centímetros;
- a utilização de uma eletrónica básica para todo o sistema que permitam a sua montagem pelos alunos do ensino básico e alguma adaptação no ciclos superiores;
- o recurso a plataformas livres para o desenvolvimento eletrónico e informático do processo de medição;
- a possibilidade de utilizar um processo manual ou um processo automático de medição;
- a obtenção prévia da localização da peça a medir através a inclusão de um processamento de visão por computador, permitindo cativar os alunos dos ciclos superiores de educação;
- a utilização de um sistema CAD para realizar o processo de medição, ficando assim disponível a utilização de várias ferramentas desses sistemas;

Assim, os sistemas de medir por coordenadas, baseados em 3 eixos são geralmente constituídos por sistemas do tipo pórtico. Desenvolveu-se um sistema baseado na variante de pórtico fixo com base móvel pela facilidade de montagem, permitindo utilizar a mesma forma de construção de movimentação dos eixos. Este sistema tem base móvel com eixo OX e pórtico fixo com eixo OY e braço móvel com eixo OZ (Fig. 14).

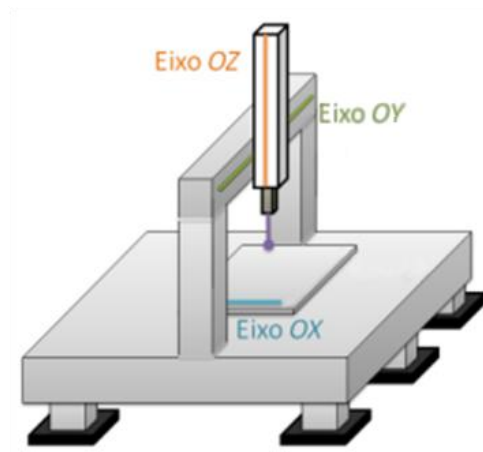


Fig. 14 Sequência cinemática de uma MMC de 3 eixos (Fernandes, 2010)

3.1. Componentes estruturais

Os componentes de uma MMC podem ser esquematizados tendo em conta a relação entre os mesmos. No dispositivo desenvolvido o pórtico está fixo à base. O eixo OX está fixo na base, o eixo OY foi fixo no pórtico e o eixo OZ foi fixo no Eixo Y. O sistema de apalpação foi fixo ao eixo OZ.

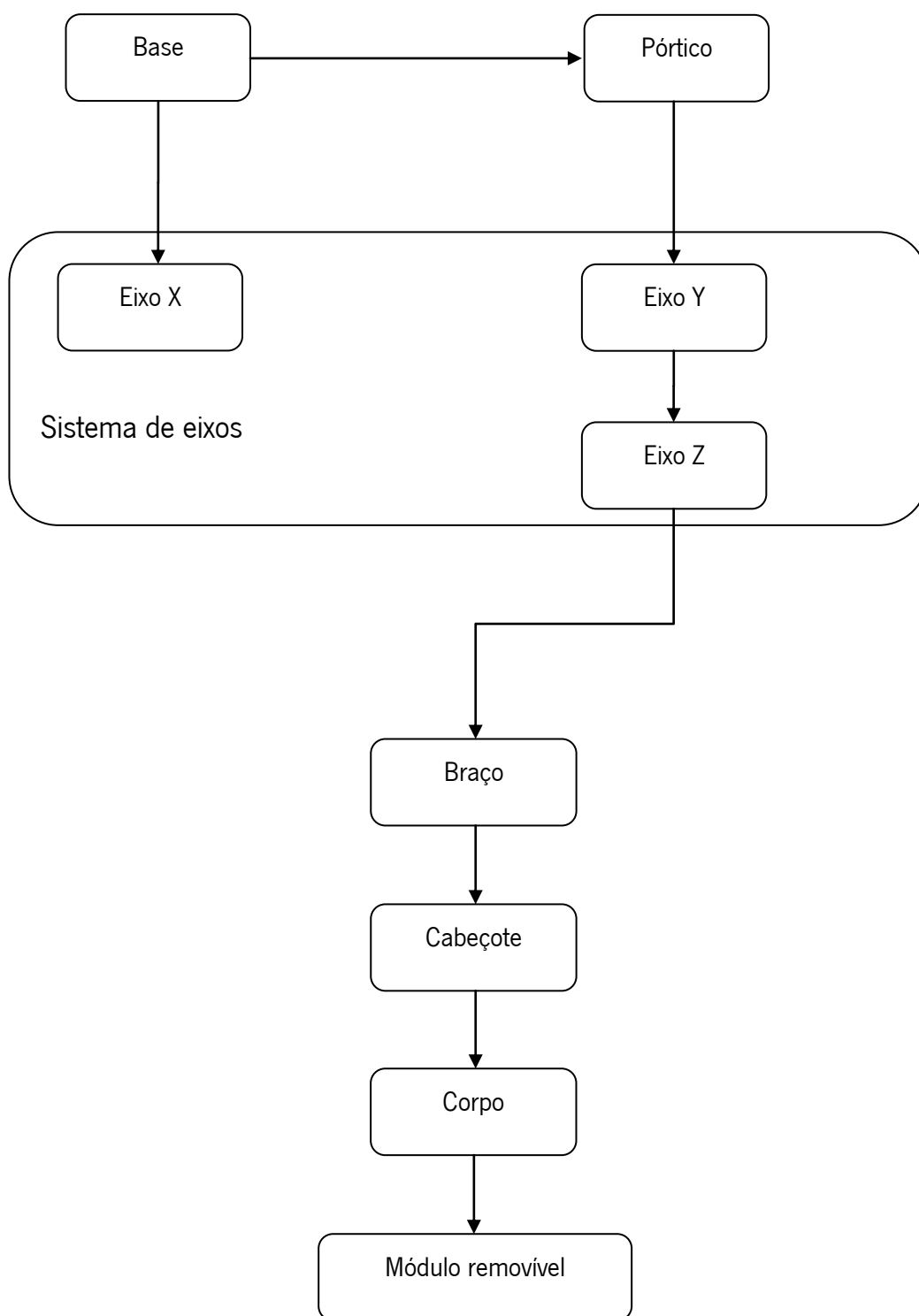


Diagrama 3 Estrutura da MMC

3.1.1. Base e Pórtico

Para os elementos estruturais relativos à base e ao pórtico, tendo em conta a facilidade de encontrar matéria-prima para a sua construção, optou-se por componentes em alumínio. Para além das qualidades já reconhecidas para a construção da MMC, o alumínio revelou-se uma ótima escolha no que diz respeito à facilidade de trabalhar o mesmo.

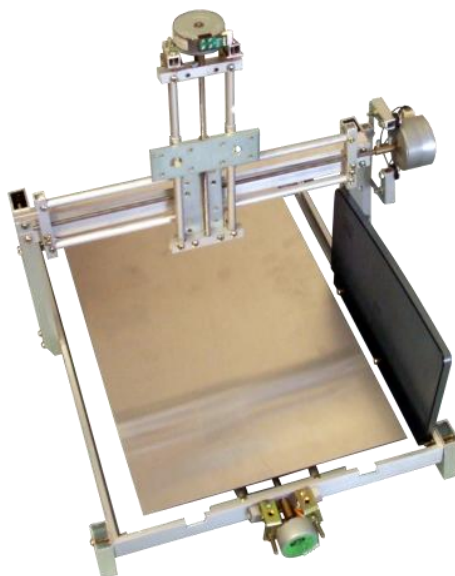


Fig. 15 MMC desenvolvida em alumínio com base e pórtico móvel

Objetivando a utilização primária de elementos em alumínio, partiu-se de uma moldura em alumínio para a base da MMC (Fig. 15). Já o pórtico foi construído a partir de perfis retangulares M12 em alumínio bruto.

3.1.2. Eixos

Para realizar o deslocamento dos eixos, ao invés de utilizar um sistema de fusos, recorreu-se a uma solução mais barata e fácil de encontrar, embora em termos de implementação não seja a mais indicada devido ao atrito e à menor longevidade. Foram utilizados varões roscados em aço inox M5 com 0,8mm de passo em conjunto com uma fêmea sextavada em aço inox M5 (Fig. 16).

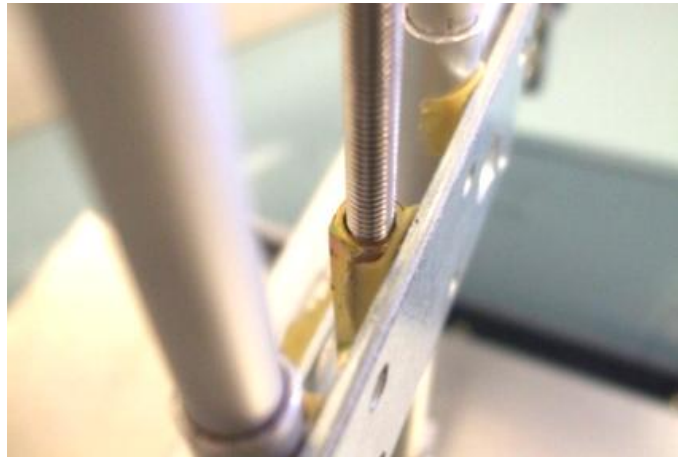


Fig. 16 Sistema de deslocamento dos eixos

Para minimizar as folgas no sistema de deslocamento dos eixos, utilizou-se um sistema de corredeiras cilíndricas recorrendo a perfis de barra redonda de $\varnothing = 8\text{mm}$ em alumínio maciço com tubos redondos 8x1mm. Esta escolha surgiu pela facilidade de encontrar este tipo de solução, embora a aplicação de perfis em alumínio imponham um grande atrito nos eixos. Uma melhor solução futura passa pela sua substituição por perfis de aço maciço com tubos em aço temperado ou teflon¹. O teflon tem uma característica que o torna impreterível para esta aplicação. É o terceiro material com mais baixo coeficiente de atrito de todos os sólidos conhecidos. Os perfis em aço maciço podem ser facilmente encontrados em impressoras em desuso sendo apenas necessário encontrar os tubos em teflon para obter um sistema com maior durabilidade. Para fixar o deslocador do eixo, aplicou-se uma resina epóxi (Fig. 17) numa placa perfurada em aço zincado de 100x35x3mm contra os tubos redondos e a fêmea sextavada. Esta placa perfurada fará o lugar do cabeçote da MMC.

¹ TEFLON - marca comercial registada pela empresa DuPont cujo material está classificado pela União Internacional de Química Pura e Aplicada como PTFE (Politetrafluoretileno)

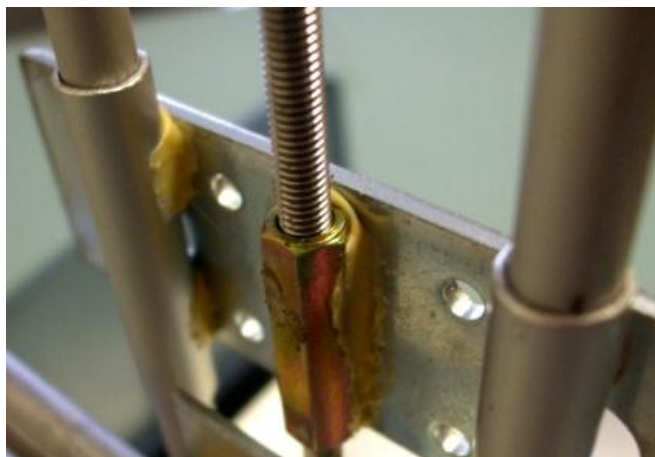


Fig. 17 Aplicação da resina epóxi

Para fixar os tubos maciços foram furados perfis em alumínio quadrados 10x10x1mm. Seguidamente foram fixados com parafusos tipo Philips, M4, L = 17,5mm, $d_k = 7\text{mm}$, $k = 2,6\text{mm}$ em conformidade com a norma ISO7048 (parafuso de cabeça cilíndrica com oco cruciforme - C) (Fig. 18).

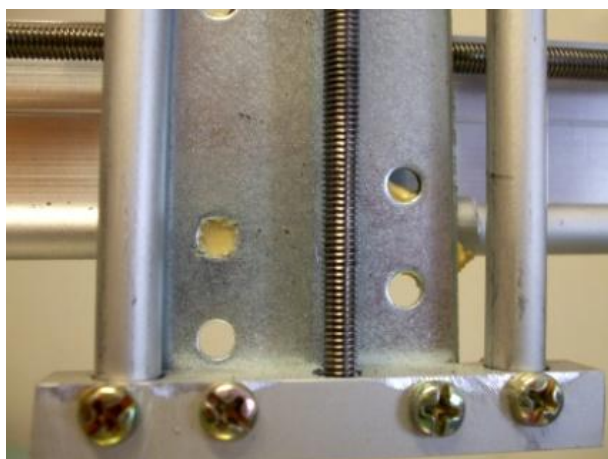


Fig. 18 Fixação com parafuso dos eixos de deslocamento

Para reduzir o número de peças na construção da MMC, foi reduzida a utilização de parafusos ao tipo Philips, M4 (norma ISO7048) com as variantes de comprimento de 17,5mm, 20mm e 30mm.

3.1.3. Apalpador

Tendo a noção de que este elemento deve ser escolhido de acordo o tipo de medição que se pretende efetuar, mas querendo alcançar a facilidade de construção da MMC para o âmbito pedagógico, optou-se pela utilização de um sistema de apalpação recorrendo a uma ponta em bico. As pontas em forma de bico (Fig. 19) não devem ser usadas como sondas XY pois foram criadas para estudos de relevo de peças, determinação de pontos específicos e linhas. Este tipo de pontas é de elevada precisão e necessita de uma maior configuração para realizar medições. Pode também ser usada para inspecionar o local de furos pequenos (Fernandes, 2010).

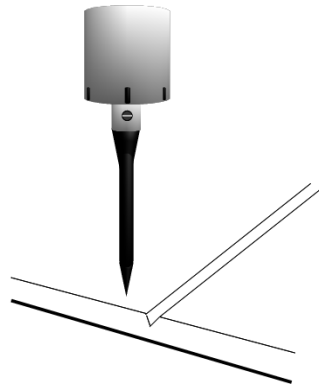


Fig. 19 Apalpador de ponta em bico (Fernandes, 2010)

Apesar da ponta em bico não ser indicada na utilização como sonda XY, esta encontra-se em vários equipamentos profissionais, como a Roland MDX20, tendo sido a escolhida pela sua facilidade de construção. Recorreu-se a uma vulgar agulha em aço soldada num sensor piezoelétrico (Fig. 20), sistema também usado na Roland MDX20.

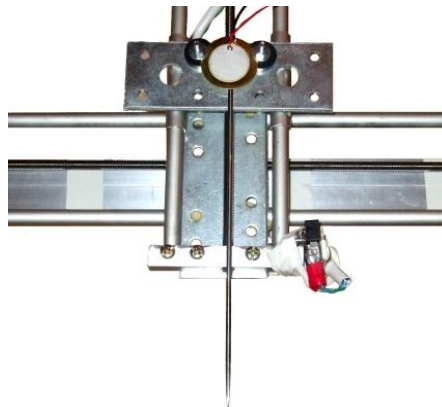


Fig. 20 Apalpador construído com recurso a uma agulha em aço

3.2. Cinemática direta

Neste âmbito, procurou-se apenas verificar a possibilidade de ser analisada a cinemática direta nestes sistemas, baseando-se o estudo num dos sistemas mais comuns, base fixa e pórtico móvel, identificando-se três eixos principais, OX, OY e OZ. Todos estes eixos são de junta colinear. Nas várias configurações possíveis do cabeçote, existem outros tipos de juntas. Considerando a configuração da Fig. 21 podemos melhorar o desempenho da MMC aplicando num cabeçote um eixo de junta colinear e um eixo de junta ortogonal.

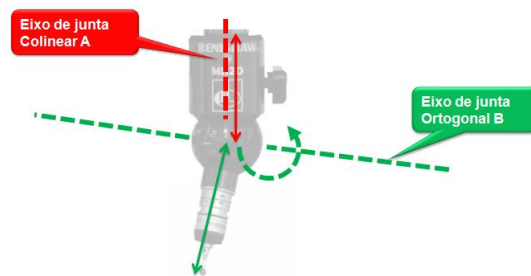


Fig. 21 Eixos do cabeçote de uma possível configuração (Renishaw, 2011)

A cinemática direta de um sistema de 3 eixos lineares é pouco relevante, mas se forem introduzidas pontas de medição com juntas ortogonais já é necessário recorrer a métodos de cálculo da cinemática. Neste ponto realiza-se o estudo da cinemática direta de um sistema de 3 eixos aplicando uma ponta de medição com uma junta ortogonal, servindo de base a qualquer outro tipo de ponta a utilizar (Fig. 22). Uma das formas de calcular a cinemática direta é recorrendo ao método de Denavit-Hartenberg (Hartenberg & Denavit, 1964).

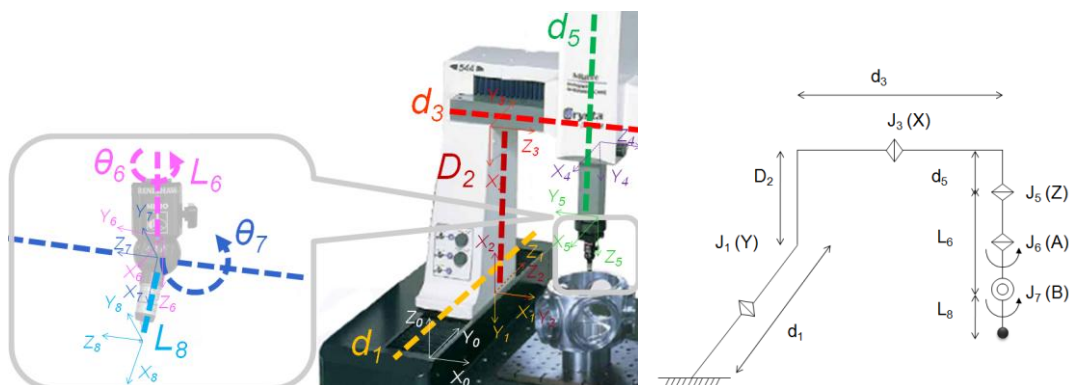


Fig. 22 Objeto de aplicação do método de Denavit-Hartenberg e respetiva estrutura das juntas (Renishaw, 2011)

Com a identificação das juntas e dos respectivos deslocamentos lineares e angulares, retiraram-se os parâmetros de Denavit-Hartenberg.

Tabela 3 Parâmetros de Denavit-Hartenberg

elos	Variável	a	α	d	θ	
{0}->{1}	d_1	0	-90°	$Z_0 \perp Z_1$	d_1	0°
{1}->{2}		0	0°	$Z_1 \parallel Z_2$	0	-90°
{2}->{3}	D_2	D_2	-90°	$Z_2 \perp Z_3$	0	180°
{3}->{4}	d_3	0	0°	$Z_3 \parallel Z_4$	d_3	-90°
{4}->{5}	d_5	0	-90°	$Z_4 \perp Z_5$	d_5	0°
{5}->{6}	L_6, θ_6	0	0°	$Z_5 \parallel Z_6$	L_6	θ_6°
{6}->{7}	θ_7	0	-90°	$Z_6 \perp Z_7$	0	$\theta_7^\circ - 90^\circ$
{7}->{8}	L_8	L_8	0°	$Z_7 \parallel Z_8$	0	0°

Após a obtenção dos parâmetros de Denavit-Hartenberg, deduziu-se cada matriz de transformação homogênea.

$$\begin{aligned}
 {}^0T_{1h} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_h &
 {}^1T_{2h} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_h &
 {}^2T_{3h} &= \begin{bmatrix} -1 & 0 & 0 & D_2 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_h \\
 {}^3T_{4h} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}_h &
 {}^4T_{5h} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_h &
 {}^5T_{6h} &= \begin{bmatrix} \cos \theta_6 & -\text{sen} \theta_6 & 0 & 0 \\ \text{sen} \theta_6 & \cos \theta_6 & 0 & 0 \\ 0 & 0 & 1 & L_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}_h \\
 {}^6T_{7h} &= \begin{bmatrix} \text{sen} \theta_7 & -\cos \theta_7 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \cos \theta_7 & -\text{sen} \theta_7 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_h &
 {}^7T_{8h} &= \begin{bmatrix} 1 & 0 & 0 & L_8 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_h
 \end{aligned}$$

Equação 1 Matrizes de transformação homogêneas

A transformação homogénea que estabelece a relação entre os referenciais 0 e 8 é obtida pela aplicação das sucessivas transformações.

$${}^0T_{8h} = {}^8T_{7h} \cdot {}^7T_{6h} \cdot {}^6T_{5h} \cdot {}^5T_{4h} \cdot {}^4T_{3h} \cdot {}^3T_{2h} \cdot {}^2T_{1h} \cdot {}^1T_{0h}$$

$${}^0T_{8h} = \begin{bmatrix} -\text{sen}\theta_6 \cdot \text{sen}\theta_7 & \text{sen}\theta_6 \cdot \cos\theta_7 & -\cos\theta_6 & -L_8 \cdot \text{sen}\theta_6 \cdot \text{sen}\theta_7 + d_3 \\ \cos\theta_6 \cdot \text{sen}\theta_7 & \cos\theta_6 \cdot \cos\theta_7 & \text{sen}\theta_6 & L_8 \cdot \cos\theta_6 \cdot \text{sen}\theta_7 + d_1 \\ -\cos\theta_7 & -\text{sen}\theta_7 & 0 & -L_8 \cdot \cos\theta_7 + D_2 - d_5 - L_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}_h$$

Equação 2 Matriz de transformação resultante relativa à MMC

Para a obtenção da geometria de um objeto numa MMC, é necessário criar um referencial do próprio objeto (Fig. 23).



Fig. 23 Introdução do referencial do objeto (Renishaw, 2011)

Isto vai introduzir mais uma transformação, do referencial da MMC para o referencial do objeto.

$${}^{-1}T_{8h} = {}^{-1}T_{0h} \cdot {}^0T_{8h}$$

$${}^{-1}T_{8h} = \begin{bmatrix} -\text{sen}\theta_6 \cdot \text{sen}\theta_7 & \text{sen}\theta_6 \cdot \cos\theta_7 & -\cos\theta_6 & -L_8 \cdot \text{sen}\theta_6 \cdot \text{sen}\theta_7 + d_3 - p_1 \\ \cos\theta_6 \cdot \text{sen}\theta_7 & \cos\theta_6 \cdot \cos\theta_7 & \text{sen}\theta_6 & L_8 \cdot \cos\theta_6 \cdot \text{sen}\theta_7 + d_1 - p_2 \\ -\cos\theta_7 & -\text{sen}\theta_7 & 0 & -L_8 \cdot \cos\theta_7 + D_2 - d_5 - L_6 - p_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}_h$$

Equação 3 Matriz de transformação resultante relativa ao objeto a medir

O desenvolvimento de sistemas de *hardware* eletrônico *opensource* é uma atividade recente que tem obtido muitos adeptos nos últimos anos. A aplicação de equipamento de acionamento eletrônico recorrendo a *hardware opensource* traz mais-valias quando se pretende implementar uma solução de baixo custo passível de ser melhorada livremente. Na implementação analisada no ponto 3.3 o compilador utilizado para desenvolver o programa era gratuito mas estava limitado a 2kbytes de memória ocupada no microcontrolador. Para utilizar toda a capacidade do mesmo era necessário adquirir uma licença proprietária. Para contornar estas limitações, recorreu-se a um sistema de desenvolvimento de protótipos *opensource*, o Arduino (ANEXO 2) (Fig. 25) (Banzi, Cuartielles, Igoe, Martino, & Mellis, 2005).

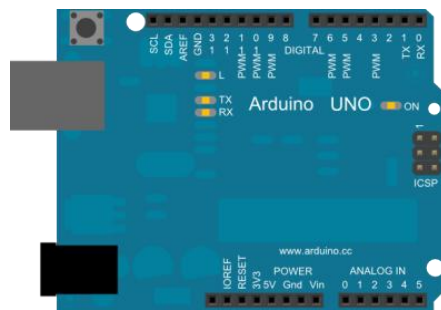


Fig. 25 Placa de desenvolvimento Arduino Uno (Banzi, Cuartielles, Igoe, Martino, & Mellis, 2005)

Este sistema tem a vantagem de incluir um sistema de desenvolvimento totalmente *opensource*, quer no que diz respeito ao *hardware* quer ao *software*. As placas de desenvolvimento podem ser facilmente montadas ou adquiridas em várias lojas. O *software* de programação é facilmente descarregável a partir da página oficial do projeto na Internet (Fig. 26).

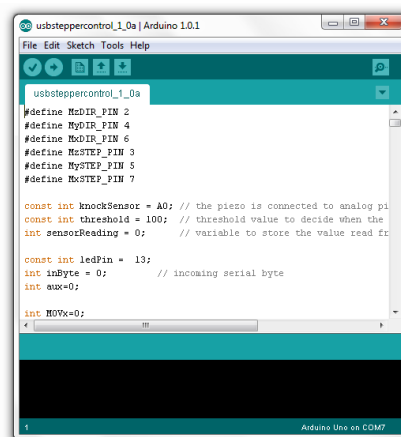


Fig. 26 Ambiente de programação Arduino (Banzi, Cuartielles, Igoe, Martino, & Mellis, 2005)

3.3.1. Controlo do motor

A utilização de uma plataforma livre de desenvolvimento, levou à necessidade de procurar uma forma de controlar os motores onde a liberdade de utilização e de expansão se mantivesse. Desta forma, foram aplicados na MMC, motores bipolares controlados a partir de placas *opensource* EasyDrive (ANEXO 3) (Fig. 27) (Schmalz, nd.). Estas placas de controlo têm inúmeras vantagens relativamente a outros sistemas.

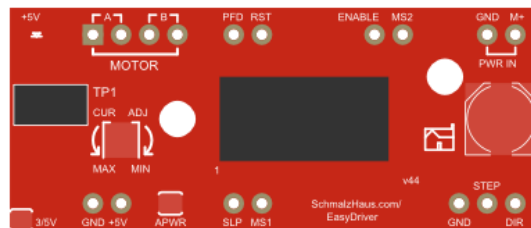


Fig. 27 Placa de controlo de motores EasyDrive (Schmalz, nd.)

Desde já têm todo o desenvolvimento em plataforma livre o que não condiciona um futuro melhoramento do sistema. Permitem a aplicação de motores até 30 V independentemente da sua potência, já que ficam limitados eletronicamente a 750mA, o que possibilita o recurso a motores usados de inúmeras impressoras. A placa é baseada no controlador Allegro A3967 permitindo aplicar uma corrente variável de 150 mA a 750 mA aos enrolamentos dos motores a partir de um regulador existente na placa. Esta situação permite colocar ao mesmo tempo em funcionamento, motores de diferentes níveis de tensão nominal. O binário dos motores foi ajustado (Fig. 28) para que no conjunto não se perdessem passos, já que foi implementado um sistema de controlo em malha aberta sem incluir meios de retorno do posicionamento. Este facto foi devido à sua facilidade de controlo/comando, mas fica em aberto a utilização codificadores de eixos capazes de permitir controlar uma possível a perda de passos.

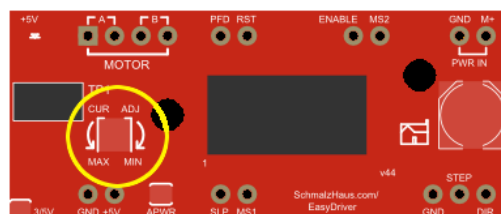


Fig. 28 Regulador de corrente - EasyDrive (Schmalz, nd.)

Por defeito, esta placa vem com um modo de *microstepping* de 8 passos. Esta solução tem vantagem em termos de obtenção de uma melhor exatidão da MMC com, por exemplo, um máximo de 1600 passos por revolução para um motor de 200 passos por revolução. O sistema de *microstepping* pode ser facilmente ajustado pelos terminais MS1 e MS2 conforme a tabela 1.

Tabela 4 Ajustamento do *microstepping* na placa EasyDrive

MS1	MS2	Resolução
Nível baixo	Nível baixo	Full step (2 fases)
Nível alto	Nível baixo	Microstepping - 2 passos
Nível baixo	Nível alto	Microstepping - 4 passos
Nível alto	Nível alto	Microstepping - 8 passos

3.3.2. Controlo do mecanismo de medição

Os vários sistemas de analisados recaem na forma de controlo do apalpador recorrendo a sensores piezoelétricos resistivos. Foi testado um conjunto de agulha em aço soldada num sensor piezoelétrico em latão de 20 mm com tensão nominal de funcionamento entre 3 e 12 Volt e uma impedância máxima de 1 k Ω . Utilizou-se a entrada analógica A0, de 10 bits de resolução, e uma resistência de 1M Ω em paralelo com o sensor para realizar a leitura de valores (Fig. 29).

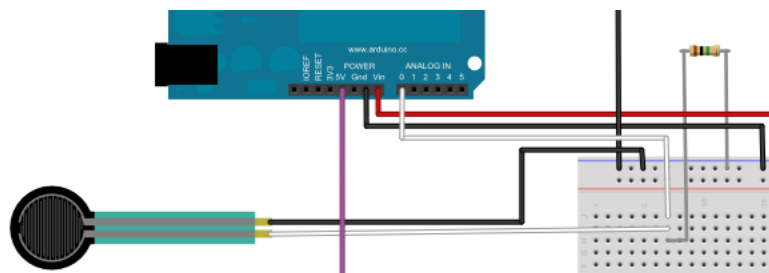


Fig. 29 Ligação do sensor piezoelétrico ao Arduino Uno

Devido à vibração causada pelos motores, denotou-se uma variação elevada de valores não sendo possível determinar quando o apalpador detetava a peça a medir. Esta solução deve ser futuramente aprofundada já que é a mais utilizada nos sistemas atuais como a ROLAND MDX20.

Foi implementada outra solução de monitorização da apalpação. Como a mesa de medição estava isolada eletricamente do resto da MMC, colocou-se na mesma uma ligação equipotencial (GND) e aplicou-se uma tensão de 5 Volt na agulha (Fig. 20). Esta forma de monitorização é válido para peças metálicas já que funciona como um contacto elétrico (Fig. 30).

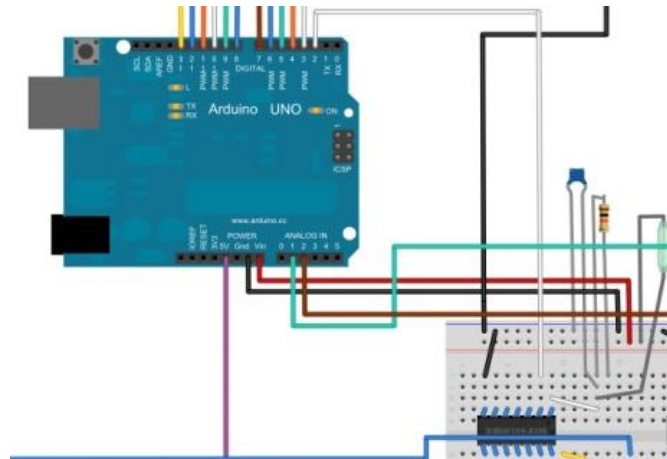


Fig. 30 Ligação da agulha ao Arduino Uno

A limitação deste método dá-se com o facto da peça a medir ter que permitir a passagem de corrente elétrica. Este sistema de contacto tem ainda um efeito de Ripple (Fig. 31) que foi ultrapassado da mesma forma que o controlo de fim de curso dos eixos apresentada no próximo ponto.

3.3.3. Controlo do fim de curso dos eixos

O descolamento dos eixos foi controlado com recurso a interruptores de fim de curso. Estes foram montados em cada um dos três eixos permitindo sinalizar o fim de curso do eixo, quer mínimo quer máximo. Devido à característica intrínseca do contacto dos fim de curso que cria variações de tensão (Ripple, Fig. 31) no momento do contacto, não é possível gerar corretamente uma interrupção única no microcontrolador.

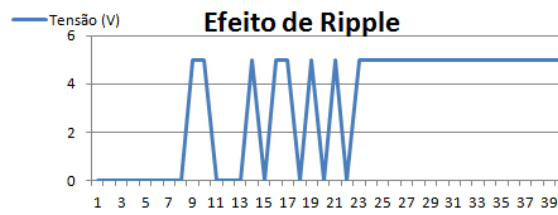


Fig. 31 Efeito de Ripple nos fim de curso

Contornou-se este efeito através da aplicação de um circuito RC série com inversor de Smith Trigger da referência HFC40106BE (Fig. 32).

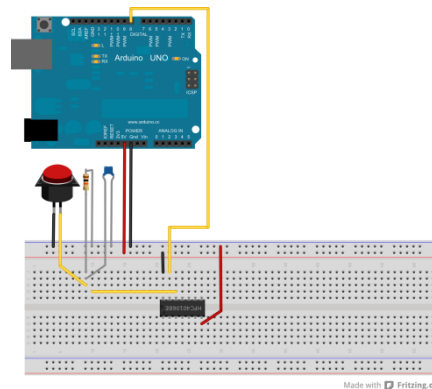


Fig. 32 Circuito de controlo dos fins de curso

O circuito RC série cria um efeito de carga e descarga que depois é negado pelo inversor (Fig. 33).

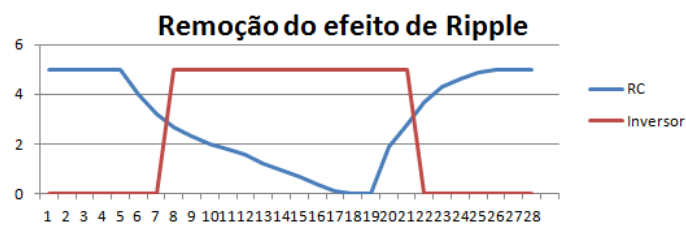


Fig. 33 Remoção do efeito de Ripple

Desta forma, é possível remover o efeito de Ripple e permitir a utilização de interrupções na plataforma Arduino.

Uma das limitações da plataforma baseada no Arduino Uno, prende-se com o facto de este estar resumido à utilização de 2 interrupções externas. Existe a necessidade de realizar operações diferentes quando os micros interruptores de fim de curso dos eixos são acionados. Como temos o sistema de apalpação a ocupar o Pino Digital 1, apenas resta o Pino Digital 2 para controlar o acionamento de todos os fins de curso. Foi encontrada uma solução baseada em lógica com o circuito integrado HEF4072BP. Trata-se de uma porta lógica OR dupla de quatro entradas. A utilização desta porta lógica permitiu realizar um OR entre todos os fins de curso.

3.3.4. Visão geral do sistema eletrónico implementado

O sistema implementado (Fig. 34), como já foi referido, recorre a partes *opensource* e alguns componentes de fácil acesso.

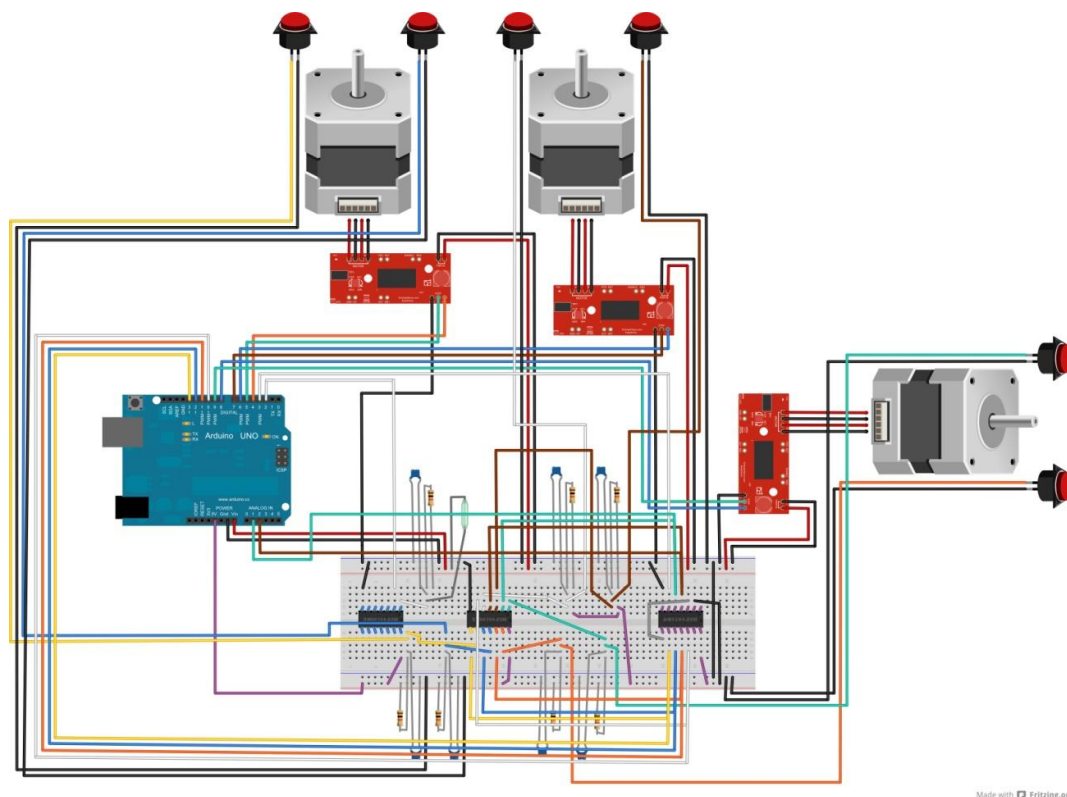


Fig. 34 Circuito completo do sistema implementado

3.4. Programação

Ao nível de desenvolvimento de programação, a MMC requereu a programação do microcontrolador e ao mesmo tempo a criação de uma DDL capaz de integrar a visão por computador com a ferramenta de modelação *Inventor*, da *Autodesk*.

3.4.1. Programação do microcontrolador

Relativamente ao Arduino, recorreu-se ao ambiente próprio (Fig. 26) disponível para o mesmo. Este ambiente já tem disponível um conjunto de soluções para interagir com o Arduino, nomeadamente a comunicação USB, que na prática funciona com um mecanismo de porta série (Fig. 35).

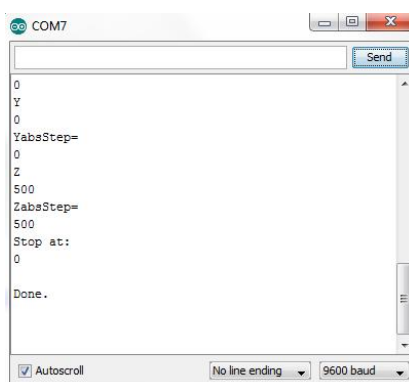


Fig. 35 Comunicação série com o Arduino

Foi desenvolvido o programa que consta no ANEXO 4 que permite o envio de um G - Code adaptado com a seguinte terminologia:

G00

Indica à MMC o deslocamento em passos, à velocidade programada num eixo de cada vez. É programado à velocidade máxima. É necessário ter em conta a falta de binário que pode suceder

e ao mesmo tempo uma menor repetibilidade devido à perda de passos (controlo em malha aberta).

X, Y e Z

Indica à MMC que virá o deslocamento nessa direção

+

Indicação do sentido

00000

Indicação do deslocamento

Como exemplo, um deslocamento de 200 passos em X e -3000 em Y será solicitado à MMC com o seguinte código:

G00X+00200Y-03000Z+00000

No algoritmo (Diagrama 4) que está na base do processamento da MMC (ANEXO 4), foi necessário implementar a transformação do texto recebido pela porta série em variáveis de valor matemático para o respetivo acionamento dos motores de passo. Foi também implementada uma rotina de inicialização do sistema onde todos os eixos são levados ao ponto (0,0,1000), em passos. Os eixos X e Y ficam na coordenada 0 embora na prática esta seja 1000 passos afastada do fim de curso. O eixo Z fica inicialmente na coordenada 1000 devido à distância obrigatória do apalpador ao plano XY, para que fique sem contacto no apalpador.

Toda a comunicação com o Arduino é baseada numa trama de 24 caracteres de texto, tendo o controlo da sua integridade realizada pela resposta da mesma trama pelo Arduino.

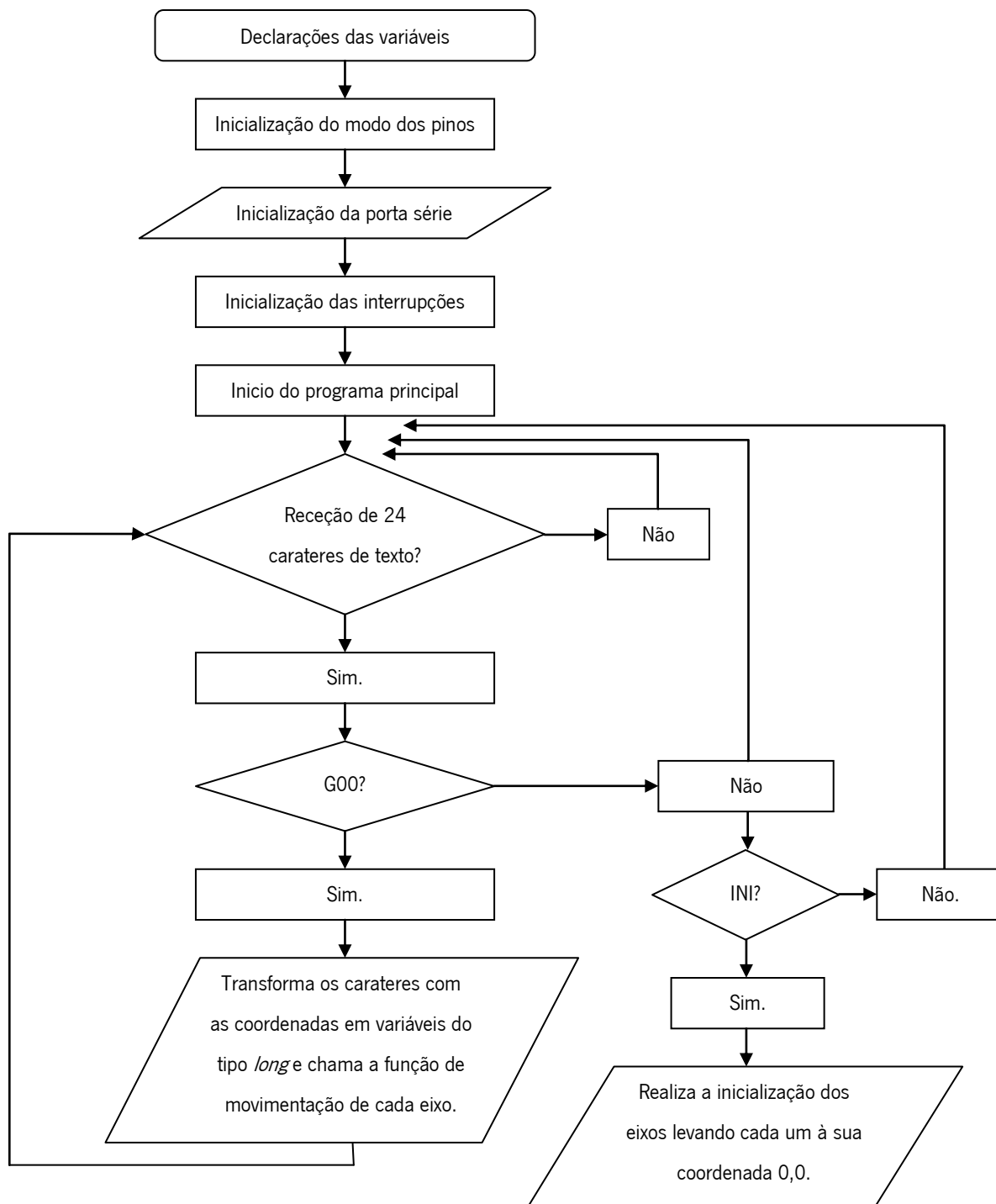


Diagrama 4 Algoritmo principal de funcionamento do Arduino

O processamento das interrupções requereu a utilização obrigatória de variáveis do tipo volátil que são alteradas aquando da respetiva interrupção. Às interrupções temos associadas duas variáveis, uma para a existência de um sinal do apalpador e outra para assinalar o fim de curso

dos eixos. Esta mudança nas variáveis das interrupções gera um movimento contrário ao que está a suceder, em 1000 passos, que foi o estabelecido para o afastamento, quer da peça a medir (no caso do apalpador), quer dos interruptores de fim de curso. Esta situação só ocorre quando se existe movimento, ou seja, na execução das funções de movimentação dos eixos (Diagrama 5).

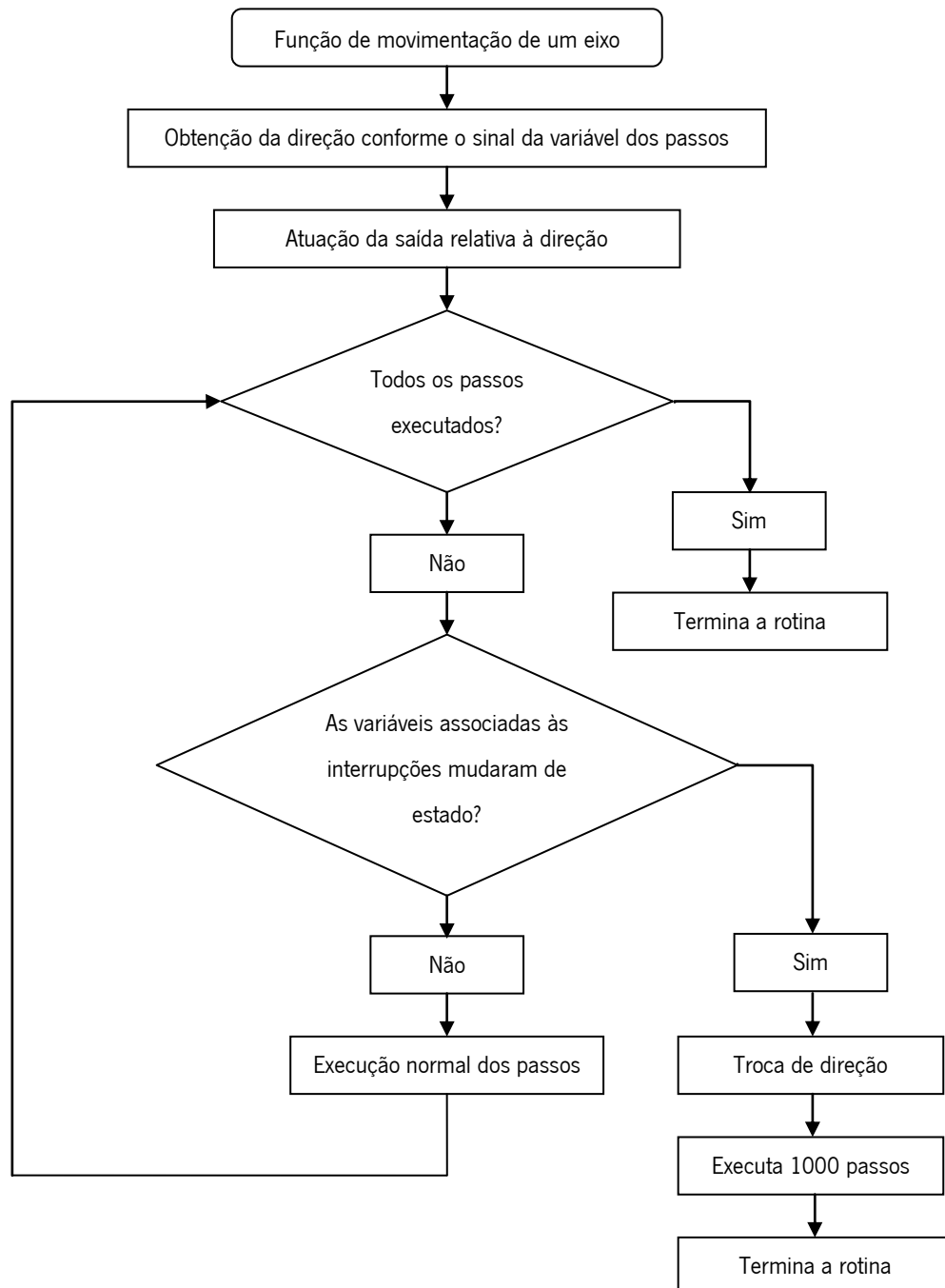


Diagrama 5 Rotina do Arduino de movimentação de um eixo

3.4.2. Programação da aplicação de visão por computador

Uma das formas de acelerar o processo de leitura de coordenadas metrológicas da MMC é a obtenção antecipada da localização da peça. Após a obtenção mais ou menos grosseira da localização da peça, podemos manualmente ou automaticamente aferir com mais rigor esses valores.

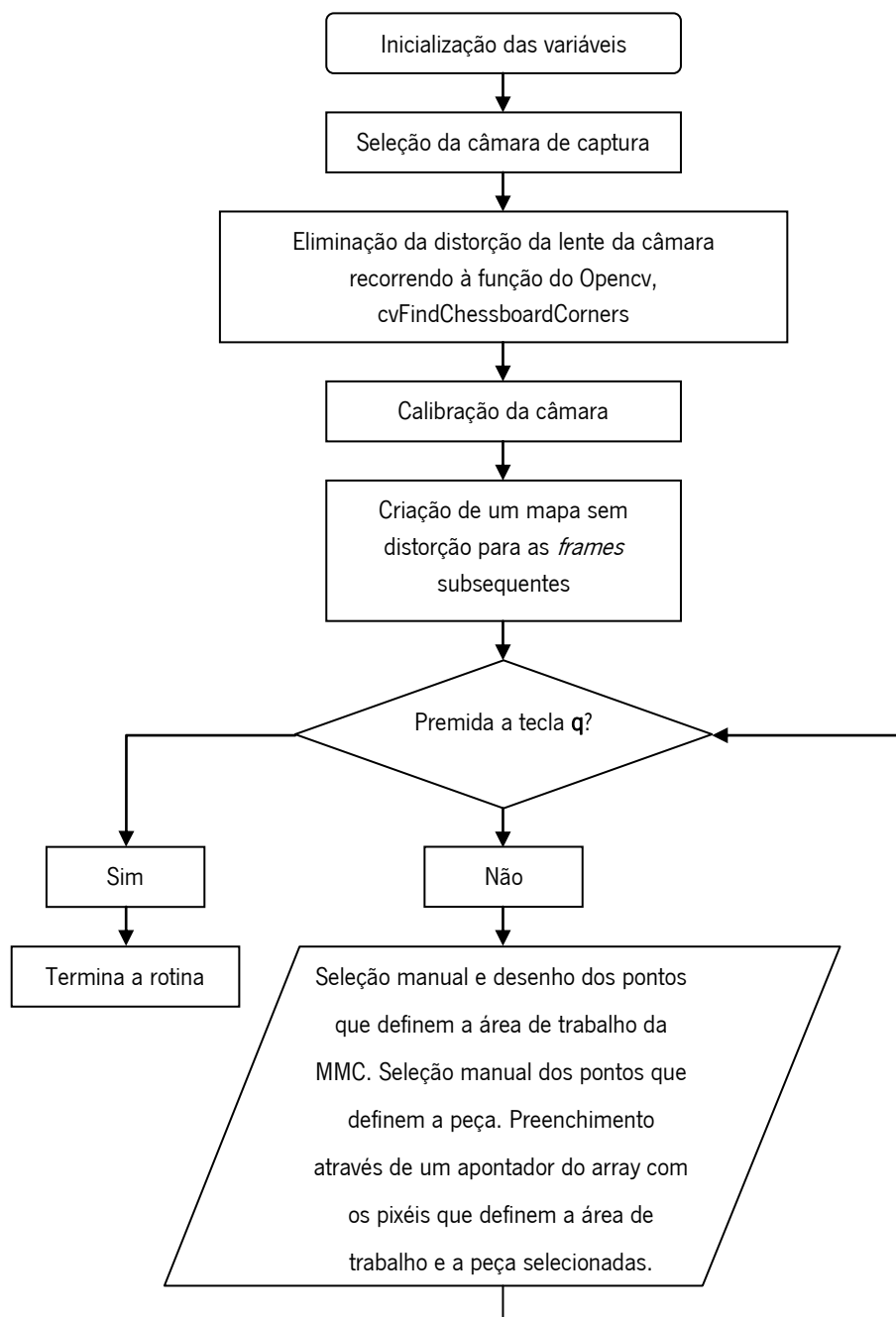


Diagrama 6 Algoritmo da biblioteca DLL com Opencv

Para obter antecipadamente a localização da peça na MMC, foi desenvolvida uma biblioteca DLL (Diagrama 6) em linguagem C++ (ANEXO 5) recorrendo ao ambiente de programação Visual C++ integrando a plataforma de visão por computador *Opencv*. Esta biblioteca inclui uma rotina a ser chamada pelo VBA, ou seja, pode ser chamada em qualquer programa que tenha disponível o ambiente de programação VBA. Inicialmente foi testada com um livro do Excel (Fig. 36), onde a execução de uma macro através de um botão, chama a DLL e no final escreve os dados do *array* de pixels passados por um apontador da folha de cálculo.

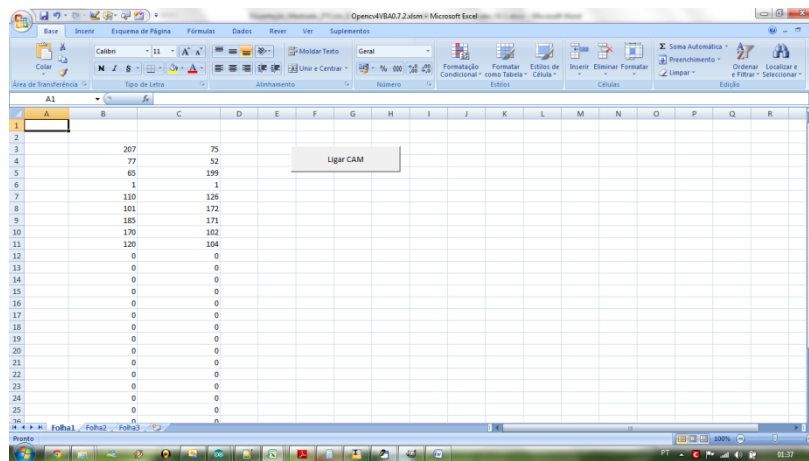


Fig. 36 Folha de cálculo preenchida com dados do *Opencv*

Após a validação dos dados obtidos pela DLL, foi criada uma macro (ANEXO 6) (Diagrama 7) no Inventor que para além de chamar a biblioteca do *Opencv*, seleciona o plano XY e desenha a área de trabalho e a peça selecionada.

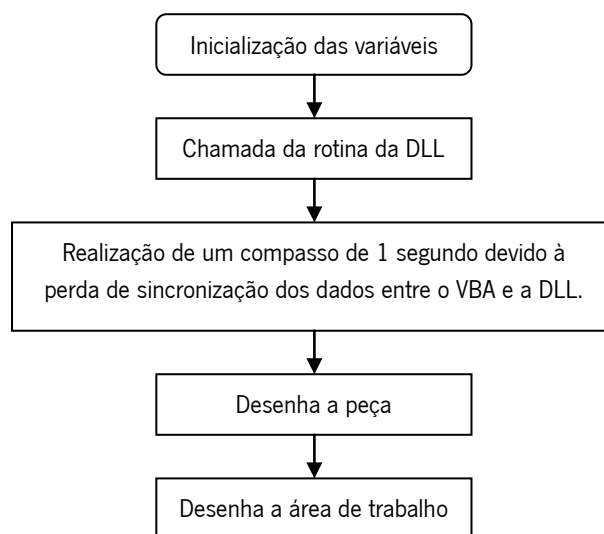


Diagrama 7 Algoritmo do código VBA no Inventor

3.4.3. Programação da leitura de coordenadas

Para realizar uma verificação manual das coordenadas foi realizada uma interface de controlo da MMC, com um formulário, para obtenção de dados (Fig. 37) (ANEXO 6).



Fig. 37 Formulário de controlo manual da MMC

Com este formulário é possível controlar os 3 eixos da MMC, assim como realizar a sua inicialização. Através do retorno do posicionamento e possível deteção da peça é possível realizar o controlo metrológico desejado.

Para além da leitura manual das coordenadas de uma peça é desejável ter duas outras formas de obtenção das coordenadas metrológicas da mesma, através de um sistema autónomo de varrimento ou pelo apuramento das coordenadas obtidas na visão por computador. Apesar de não terem sido implementadas estas soluções nesta dissertação, apresentam-se duas propostas de algoritmo para as mesmas.

Para a leitura por varrimento é necessário registar uma nuvem de coordenadas dos pontos de contacto com a peça (Diagrama 8). De notar que este tipo de solução é lenta pelo facto de se ter que percorrer toda a área de trabalho da MMC.

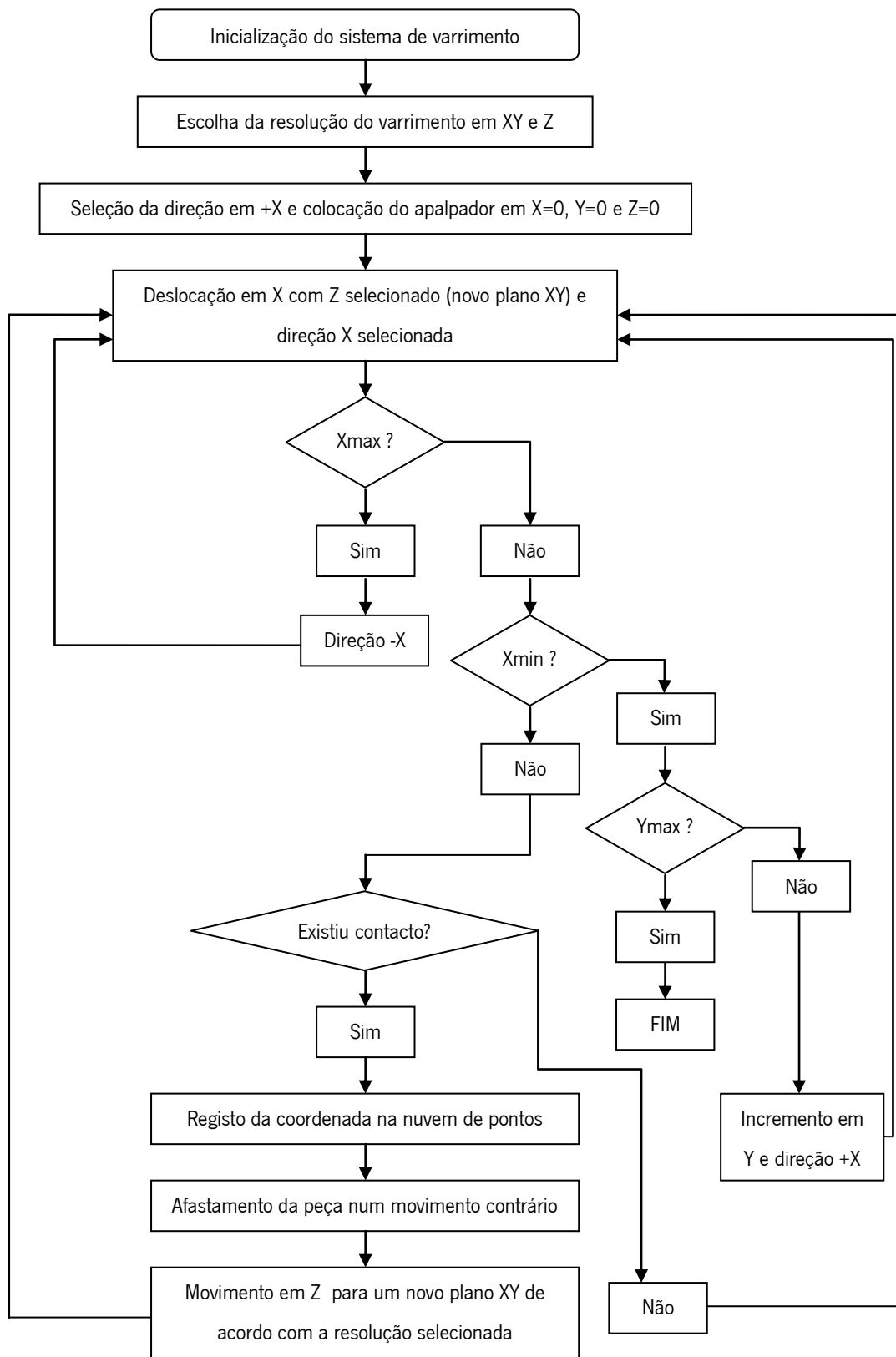


Diagrama 8 Algoritmo de varrimento

Para a leitura por apuramento das coordenadas obtidas na visão por computador (Diagrama 9) é necessário percorrer os pontos obtidos e realizar o apuramento dos mesmos, tornando o algoritmo mais rápido. Os pontos obtidos são neste caso parte integrante do desenho do *Inventor*, obtido por processamento da visão por computador e acessível através do VBA.

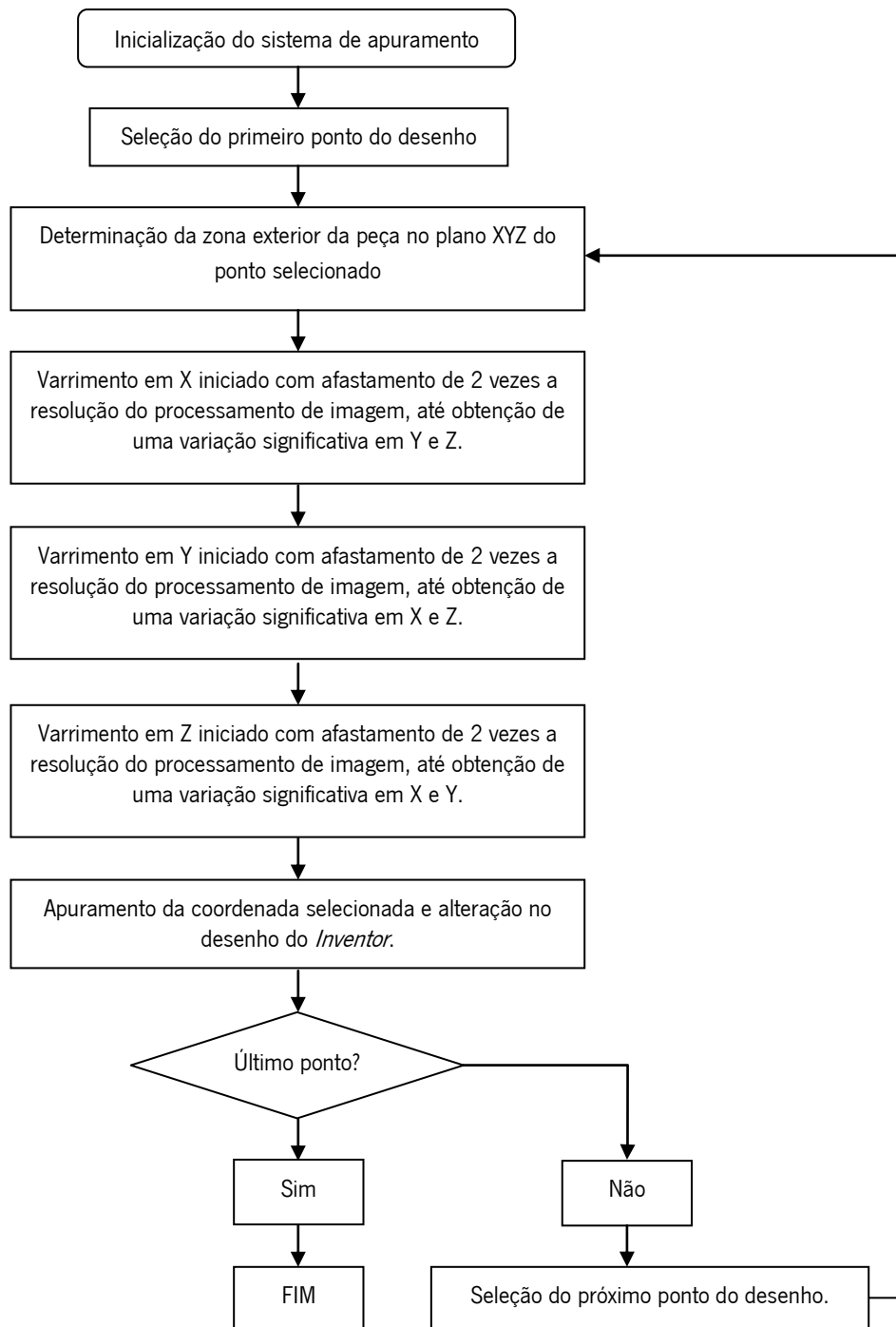


Diagrama 9 Algoritmo de apuramento das coordenadas obtidas na visão por computador

4. Resultados

A título meramente indicativo, a MMC desenvolvida permite uma dimensão máxima das peça a medir de 136,5x186,5x121,5 (mm) (Tabela 6). A construção mecânica do equipamento apresentou diversos defeitos devido à insuficiência de planeamento e aos recursos materiais utilizados. Mas este facto foi de certa forma intencional dado o interesse educativo objetivado para esta dissertação. Isto torna real a possibilidade de construção por alunos dos ciclos de ensino iniciais, mesmo que em termos metrológicos haja pouca exatidão nos resultados que se podem esperar.

Toda a eletrónica simplista utilizada permite uma fácil montagem do sistema e reduz o custo associado à mesma. Foram utilizados os motores de passo reciclados, controlados por placas dedicadas capazes de obter destes 8 micropassos por cada passo, aumentando assim a resolução do sistema. A utilização de uma plataforma eletrónica baseada num projeto aberto, potencializou a vertente educativa e ao mesmo tempo permitiu obter algum nível de autonomia no controlo do processo de medição, principalmente na gestão do contacto pelo apalpador.

O sistema de apalpador implementado com recurso à leitura pela ADC de 10 bit do Arduino, revelou variações durante todo o processo de leitura (Tabela 5) o que não permitiu que fosse usado para obter o contacto com a peça a medir. Note-se que na sequência transcrita na Tabela 5 existiu um contacto com a peça a partir da leitura 7, o que revela a dificuldade de obter um valor a partir do qual se obteve contacto.

Tabela 5 Excerto de leitura da ADC pelo apalpador baseado no piezoelétrico

Leitura	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Valor	23	24	21	27	23	24	25	22	23	21	24	21	23	18

Com o conjunto de motores de passo e controladores obteve-se uma exatidão do equipamento menor que 0,002 mm por eixo, mas sendo calibrada por um paquímetro, a MMC apenas pode garantir uma exatidão de 0,05 mm, igual à do próprio paquímetro. Estes dados foram obtidos por média após a realização de 10 medições, mas apenas servem como referência para a sua

utilização no *Inventor* quando se faz o processamento de imagem e se realiza a transferência dos dados obtidos para a MMC. Estes dados não se devem tomar como valores definitivos da MMC devido à falibilidade da construção mecânica realizada.

Tabela 6 Dados intrínsecos da MMC

Eixo	mm	Passos	Passos/mm	Exatidão	Exatidão do paquímetro
X	136,5	86390	633	0,0016	0,05
Y	186,5	108748	583	0,0017	0,05
Z	121,5	120569	992	0,0010	0,05

Quanto à aplicação de visão por computador associada a um programa CAD, foi possível realizar uma modelação rudimentar da peça, principalmente pela obtenção dos dados sobre as coordenadas no plano XY da mesma, afim de posteriormente se apurar esses dados metrológicos. Foi possível determinar a localização da peça a medir no plano XY com uma resolução menor que 1 mm por pixel. Conforme se pode verificar na Fig. 38, com o recurso ao Inventor/Opencv obteve-se numa peça uma cota de 51,3697mm e o paquímetro indicou para a mesma cota 51,65mm. A diferença é de 0,2803 o que demonstra claramente a potencialidade desta solução para a obtenção da localização primária da peça a medir.

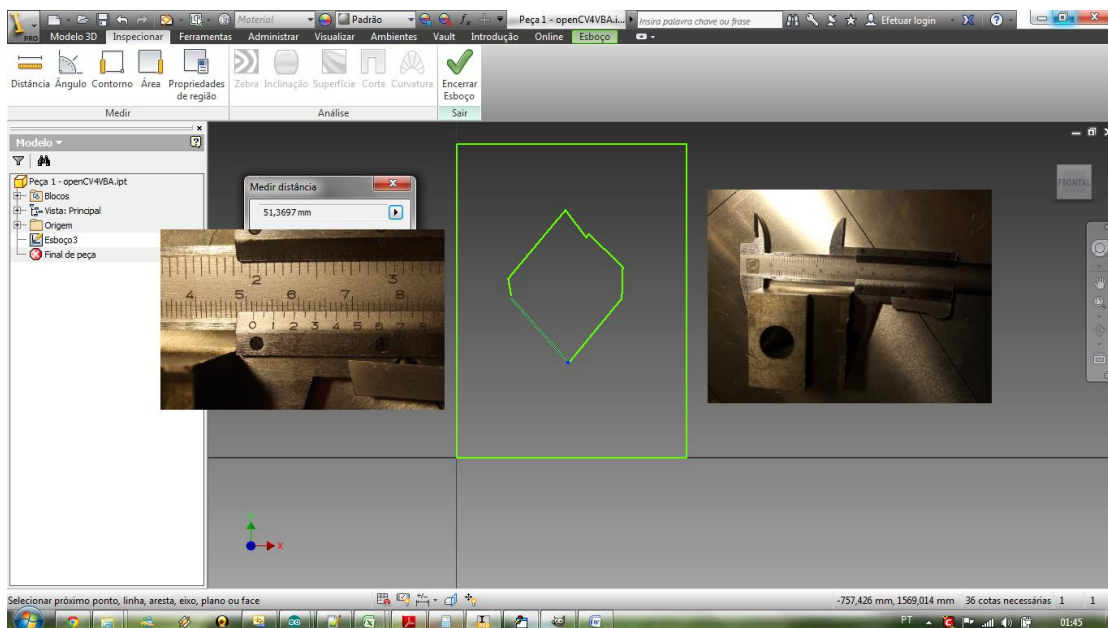


Fig. 38 Verificação do tamanho da peça com Opencv

5. Conclusões e trabalho futuro

Foi construída manualmente uma estrutura que apesar de ter diversos defeitos dimensionais é elucidativa da potencialidade pedagógica e científica. É principalmente nestes campos que o baixo custo se reflete quando se pretende construir conhecimento. Realizar a abertura de equipamentos proprietários ao desenvolvimento de máquinas deste tipo requeria um esforço económico elevado e apenas quando permissível pelos detentores dos direitos. Esta dissertação vem provar que, com um custo muito reduzido e muitas vezes recorrendo a materiais reciclados de outros equipamentos em fim de vida, se pode edificar conhecimento numa vertente pedagógica e científica.

O suporte físico da MMC demonstra a falta de um correto alinhamento dos eixos, o que provoca flutuações visíveis quando há movimento dos mesmos. O futuro desenvolvimento de um equipamento destinado à aferição metrológica de peças deve obedecer a um correto dimensionamento mecânico e a uma escolha mais adequada de materiais, especialmente para a aplicação das guias. Esse dimensionamento pode reduzir claramente os efeitos da falta de alinhamento dos eixos, que juntamente com guias mais robustas e com menos atrito pode permitir a utilização de motores de passo com maior exatidão. Como a guia utilizada para a movimentação da mesa tinha pouca robustez, só a colocação da peça que foi usada em todos os testes, criava pelo efeito do seu peso, um afastamento nas extremidades da mesa em cerca de 900 passos, ou seja, perto de 1mm apontando novamente para a necessidade de um engenho mecânico mais apurado.

Relativamente à eletrónica, esta demonstrou ser capaz de responder às necessidades básicas de uma MMC. O recurso a equipamentos *Opensource* é uma mais-valia para projetos pedagógicos devido às grandes comunidades que estão normalmente associadas aos mesmos. Fica por explorar a utilização de outras formas de atuação dos eixos, nomeadamente com servo motores e motores DC que exigem um maior domínio do controlo dos mesmos perspetivando-se a aplicação de controlo PID em malha fechada.

O sistema de apalpação utilizado, recorrendo ao toque por contacto elétrico, demonstrou viabilidade para um conjunto de situações onde se utilizem peças metálicas. Como a resistência elétrica do ar não é infinita, a resolução de 0,002mm por passo será necessário aprofundar até que ponto há influência da variação da resistência do ar, quer pela humidade quer pela baixa resistência que o ar tem, quando temos contactos assim próximos. Será necessário estudar os sistemas de apalpador baseados em sensores piezoelétricos, já que estes permitem alargar a medição de peças construídas noutros materiais, para além dos que apresentam uma maior condutividade elétrica.

Outro campo onde ficam abertas portas de desenvolvimento futuro é o da aplicação da visão por computador, quer desde a deteção automática do posicionamento das peças até mesmo à sua utilização para o mapeamento total das peças sem recurso a apalpadores por toque. Ficou demonstrada a potencialidade desta solução, integrada nas aplicações de modelação como o Inventor. A facilidade que estes programas permitem a modelação computacional pode ser combinada com ferramentas mecatrónicas otimizando processos de aferição metrológica. É possível a exploração de rotinas de alto nível para realizar determinadas tarefas de medição devendo ser acompanhadas por rotinas de baixo nível no microcontrolador. Ainda neste campo, ficaram propostos dois algoritmos de medição que terão que ser ajustados ao tipo de apalpador utilizado e otimizados para a aferição metrológica que se pretende, como por exemplo, a planicidade ou a circularidade.

Com a miniaturização dos sistemas informáticos, existe a possibilidade de recurso sistemas embebidos capazes de fazer um processamento completo com rotinas pré-programadas de medição. Fica em aberto a aplicação de sistemas como Raspberry Pi ou Arduino DUE para a aplicação de sistemas embebidos.

Bibliografia

APM Technologies. (s.d.). *White Light Scanning, Whiteline Scanners, 3d Scanning Services, Reverse Engineering Services, Rapid Prototyping Services*. Obtido em 01 de 08 de 2011, de APM Technologies: http://www.whitelightscanning.com/php/catg_details.php?catid=14&catgName=ATOS%203D%20White%20Light%20Scanner&pid=14

Banzi, M., Cuartielles, D., Igoe, T., Martino, G., & Mellis, D. (21 de Abril de 2005). *@Arduino*. Obtido em 31 de 03 de 2011, de Arduino - HomePage: <http://arduino.cc>

Barczak, A. L. (1996). *Comparação entre algoritmos de mínimos quadrados e de zona mínima para desvios de circularidade*. São Paulo - Brasil: Universidade Estadual de Campinas - Faculdade de Engenharia Mecânica.

Bennett, E., & Weekers, W. (2010). *Modern CMM - Design Concepts*. Obtido em 01 de 08 de 2011, de TES - Today's Energy Solutions: <http://www.onlinetes.com/FileUploads/file/CMM-Design-Concepts-990137.pdf>

Bosh, J. (1995). *Coordinate Measuring Machines and Systems*. New York: Marcel Dekker, Inc.

Campos, M. J. (21 de 10 de 2010). Desenvolvimento de um sistema de visão para o controlo de qualidade de injeção de peças de automóvel. Guimarães, Minho, Portugal.

Comissão Europeia. (25 de 07 de 2012). *An Integrated Industrial Policy for the Globalisation Era Putting Competitiveness and Sustainability at Centre Stage*. Obtido em 10 de 20 de 2012, de Industrial policy: http://ec.europa.eu/enterprise/policies/industrial-competitiveness/industrial-policy/index_en.htm

Comissão Europeia. (15 de 10 de 2012). *Estratégia para um crescimento inteligente, sustentável e inclusivo*. Obtido em 20 de 10 de 2012, de Europe 2020 - Europe's growth strategy: http://ec.europa.eu/europe2020/index_en.htm

Crato, N. P. (12 de 12 de 2011). Despacho n.º 17169/2011. Lisboa, Portugal.

Eurostat. (19 de 10 de 2012). *Comissão Europeia - Eurostat*. Obtido em 20 de 10 de 2012, de Database:

http://epp.eurostat.ec.europa.eu/portal/page/portal/european_business/data/database

Fernandes, R. (2010). *Desenvolvimento de Sistema de Medição por Coordenadas de Baixo Custo*. Guimarães: UMinho.

Foundation, R. P. (Janeiro de 2012). *Raspberry Pi*. Obtido em 27 de Outubro de 2012, de Raspberry Pi: <http://www.raspberrypi.org>

Guedes, P. (2011). *Metrologia Industrial*. Lisboa: ETEP - Edições Técnicas e Profissionais.

Hartenberg, R. S., & Denavit, J. (1964). *Kinematic Synthesis of Linkages*. New York: McGraw-Hill.

Healthcare, P. (2009). *L12-5 50 mm - Philips*. Obtido em 09 de 12 de 2012, de Philips Healthcare:

http://www.healthcare.philips.com/main/products/ultrasound/transducers/l12_5.wpd

MikroElektronika. (1998). *MikroElektronika - Development tools, Compilers, Books*. Obtido em 28 de outubro de 2012, de MikroElektronika: <http://www.mikroe.com>

Mitutoyo America Corporation. (s.d.). Obtido em 06 de 05 de 2011, de Mitutoyo America Corporation: www.mitutoyo.com

Museum, o. S. (2008). *Timeline / MOSI*. Obtido em 14 de 10 de 2012, de Museum of Science and Industry | MOSI: <http://www.mosi.org.uk/collections/explore-the-collections/ferranti-online/timeline.aspx>

Nikon. (2011). *Nikon Metrology - 3D scanning benefits of laser line probes on Coordinate Measuring Machines*. Obtido em 01 de 08 de 2011, de Nikon Metrology: http://www.nikonmetrology.com/3d_scanning

Park Systems. (2008). *Park Systems Atomic Force Microscope, AFM/SPM*. Obtido em 01 de 08 de 2011, de Park Systems: <http://www.parkafm.com//resources/01general.php>

Pruvot, F. (1993). *Conceptions et calcul des machines-outils*. Lausanne: Presses polytechniques et universitaires romandes.

Renishaw. (2011). *Tipos de Pontas*. Obtido em 06 de 05 de 2011, de Renishaw: <http://www.renishaw.com.br>

Schmalz, B. (nd. de nd. de nd.). *SchmalzHaus.com Brian Schmalz Homepage*. Obtido em 28 de 04 de 2011, de Easy Driver stepper motor driver: <http://www.schmalzhaus.com/EasyDriver/>

The London Podiatry Centre - Scanning Technology. (2009). Obtido em 01 de 08 de 2011, de The London Podiatry Centre: <http://www.londonpodiatry.com/podiatry/scanning-technology>

ANEXOS

ANEXO 1 - Programa criado para a placa PICReady1

ANEXO 2 - Informações sobre a placa de desenvolvimento Arduino Uno

ANEXO 3 - Informações sobre a placa de controlo de motores de passo Easydriver

ANEXO 4 - Código criado para a placa de desenvolvimento Arduino Uno

ANEXO 5 - Código criado para a DLL de processamento de imagem - OpenCV4VBA

ANEXO 6 - Módulo do VBA para chamada da DLL e do formulário de controlo da MMC

ANEXO 1

Programa criado para a placa PICReady1


```

/*      Copyright (C) 2011 Ricardo Fernandes PG15168 - Eng. Mecatrónica 2º Ciclo
        Version 0.3 last updated on 05/08/2011

PICMMC 0.3      Program MAIN
                0.2 *Data reception and transmission
                0.3 *Radip and Coordinated motion functions declarations
                    *Max displacement values of X,Y and Z = "65535"
                    *Max feedrate = 200pps (motor nominal value)
                    *Max feedrate = 5pps (max. microcontroler timer1 value)
*/

/*****PIN_CONFIGURATION_START*****/
/*
PORTB lowest nibble 0b0000xxxx -> X Motor
PORTB highest nibble 0bxxxx0000 -> Y Motor
PORTC lowest nibble 0b0000xxxx -> Z Motor
*/
/*****PIN_CONFIGURATION_END*****/

/*****GLOBALS_START*****/
unsigned int a=0;
unsigned int i=0;
unsigned int k=0;
unsigned int x_value=0; //RANGE = "0..65535"
unsigned int y_value=0; //RANGE = "0..65535"
unsigned int z_value=0; //RANGE = "0..65535"
unsigned short x_step=0; //RANGE = "0..255"
unsigned short y_step=0; //RANGE = "0..255"
unsigned short z_step=0; //RANGE = "0..255"
unsigned short x_portb=0; //RANGE = "0..255" holds bits for x displacement
unsigned short y_portb=0; //RANGE = "0..255" holds bits for y displacement
unsigned short z_portc=0; //RANGE = "0..255" holds bits steps to place in PortC
unsigned short xy_portb=0; //RANGE = "0..255" holds bits steps to place in PortB
//Table step list
unsigned short steptable[4]= {0b00001001,0b00001100,0b0000110,0b00000011};
unsigned int codeindex=0; //Message - ASCII index position
unsigned int feedrate=0; //Feed rate
unsigned int timer1value=0; //to calculate the value for timer 1
unsigned long int feedratetime=0; //Feed rate in miliseconds
unsigned int g_code=0; //GCODE G?? type
char uart_rw=0; //UART variable message IN/OUT
char message_in [50]; //Data reception array
char message_out [50]; //Data transmission array
char decimalascii [4]; //Decimal value to convert from ASCII to binary value
char auxascii [4]; //Decimal value to convert from ASCII to binary value

/*****GLOBALS_END*****/

/*****FUNCTIONS_START*****/
/*Program funcions declaration*/
void standard_init(void);
void g_motion(void);
int str2int(char *string); // convert decimal to binary
/*****FUNCTIONS_END*****/

/*****MAIN_START*****/
void main() {
//initialize device
//TRISA=0xff; //PORTA input
//TRISB=0x00; //PORTB output
//PORTB=0b00000000; //PORTB inicalization

standard_init();

//UART Configuration
UART1_Init(115200); // Initialize UART module at 115200 bps
//UART1_Init(57600); // Initialize UART module at 57600 bps
Delay_ms(100); // Wait for UART module to stabilize

while(1){
//Wait for full message
a=0;
codeindex=0;
while(a<50){
if (UART1_Data_Ready()==1) {
message_in[a]=UART1_Read();
a++;
}
}
}
}

```

```

}
//Send acknowledge
uart_rw=79; // ASCII O
UART1_Write(uart_rw); //send data via UART
uart_rw=75; // ASCII K
UART1_Write(uart_rw); //send data via UART

switch (message_in[codeindex]){
case 71: { //G - Motion GCode
/*****MOTION_GCODE_START*****/
/*****G_MOTION_START*****/
codeindex++; //Update code index to get decimal numbers
//codeindex is @ F - Feedrate
//START Send debug info
uart_rw=33; // ASCII !
UART1_Write(uart_rw); //send data via UART
delay_ms(1000);//Debug
delay_ms(500);
UART1_Write(message_in[codeindex]); //send data via UART
uart_rw=33; // ASCII !
UART1_Write(uart_rw); //send data via UART
delay_ms(1000);//Debug
//END Send debug info
/*****getdecimalvalues_START*****/

for(i=0;i<5;i++){
a=codeindex+i;
decimalascii[i]= message_in[a];
UART1_Write(decimalascii[i]);
}
codeindex=codeindex+5; //Update code index
/*****getdecimalvalues_END*****/
uart_rw=61; // ASCII =
UART1_Write(uart_rw); //send data via UART
delay_ms(1000);//Debug
for (i=0;i<5;i++){
UART1_Write(decimalascii[i]); //Debug send data via UART
}

//START Send debug info
uart_rw=33; // ASCII !
UART1_Write(uart_rw); //send data via UART
delay_ms(2000);
//END Send debug info

g_code= atoi(decimalascii);

//START Send debug info
uart_rw=126; // ASCII ~
UART1_Write(uart_rw); //send data via UART
uart_rw=g_code+48;
UART1_Write(uart_rw); //send data via UART
delay_ms(2000);

uart_rw=61; // ASCII =
UART1_Write(uart_rw); //send data via UART
//END Send debug info

switch (g_code){ // test G?? code
case 0:{
//0 - Rapid Motion @ MAX Feedrate (Normally not used)
/*****RAPID_MOTION_START*****/

/*****RAPID_MOTION_END*****/
}
case 1: {
//1 - Coordinated Motion (This can substitute Rapid Motion)
/*****COORDINATED_MOTION_START*****/
//Feedrate decimal to binay

codeindex++; //update to get decimal feedrate values
//convert decimal to int to get feedrate value
//getdecimalvalues();
/*****getdecimalvalues_START*****/

for(i=0;i<5;i++){
a=codeindex+i;
decimalascii[i]= message_in[a];
//UART1_Write(decimalascii[i]);
}
}
}

```

```

        codeindex=codeindex+5; //Update code index
/*****getdecimalvalues_END*****/
        feedrate= atoi(decimalascii);
        //codeindex is @ X - X displacement
        codeindex++; //Update code index to X displacement value
        uart_rw=95; //ASCII -
        UART1_Write(uart_rw); //send data via UART
        uart_rw=feedrate;
        UART1_Write(uart_rw); //send data via UART
        //X motion value decimal to binay
        //convert decimal to int to get feedrate value
        //getdecimalvalues();
/*****getdecimalvalues_START*****/

        for(i=0;i<5;i++){
                a=codeindex+i;
                decimalascii[i]= message_in[a];
                //UART1_Write(decimalascii[i]);
        }
        codeindex=codeindex+5; //Update code index
/*****getdecimalvalues_END*****/
        x_value= atoi(decimalascii);
        uart_rw=95; //ASCII -
        UART1_Write(uart_rw); //send data via UART
        uart_rw=x_value;
        UART1_Write(uart_rw); //send data via UART
        //codeindex is @ Y - Y displacement
        codeindex++; //Update code index to Y displacement value

        //Y motion value decimal to binay
        //convert decimal to int to get feedrate value
        //getdecimalvalues();
/*****getdecimalvalues_START*****/

        for(i=0;i<5;i++){
                a=codeindex+i;
                decimalascii[i]= message_in[a];
                //UART1_Write(decimalascii[i]);
        }
        codeindex=codeindex+5; //Update code index
/*****getdecimalvalues_END*****/
        y_value= atoi(decimalascii);
        uart_rw=95; //ASCII -
        UART1_Write(uart_rw); //send data via UART
        uart_rw=y_value;
        UART1_Write(uart_rw); //send data via UART
        //codeindex is @ Z - Z displacement
        codeindex++; //Update code index to Z displacement value

        //Y motion value decimal to binay
        //convert decimal to int to get feedrate value
        //getdecimalvalues();
/*****getdecimalvalues_START*****/

        for(i=0;i<5;i++){
                a=codeindex+i;
                decimalascii[i]= message_in[a];
                //UART1_Write(decimalascii[i]);
        }
        codeindex=codeindex+5; //Update code index
/*****getdecimalvalues_END*****/
        z_value= atoi(decimalascii);
        uart_rw=95; //ASCII -
        UART1_Write(uart_rw); //send data via UART
        uart_rw=z_value;
        UART1_Write(uart_rw); //send data via UART
        //codeindex is @ Z - Z displacement
        codeindex++; //Update code index to Z displacement value

        //Enable timer1 interrupt for motion
        T1CON = 1; // Set timer TMR1
        PIR1.TMR1IF = 0; // Reset bit TMR1IF
        // Set initial value for timer TMR1

//Internal instruction cycle =
//= 1 / [(Processor Frequency) / 4] = 1 / (11,0592 MHz / 4) = 361,7nS
//Max Step = 200pps => 5ms max
// TIMER1 is 2^16 internal instruction cycle
//= 361,7nS x 65535 = 23704009,5nS= 23704 μS = 23,7 mS
//prescaler @ 2 => = 23,7 mS*2 = 47,4mS
//prescaler @ 4 => = 23,7 mS*4 = 94,8mS
//prescaler @ 5 => = 23,7 mS*8 = 189,6mS

```

```

//Calculate the feedrate to place in timer
//Prescaler: for each 8 clock cycle I get one timer count
T1CON.T1CKPS1 = 1; //bits 5-4 Prescaler Rate Select bits
T1CON.T1CKPS0 = 1; //bit 4
// This mean one count @ each 2893,6nS = 8*361,7nS
//get feedrate nS
feedratetime=((1/feedrate)*1000*1000*1000);
//Calculate Timer1 Value to assign
timer1value= feedratetime/2893;
uart_rw=95; //ASCII -
UART1_Write(uart_rw); //send data via UART
uart_rw=timer1value;
UART1_Write(uart_rw); //send data via UART
//MAX FEEDRATE of 200pps = 5mS => timer1value=1728
//MAX TIMER1value=65535 => 189,6mS => MIN 5pps

//////////////////////////////////////
TMR1H = feedrate && 0xF0;
TMR1L = feedrate && 0x0F;
PIE1.TMR1IE = 1; // Enable interrupt on overflow
INTCON = 0xC0; // Enable interrupt (bits GIE and PEIE)
uart_rw=91; //ASCII [
UART1_Write(uart_rw); //send data via UART
uart_rw=x_value;
UART1_Write(uart_rw); //send data via UART
uart_rw=93; //ASCII ]
UART1_Write(uart_rw); //send data via UART
uart_rw=x_value;
while ((x_value!=0) && ((y_value!=0) && (z_value!=0))){
    if(x_value<uart_rw){
        uart_rw=91; //ASCII [
        UART1_Write(uart_rw); //send data via UART
        uart_rw=x_value;
        UART1_Write(uart_rw); //send data via UART

        uart_rw=95; //ASCII -
        UART1_Write(uart_rw); //send data via UART
        uart_rw=xy_portb;
        UART1_Write(uart_rw); //send data via UART
        uart_rw=95; //ASCII -
        UART1_Write(uart_rw); //send data via UART
        uart_rw=z_portc;
        UART1_Write(uart_rw); //send data via UART

        uart_rw=93; //ASCII ]
        UART1_Write(uart_rw); //send data via UART
        uart_rw=x_value;
    }
}
//Disable timer1 interrupt
T1CON = 0; // Unset timer TMR1
PIR1.TMR1IF = 0; // Reset bit TMR1IF
PIE1.TMR1IE = 0; // Disable interrupt on overflow
INTCON = 0x00; // disable interrupt (bits GIE and PEIE)
PORTB=0b00000000; // rest motor coils for no overheating
break;
/*****COORDINATED_MOTION_END*****/
/*****DEFAULT_ERROR_MESSAGE_"ER"_START*****/
uart_rw=79; // ASCII O
UART1_Write(uart_rw); // and send data via UART
uart_rw=75; // ASCII K
UART1_Write(uart_rw); // and send data via UART
/*****DEFAULT_ERROR_MESSAGE_"ER"_END*****/
}
default:{
/*****DEFAULT_ERROR_MESSAGE_"ER"_START*****/
uart_rw=69; // ASCII E
UART1_Write(uart_rw); // and send data via UART
uart_rw=82; // ASCII R
UART1_Write(uart_rw); // and send data via UART
/*****DEFAULT_ERROR_MESSAGE_"ER"_END*****/
}
} // switch
/*****COORDINATED_MOTION_END*****/
break;
/*****MOTION_GCODE_END*****/
}
default:{

```



```

        uart_rw=69;    // ASCII E
        UART1_Write(uart_rw);    // and send data via UART
        uart_rw=82;    // ASCII R
        UART1_Write(uart_rw);    // and send data via UART
    }

} // switch

Delay_ms(500); //DELAY
a=0;
while(a<50){
    UART1_Write(message_in[a]);
    a++;
}
} //while
} //main
/*****MAIN_END*****/

/*****INTERRUPT_FUNCTION_START*****/
void interrupt() {

    PIR1.TMR1IF = 0;    // Reset bit TMR1IF
    //set step number

    xy_portb=steptable[x_step] & 0xFF;
    xy_portb &=(steptable[Y_step] & 0xFF);
    z_portc=steptable[z_step];

    PORTB=xy_portb;
    PORTC=z_portc;

    x_value--;    // Interrupt causes x_value to be decremented by 1
    y_value--;    // Interrupt causes y_value to be decremented by 1
    z_value--;    // Interrupt causes z_value to be decremented by 1

    //Step increment
    if (x_step==4){
        x_step=0;
    }else{
        x_step++;
    }
    if (y_step==4){
        y_step=0;
    }else{
        y_step++;
    }
    if (z_step==4){
        z_step=0;
    }else{
        z_step++;
    }
}
/*****INTERRUPT_FUNCTION_END*****/

/*****AUXILIARY_FUNCTIONS_START*****/
/*****str2int_FUNCTION_START*****/
int str2int(char *string){
    int outputint = 0;
    int i = 1;
    int stringlength = 5;
    int temp = 0;
    int numberlength = 0;
    int j;
    char tempchar;

    //Math string to decimal

    for(j=0;j<5;j++){
        tempchar = *string;
        *string++;
        temp = tempchar - 48;
        if (temp >= 0 && temp <= 9)
        {
            outputint += temp * pow(10, (5-j));
            //j++;
        }
    }
}

```

```

return outputint;
}
/*****str2int_FUNCTION_END*****/

/*****standard_init_FUNCTION_START*****/
void standard_init(void) {
//HARDWARE_STARTUP_SETTINGS_START
//Device Port Function Control
CM1CON0 = 0x00; // Comparator 1 set off
CM2CON0 = 0x00; // Comparator 2 set off
CM2CON1 = 0x02; // Comparator 2 Timer1 Control register default set
ADCON0 = 0x00; // ADC off
ADCON1 = 0x80; // ADC right justified
ANSEL = 0x00; // ALL set as Digital I/O
ANSELH = 0x00; // ALL set as Digital I/O

// Interrupt Reset

INTCON.GIE = 0x00; // Bit7 disable Global interrupts
INTCON.PEIE = 0x00; // Bit6 disable Peripheral interrupts
INTCON.T0IE = 0x00; // Bit5 disable Timer0 Overflow interrupt
INTCON.INTE = 0x00; // Bit4 disable External interrupts
INTCON.RBIE = 0x00; // Bit3 disable PortB Change interrupts

INTCON.T0IF = 0x00; // Bit2 Clear Timer0 Overflow interrupt Flag
INTCON.INTF = 0x00; // Bit1 Clear External interrupt Flag
INTCON.RBIF = 0x00; // Bit0 Clear portB Change interrupt Flag

//Option register OPTION_REG.bit

OPTION_REG.NOT_RBPU = 0x1; // Bit7 portB pull-ups [0 = ON 1 = OFF]
// Bit6 Interrupt Edge Detect bit [0 = Disabled 1 = Enabled]
OPTION_REG.INTEDG = 0x0;
// Bit5 Timer0 clock source select bit [0 = FOSC/4 1 = T0CKI]
OPTION_REG.T0CS = 0x0;
// Bit4 Timer0 source edge select bit [0 = Low to High 1 = High to Low]
OPTION_REG.T0SE = 0x0;
// Bit3 Prescaler Assignment bit [0 = Timer0 1 = WDT]
OPTION_REG.PSA = 0x0;
OPTION_REG.PS2 = 0x0; // Bit2 Prescaler Rate Select
OPTION_REG.PS1 = 0x0; // Bit1 Prescaler Rate Select
OPTION_REG.PS0 = 0x0; // Bit0 Prescaler Rate Select

// Timer1 setup
PIE1.TMR1IE = 0; // Timer1 interrupt disabled
PIR1.TMR1IF = 0; // Timer1 interrupt Flag cleared

T1CON.T1GINV = 0;
T1CON.TMR1GE = 0;
T1CON.T1CKPS1 = 1;
T1CON.T1CKPS0 = 1;
T1CON.T1OSCEN = 0;
T1CON.T1SYNC = 0;
T1CON.TMR1CS = 0;
T1CON.TMR1ON = 1;

// I/O Port standard setup
TRISA = 0x00; // Set ALL ports as output, this is device default
TRISB = 0x00; // Set ALL ports as output, this is device default
TRISC = 0x00; // Set ALL ports as output, this is device default
TRISD = 0xFF; // Set ALL ports as input, this is device default
TRISE = 0xFF; // Set ALL ports as input, this is device default

PORTA = 0x00; // send 0 to Outputs just to have a known buffer value
PORTB = 0x00; // send 0 to Outputs just to have motors coil non excited
PORTC = 0x00; // send 0 to Outputs just to have motors coil non excited
PORTD = 0x00; // send 0 to Inputs just to have a known buffer value
PORTE = 0x00; // send 0 to Inputs just to have a known buffer value

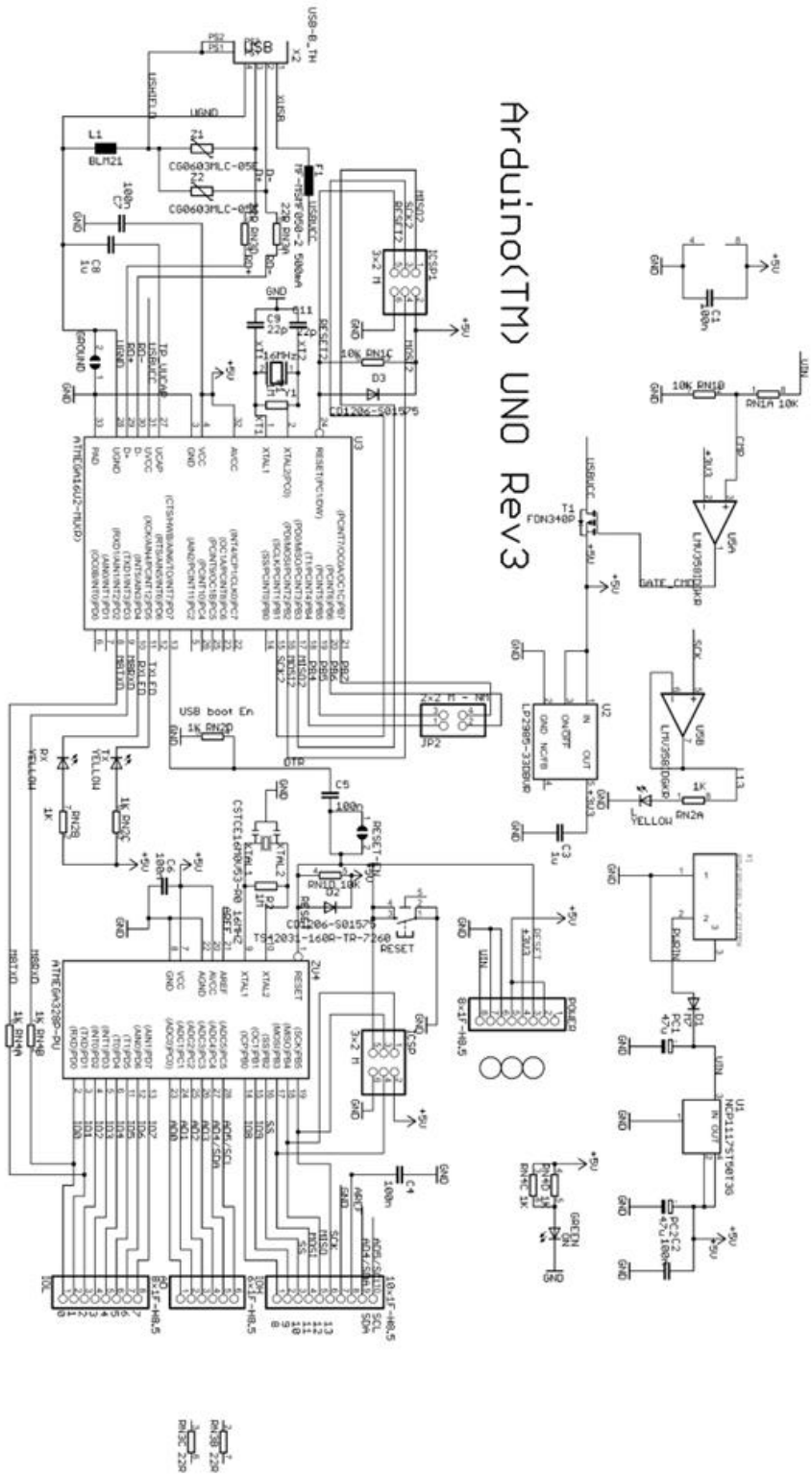
//HARDWARE_STARTUP_SETTINGS_END
}
/*****standard_init_FUNCTION_END*****/
/*****AUXILIARY_FUNCTIONS_END*****/

```

ANEXO 2

Informações sobre a placa de desenvolvimento Arduino Uno

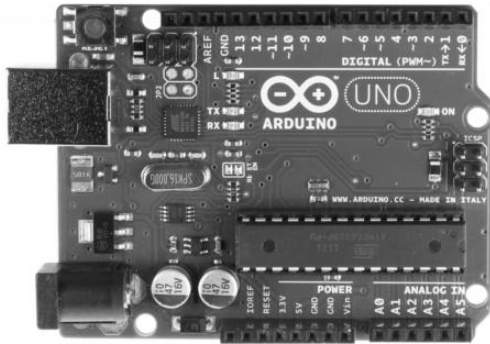
Esquema



Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information. ARDUINO is a registered trademark.

Use of the ARDUINO name must be compliant with <http://www.arduino.cc/en/Main/Policy>

Foto



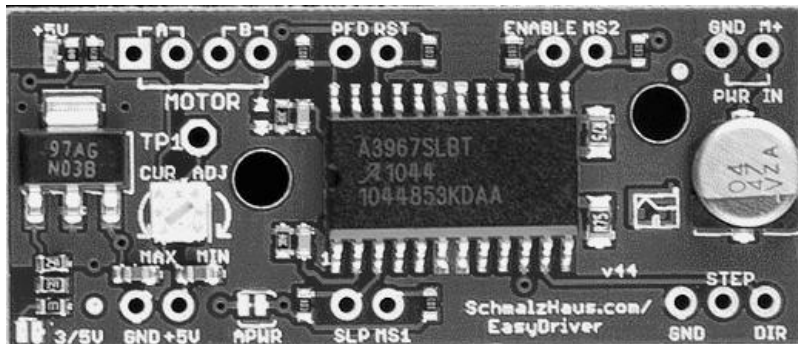
Endereço oficial na Internet

- www.arduino.cc

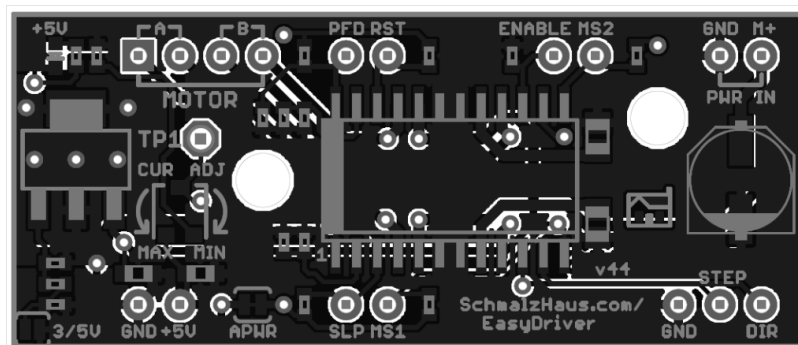
ANEXO 3

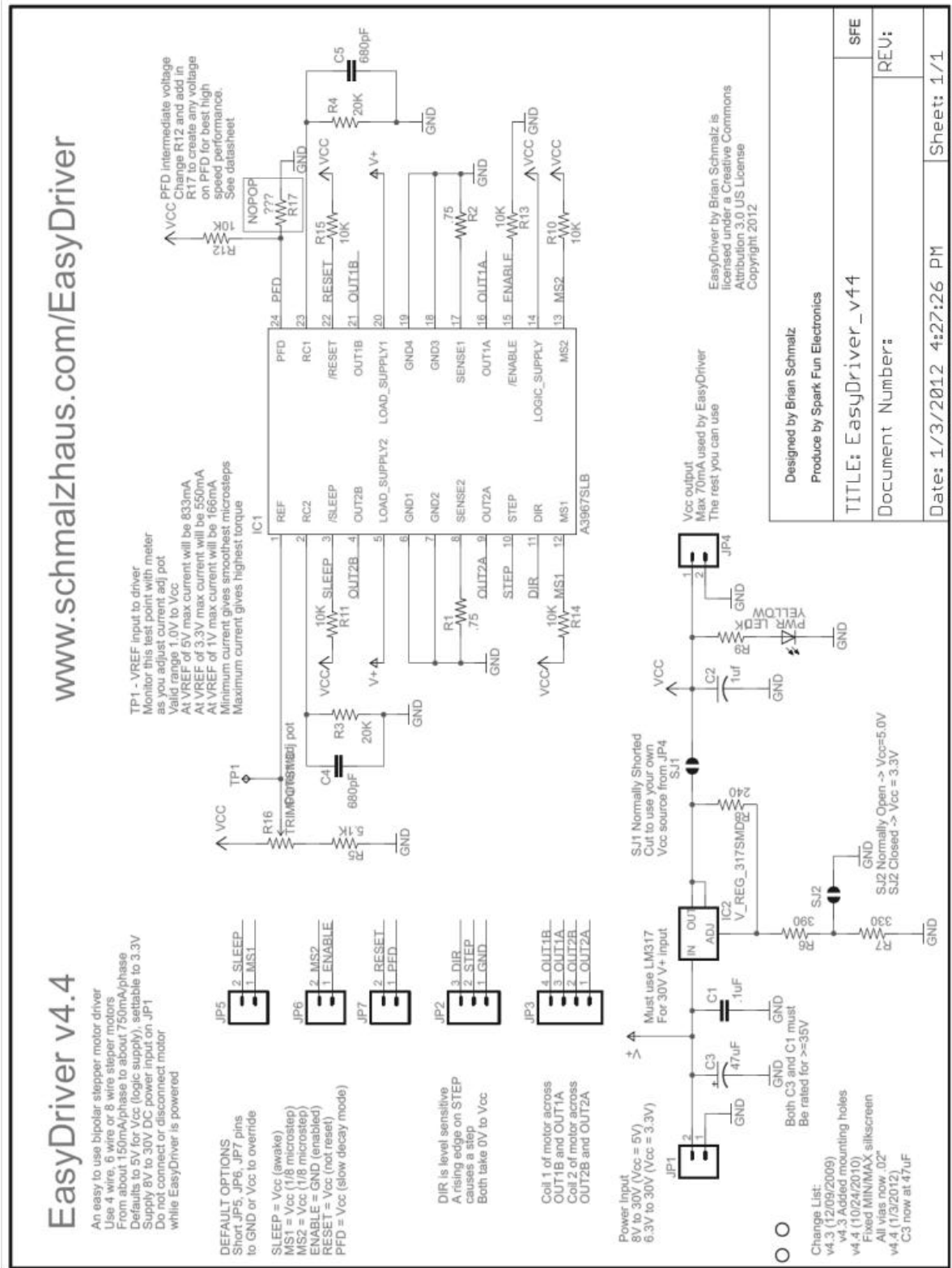
Informações sobre a placa de controlo de motores de passo Easydriver

Foto



PCB





ANEXO 4

Código criado para a placa de desenvolvimento Arduino Uno

Código do Arduino

```
/* Copyright (C) 2012 Ricardo Fernandes <rccff@gmail.com>
   PG15168 - Eng. Mecatrónica 2º Ciclo
   Version 1.0 last updated on 24/10/2012

Arduino4MMC 1.0    Program MAIN
                   0.2 *Data reception and transmission
                   0.3 *Stepper movement tests
                   0.4 *G00 motion
                   0.5 *Read Piezo ADC Values
                   0.6 *Test for contact readings
                   0.7 *Interrupt for probe contact
                   0.8 *Interrupt for axis limits
                   0.9 *Correction of movement G00
                   1.0 *INIT routine
*/

#define MzDIR_PIN 4
#define MyDIR_PIN 6
#define MxDIR_PIN 8
#define MzSTEP_PIN 5
#define MySTEP_PIN 7
#define MxSTEP_PIN 9
#define MicroXMinus_PIN 10
#define MicroXPlus_PIN 11
#define MicroYMinus_PIN 12
#define MicroYPlus_PIN 13
#define MicroZMinus_PIN A0
#define MicroZPlus_PIN A1

//const int piezoSensor = A0; //OLD the piezo is connected to analog pin 0
//int piezoSRead = 0; //OLD variable to store the value read from the sensor pin

//Probe and Microswitch variables for Interrupt
int microTouchInt=0; //Pin 2
int microSwitchInt=1; //Pin 3

//flag Probe Touch and micro to store values inside Interrupt
volatile int flagTouchContact=0;
volatile int flagSwitchContact=0;

//Flags for X,Y and Z limits
int flagMicroXMinus;
int flagMicroXPlus;
int flagMicroYMinus;
int flagMicroYPlus;
int flagMicroZMinus;
int flagMicroZPlus;

//Flag for system initialization
int flagMicroInit=1;

//Variable to store Absolute Steps
long absoluteXstep=86390;
long absoluteYstep=108748;
long absoluteZstep=120569;

//Variable to store Probe stop Steps from ordered steps
int stopStep=0;

//Variable to calculate the steps ordered in X,Y and Z
long MOVx=0;
long MOVy=0;
long MOVz=0;

//Variables for SERIAL communications
//Allocate memory for the serial incoming string
char inData[26];
//Index for the array
byte index = 0;
//Variable to store the character read
char inChar;
//Variable to store the character to write
char outChar;

// Routine to setup initial parameters
void setup() {
```

```

//Setup PIN Modes
pinMode(MzDIR_PIN, OUTPUT);
pinMode(MyDIR_PIN, OUTPUT);
pinMode(MxDIR_PIN, OUTPUT);
pinMode(MzSTEP_PIN, OUTPUT);
pinMode(MySTEP_PIN, OUTPUT);
pinMode(MxSTEP_PIN, OUTPUT);

pinMode(MicroXMinus_PIN, INPUT);
pinMode(MicroXPlus_PIN, INPUT);
pinMode(MicroYMinus_PIN, INPUT);
pinMode(MicroYPlus_PIN, INPUT);
pinMode(MicroZMinus_PIN, INPUT);
pinMode(MicroZPlus_PIN, INPUT);

//Setup initial port values
digitalWrite(MzDIR_PIN, LOW);
digitalWrite(MyDIR_PIN, LOW);
digitalWrite(MxDIR_PIN, LOW);
digitalWrite(MzSTEP_PIN, LOW);
digitalWrite(MySTEP_PIN, LOW);
digitalWrite(MxSTEP_PIN, LOW);
//attach the Interrupts
attachInterrupt(microTouchInt,attendtouch, RISING);
attachInterrupt(microSwitchInt,attendmicro, RISING);

//start serial port at 9600 bps and wait for port to open:
Serial.begin(9600);
while (!Serial) {
  ; // wait for serial port to connect.
}
// send a byte to establish contact until receiver responds
establishContact();

}

void loop(){

//Wait for a 24 character set
while(index<24){
//Wait for some character transmission
while(Serial.available() > 0)
{
  //Read until de 24 item transmission
  if(index < 24)
  {
    //Read a character
    inChar = Serial.read();
    //Return the same character
    //Serial.print(inChar);
    //Store the character
    inData[index] = inChar;
    //Increment the index to write next
    index++;
    //Terminate the string with NULL
    inData[index] = '\0';
  }
}
}
//Test the transmission for adapted G00 (Steps received)
if (inData[0]== 'G' and inData[1]== '0' and inData[2]== '0') //G00
{
  //Format received should match G00X(signal)xxxxY(signal)yyyyZ(signal)zzzz
  //Give Serial feedback of received G00
  Serial.println("-G00");
  //Give Serial feedback of Probe Touch
  Serial.println("-");
  Serial.println(flagTouchContact,DEC);

  //Calculate X steps from received string
  MOVx=0;
  MOVx+= (long(inData[5]- 0x30) * 10000);
  MOVx+= (long(inData[6]- 0x30) * 1000);
  MOVx+= (long(inData[7]- 0x30) * 100);
  MOVx+= (long(inData[8]- 0x30) * 10);
  MOVx+= (long(inData[9]- 0x30) * 1);
  //Calculate X Step direction
  if (inData[4]=='-')
  {

```

```

        MOVx=-MOVx;
    }
    //Feedback X steps
    Serial.println("-");
    Serial.println(MOVx, DEC);
    //Move X steps at speed 0.15
    movex(MOVx, .15);
    //Feedback X Absolute Setps
    Serial.println("-");
    Serial.println(absoluteXstep, DEC);

    //Calculate Y steps from received string
    MOVy=0;
    MOVy+= (long(inData[12]- 0x30) * 10000);
    MOVy+= (long(inData[13]- 0x30) * 1000);
    MOVy+= (long(inData[14]- 0x30) * 100);
    MOVy+= (long(inData[15]- 0x30) * 10);
    MOVy+= (long(inData[16]- 0x30) * 1);
    //Calculate Y Step direction
    if (inData[11]=='-')
    {
        MOVy=-MOVy;
    }
    //Feedback Y steps
    Serial.println("-");
    Serial.println(MOVy, DEC);
    //Move X steps at speed 0.25
    movey(MOVy, .25);
    //Feedback Y Absolute Setps
    Serial.println("-");
    Serial.println(absoluteYstep, DEC);

    //Calculate Z steps from received string
    MOVz=0;
    MOVz+= (long(inData[19]- 0x30) * 10000);
    MOVz+= (long(inData[20]- 0x30) * 1000);
    MOVz+= (long(inData[21]- 0x30) * 100);
    MOVz+= (long(inData[22]- 0x30) * 10);
    MOVz+= (long(inData[23]- 0x30) * 1);
    //Calculate Z Step direction
    if (inData[18]=='-')
    {
        MOVz=-MOVz;
    }
    //Feedback Z steps
    Serial.println("-");
    Serial.println(MOVz, DEC);
    //Move Z steps at speed 0.25
    movez(MOVz, .25);
    //Feedback Z Absolute Setps
    Serial.println("-");
    Serial.println(absoluteZstep, DEC);

    //Feedback how many steps were done
    Serial.println("-S");
    Serial.println(stopStep,DEC);
    //Reset stoped steps
    stopStep=0;
}
//Test the transmission for adapted G00 (Steps received)
if (inData[0]=='I' and inData[1]=='N' and inData[2]=='I') //G00
{
    //Format received should match G00X(signal)xxxxY(signal)yyyyZ(signal)zzzz
    //Give Serial feedback of received G00
    Serial.println("-INITIALIZATION-");
    flagMicroInit=1;
    initialization();
    //Init done 1
    Serial.println("1");
}
//Reset index value to receive new string
index=0;
//Final Feedback
Serial.println("-D");
}

//Attend interrupt for Probe
void initialization()
{
    //Get 0 position on all axis
    while(absoluteXstep!=0){

```

```

    //Move X steps at speed 0.2
    movex(-4000, .1);
    Serial.println("Xabs=");
    Serial.println(absoluteXstep, DEC);
}
//Get 0 position on all axis
while(absoluteYstep!=0){
    //Move X steps at speed 0.2
    movey(-4000, .2);
    Serial.println("Yabs=");
    Serial.println(absoluteYstep, DEC);
}
//Get 0 position on all axis
while(absoluteZstep!=1000){
    //Move X steps at speed 0.2
    movez(-4000, .2);
    Serial.println("Zabs=");
    Serial.println(absoluteZstep, DEC);
}
}

//Attend interrupt for Probe
void attendtouch()
{
    //A touch happened, changing flag
    flagTouchContact=1;
}
//Attend interrupt for Microswitch
void attendmicro()
{
    //A microswitch was activated, changing flag
    flagSwitchContact=1;
}

//OLD Test routine for piezo readings
//void readpiezo(){
//    piezoSRead = analogRead(piezoSensor);
//    Serial.println(piezoSRead);
//}

//Wait for SERIAL conection
void establishContact() {
    while (Serial.available() <= 0) {
        //String sent to wait for connection
        Serial.println("Ready");
        delay(1000);
    }
}

//Routine to move in X
void movex(long steps, float speed){
    //Move a specific number of microsteps (8 microsteps = 1 full step) - (negative for reverse movement)
    //Speed is any number from .01 -> 1 with 1 being fastest - Slower is stronger binary
    //Determine direction
    int dir = (steps > 0)? HIGH:LOW;
    //Variable to use in FOR cycle
    int k=0;
    //Get absolute steps for EASYDRIVE
    steps = abs(steps);
    //Set X direction on EASYDRIVE
    digitalWrite(MxDIR_PIN,dir);
    //Determine delay to set speed
    float usDelay = (1/speed) * 70;
    //FOR cycle where X movement happens
    for(int i=0; i < steps; i++){
        //Test Probe and MicroSwitch flags
        if (flagTouchContact==0 && flagSwitchContact==0)
        {
            //No Touch or Switch : normal steps
            //Increment or Decrement Absolute Value
            if (dir==HIGH)absoluteXstep++;
            else absoluteXstep--;
            //Move ONE X STEP
            digitalWrite(MxSTEP_PIN, HIGH);
            delayMicroseconds(usDelay);
            digitalWrite(MxSTEP_PIN, LOW);
            delayMicroseconds(usDelay);
        }else{
            //A Touch or Switch happens : special steps
            //Read values to check which happens: Touch or Switch
            flagMicroXMinus=digitalRead(MicroXMinus_PIN);

```



```

flagMicroXPlus=digitalRead(MicroXPlus_PIN);
//X Minus SWITCH attend
if (flagMicroXMinus==1){
  //Reset flag
  flagSwitchContact=0;
  //X -> SET Absolut variable to 0
  absoluteXstep=0;
}
//X Plus SWITCH attend
if (flagMicroXPlus==1) {
  //Reset Swicth flag
  flagSwitchContact=0;
}
// Change direction to stop Swicth signal
if (dir==HIGH){
  dir=LOW;
}else{
  dir=HIGH;
}
//Set new X direction on EASYDRIVE
digitalWrite(MxDIR_PIN,dir);
//Set the number of setps done

stopStep=i+1;
//Go 1000 Steps away from Switch or Probe
for(int k=0; k < 1000; k++){
  if (flagMicroInit!=1) {
    //Increment or Decrement Absolute Value
    if (dir==HIGH)absoluteXstep++;
    else absoluteXstep--;
  }
  //Move ONE X STEP
  digitalWrite(MxSTEP_PIN, HIGH);
  delayMicroseconds(usDelay);
  digitalWrite(MxSTEP_PIN, LOW);
  delayMicroseconds(usDelay);
}
flagSwitchContact=0;
flagTouchContact=0;
i=steps;
}
}
}

//Routine to move in Y
void movey(long steps, float speed){
  //Move a specific number of microsteps (8 microsteps = 1 full step) - (negative for reverse movement)
  //Speed is any number from .01 -> 1 with 1 being fastest - Slower is stronger binary
  //Determine direction
  int dir = (steps > 0)? HIGH:LOW;
  //Variable to use in FOR cycle
  int k=0;
  //Get absolute steps for EASYDRIVE
  steps = abs(steps);
  //Set Y direction on EASYDRIVE
  digitalWrite(MyDIR_PIN,dir);
  //Determine delay to set speed
  float usDelay = (1/speed) * 70;
  //FOR cycle where Y movement happens
  for(int i=0; i < steps; i++){
    if (flagTouchContact==0 && flagSwitchContact==0)
    {
      //No Touch or Switch : normal steps
      //Increment or Decrement Absolute Value
      if (dir==HIGH)absoluteYstep++;
      else absoluteYstep--;
      //Move ONE Y STEP
      digitalWrite(MySTEP_PIN, HIGH);
      delayMicroseconds(usDelay);
      digitalWrite(MySTEP_PIN, LOW);
      delayMicroseconds(usDelay);
    }else{
      //A Touch or Switch happens : special steps
      //Read values to check which happens: Touch or Switch
      flagMicroYMinus=digitalRead(MicroYMinus_PIN);
      flagMicroYPlus=digitalRead(MicroYPlus_PIN);
      //Y Minus SWITCH attend
      if (flagMicroYMinus==1) {
        //Y -> SET Absolut variable to 0
        absoluteYstep=0;
      }
    }
  }
}

```

```

//Y Plus SWITCH attend
if (flagMicroYPlus==1) {
  //Reset Swicth flag
  flagSwitchContact=0;
}
// Change direction to stop Swicth signal
if (dir==HIGH){
  dir=LOW;
}else{
  dir=HIGH;
}
//Set new Y direction on EASYDRIVE
digitalWrite(MyDIR_PIN,dir);
//Set the number of setps done
stopStep=i+1;
//Go 1000 Steps away from Switch or Probe
for(int k=0; k < 1000; k++){
  //Increment or Decrement Absolute Value
  if (flagMicroInit!=1) {
    //Increment or Decrement Absolute Value
    if (dir==HIGH)absoluteYstep++;
    else absoluteYstep--;
  }
  //Move ONE Y STEP
  digitalWrite(MySTEP_PIN, HIGH);
  delayMicroseconds(usDelay);
  digitalWrite(MySTEP_PIN, LOW);
  delayMicroseconds(usDelay);
}
flagSwitchContact=0;
flagTouchContact=0;
i=steps+1;
}
}
}

//Routine to move in Z
void movez(long steps, float speed){
  //Move a specific number of microsteps (8 microsteps = 1 full step) - (negative for reverse movement)
  //Speed is any number from .01 -> 1 with 1 being fastest - Slower is stronger binary
  //Determine direction
  int dir = (steps > 0)? HIGH:LOW;
  //Variable to use in FOR cycle
  int k=0;
  //Get absolute steps for EASYDRIVE
  steps = abs(steps);
  //Set Z direction on EASYDRIVE
  digitalWrite(MzDIR_PIN,dir);
  //Determine delay to set speed
  float usDelay = (1/speed) * 70;
  //FOR cycle where Z movement happens
  for(int i=0; i < steps; i++){
    //Test Probe and MicroSwitch flags
    if (flagTouchContact==0 && flagSwitchContact==0)
    {
      //No Touch or Switch : normal steps
      //Increment or Decrement Absolute Value
      if (dir==HIGH)absoluteZstep++;
      else absoluteZstep--;
      //Move ONE Z STEP
      digitalWrite(MzSTEP_PIN, HIGH);
      delayMicroseconds(usDelay);
      digitalWrite(MzSTEP_PIN, LOW);
      delayMicroseconds(usDelay);
    }else{
      //A Touch or Switch happens : special steps
      //Read values to check which happens: Touch or Switch
      flagMicroZMinus=digitalRead(MicroZMinus_PIN);
      flagMicroZPlus=digitalRead(MicroZPlus_PIN);
      //Z Minus SWITCH attend
      if (flagMicroZMinus==1){
        //Reset flag
        flagSwitchContact=0;
        //Z -> SET Absolut variable to 0
        absoluteZstep=0;
      }
      //Z Plus SWITCH attend
      if (flagMicroZPlus==1) {
        //Reset Swicth flag
        flagSwitchContact=0;
      }
    }
  }
}

```

```

    if (flagMicroInit==1) {
        //Z -> SET Absolut variable to 0 by probe
        absoluteZstep=0;
        flagMicroInit=0;
    }
    // Change direction to stop Swicth signal
    if (dir==HIGH){
        dir=LOW;
    }else{
        dir=HIGH;
    }
    //Set new Z direction on EASYDRIVE
    digitalWrite(MzDIR_PIN,dir);
    //Set the number of setps done
    stopStep=i+1;
    //Go 1000 Steps away from Switch or Probe
    for(int k=0; k < 1000; k++){
        //Increment or Decrement Absolute Value
        if (dir==HIGH)absoluteZstep++;
        else absoluteZstep--;
        //Move ONE Z STEP
        digitalWrite(MzSTEP_PIN, HIGH);
        delayMicroseconds(usDelay);
        digitalWrite(MzSTEP_PIN, LOW);
        delayMicroseconds(usDelay);
    }
    flagSwitchContact=0;
    flagTouchContact=0;
    i=steps;
}
}
}

```


ANEXO 5

Código criado para a DLL de processamento de imagem - OpenCV4VBA

OpenCV4VBA

```
/* opencv4VBA 0.6
* Author Ricardo Fernandes <rrccff@gmail.com>
* PG15168 - Eng. Mecatrónica 2º Ciclo
* Version 0.6 last updated on 06/10/2012
* License Copyright
*
* V0.1 Basic Dll
* V0.2 Chess Corners
* V0.3 Calibration complete
* V0.4 Mouse handling and line drawing
* V0.5 Draw coordinates to VBA
* V0.6 Draw Origin of MMC and Object Lines;
*/

#include <stdio.h>
#include <cv.h>
#include <highgui.h>
#include <stdlib.h>

#define TPL_WIDTH 12 /* template width */
#define TPL_HEIGHT 12 /* template height */
#define WINDOW_WIDTH 24 /* search window width */
#define WINDOW_HEIGHT 24 /* search window height */
#define THRESHOLD 0.3
int key;
double o4v_data_width;

int corner_count;
int n_boards = 0; //Number of snapshots of the chessboard
const int board_dt = 20;
int board_w;
int board_h;
int board_total;

int mousex=0;
int mousey=0;
int seq=0;

// Structure of the array to receive by reference
typedef struct
{
    int a[30];
    int b[30];
} MyStruct;

//Mouse attend events
void __cdecl on_mouse( int event, int x, int y, int flags, void* param )
{
    if(event==CV_EVENT_RBUTTONDOWN)
    {
        seq=seq+1;
        if (seq==2){
            mousex=1;
            mousey=1;
        }
    }
    if(event==CV_EVENT_LBUTTONDOWN && seq>0)
    {
        mousex=x;
        mousey=y;
    }
}

//Main function
_declspec (dllexport) double o4v_cam (MyStruct * ms)
{
    IplImage *frame=0, *out=0, *in=0, *gray=0, *graydst=0;
    CvCapture *capture;
    CvFont font;
    cvInitFont(&font, CV_FONT_HERSHEY_SIMPLEX, 0.5, 0.5, 0, 1, CV_AA);

    board_w = 8; // Board width in squares
    board_h = 8; // Board height
    n_boards = 20; // Number of boards
}
```

```

board_total = board_w*board_h;
CvSize board_sz= cvSize(board_w,board_h);

/* initialize camera */
capture = cvCaptureFromCAM( 0 );

/* check camera */
if( !capture ) return 1;

/*get image properties*/
frame = cvQueryFrame( capture );
if ( !frame ) return 1;

/* create window */
cvNamedWindow( "video", CV_WINDOW_AUTOSIZE );
//cvNamedWindow( "cinza", CV_WINDOW_AUTOSIZE );
int mouseParam=5;
//cvSetMouseCallback("video",on_mouse, &mouseParam); // ADD this line
cvSetMouseCallback( "video", on_mouse, 0 );

CvMat* image_points      = cvCreateMat(n_boards*board_total,2,CV_32FC1);
CvMat* object_points    = cvCreateMat(n_boards*board_total,3,CV_32FC1);
CvMat* point_counts     = cvCreateMat(n_boards,1,CV_32SC1);
CvMat* intrinsic_matrix = cvCreateMat(3,3,CV_32FC1);
CvMat* distortion_coeffs = cvCreateMat(4,1,CV_32FC1);

CvPoint2D32f* corners = new CvPoint2D32f[ board_total ];

int successes = 0;
int step = 0;
int lframe = 0;
int nframe = 0;
int seqaux = 0;

while( successes < n_boards ) {

/* get a frame */
frame = cvQueryFrame( capture );

/* always check */
if( !frame ) break;
        nframe++;
/* 'fix' frame */
frame->origin = 0;

        /* make a grey image clone*/
        gray = cvCreateImage( cvSize(frame->width, frame->height), IPL_DEPTH_8U, 1 );
        cvCvtColor( frame, gray, CV_BGR2GRAY );
        //out= cvCloneImage( gray );
        /* Apply canny*/
        //cvCanny(gray, out, 50, 150, 3);

        /* Apply cvFindChessboardCorners*/
        CvPoint2D32f* corners = new CvPoint2D32f[ board_total ];
        int found = cvFindChessboardCorners(gray,board_sz,
corners,&corner_count,CV_CALIB_CB_ADAPTIVE_THRESH | CV_CALIB_CB_FILTER_QUADS );

        cvCvtColor( frame, gray, CV_BGR2GRAY );
        cvFindCornerSubPix(gray, corners, corner_count,cvSize(11,11),cvSize(-1,-
1),cvTermCriteria(CV_TERMCRIT_EPS+CV_TERMCRIT_ITER, 30, 0.1 ));
        cvDrawChessboardCorners(frame, board_sz, corners, corner_count, found);

        // If we got a good board, add it to our data
        if(corner_count == board_total)
        {
                cvShowImage( "Snapshot", frame );
                //show in color if we did collect the image
                step = successes*board_total;
                for( int i=step, j=0; j<board_total; ++i,++j ) {
                        CV_MAT_ELEM(*image_points, float,i,0) = corners[j].x;
                        CV_MAT_ELEM(*image_points, float,i,1) = corners[j].y;
                        CV_MAT_ELEM(*object_points,float,i,0) = (float) j/board_w;
                        CV_MAT_ELEM(*object_points,float,i,1) = (float) (j%board_w);
                        CV_MAT_ELEM(*object_points,float,i,2) = 0.0f;
                }
                CV_MAT_ELEM(*point_counts, int,successes,0) = board_total;
                successes++;
                /*add text*/

```



```

        cvPutText(frame, "Opencv4VBA", cvPoint(10, 30), &font, cvScalar(255, 255, 255, 0));
/* display frame */
cvShowImage( "video", frame );
/* exit if user press 'q' */
key = cvWaitKey( 1 );

    }
}
//Calbrating the camera
CvMat* object_points2 = cvCreateMat(successes*board_total,3,CV_32FC1);
CvMat* image_points2 = cvCreateMat(successes*board_total,2,CV_32FC1);
CvMat* point_counts2 = cvCreateMat(successes,1,CV_32SC1);
//Tranfer the points to matrices
for(int i = 0; i<successes*board_total; ++i)
{
    CV_MAT_ELEM( *image_points2, float, i, 0) = CV_MAT_ELEM( *image_points, float, i, 0);
    CV_MAT_ELEM( *image_points2, float,i,1) = CV_MAT_ELEM( *image_points, float, i, 1);
    CV_MAT_ELEM(*object_points2, float, i, 0) = CV_MAT_ELEM( *object_points, float, i, 0) ;
    CV_MAT_ELEM( *object_points2, float, i, 1)= CV_MAT_ELEM( *object_points, float, i, 1) ;
    CV_MAT_ELEM( *object_points2, float, i, 2)= CV_MAT_ELEM( *object_points, float, i, 2) ;
}
for(int i=0; i<successes; ++i)
{
    CV_MAT_ELEM( *point_counts2, int, i, 0)= CV_MAT_ELEM( *point_counts, int, i, 0);
    //These are all the same number
}
cvReleaseMat(&object_points);
cvReleaseMat(&image_points);
cvReleaseMat(&point_counts);

// Initialize the intrinsic matrix with both the two focal lengths in a ratio of 1.0
CV_MAT_ELEM( *intrinsic_matrix, float, 0, 0 ) = 1.0f;
CV_MAT_ELEM( *intrinsic_matrix, float, 1, 1 ) = 1.0f;

//Calibrate the camera
cvCalibrateCamera2(object_points2, image_points2, point_counts2,
                  cvGetSize( frame ),
                  intrinsic_matrix, distortion_coeffs, NULL, NULL,0 );

//CV_CALIB_FIX_ASPECT_RATIO
CvMat *intrinsic = intrinsic_matrix;
CvMat *distortion = distortion_coeffs;

// Build the undistort map used for all subsequent frames.
IplImage* mapx = cvCreateImage( cvGetSize(frame), IPL_DEPTH_32F, 1 );
IplImage* mapy = cvCreateImage( cvGetSize(frame), IPL_DEPTH_32F, 1 );
cvInitUndistortMap(intrinsic,distortion,mapx,mapy);

cvNamedWindow( "Undistort",CV_WINDOW_AUTOSIZE );
CvPoint* array = new CvPoint[30];
int i;
CvPoint aux= cvPoint(0,0);
for (i=0;i<31;i++)
{
    array[i]=cvPoint(0,0);
    ms->a[i] = 0;
    ms->b[i] = 0;
}
seq=0;
mousex=0;
mousey=0;
while( key != 'q' ) {
/* get a frame */
frame = cvQueryFrame( capture );

    if (aux.x!=mousex || aux.y!=mousey)
    {
        for (i=0;i<31;i++)
        {
            if (array[i].x==0 && array[i].y==0)
            {
                array[i]=cvPoint(mousex, mouseY);
                ms->a[i] = (short)mousex;
                ms->b[i] = (short)mousey;

                aux.x=mousex;
                aux.y=mousey;
                i=31;
            }
        }
    }
}
}

```

```

        };
    }
}
//make lines with mouse pressed points
for (i=1;i<31;i++)
{
    if (array[i].x==1 && array[i].y==1)
    {
        i=i+2;
        if (array[i].x!=0 && array[i].y!=0)
        {
            cvLine(frame, array[i], array[i-1], cvScalar(0,255,0),
1);
        }else{
            if (array[i].x!=0 && array[i].y!=0)
            {
                cvLine(frame, array[i], array[i-1],
cvScalar(0,255,0), 1);
            }else{
                i=31;
            }
        }
    }
}
/* always check */
if( !frame ) break;
/* 'fix' frame */
frame->origin = 0;
IplImage *t = cvCloneImage(frame);
cvShowImage( "video", frame );// Show raw image
cvRemap( t, frame, mapx, mapy ); // Undistort image
cvReleaseImage(&t);
gray = cvCreateImage( cvSize(frame->width, frame->height), IPL_DEPTH_8U, 1 );
cvCvtColor(frame, gray,CV_BGR2GRAY);
/*add text*/
cvPutText(gray, "Opencv4VBA", cvPoint(10, 20), &font, cvScalar(255, 255, 255, 0));
graydst = cvCloneImage( gray );
cvShowImage("Undistort", graydst); // Show corrected image
/* exit if user press 'q' */
key = cvWaitKey( 1 );
}
/* free memory */
cvDestroyWindow( "Snapshot" );
cvDestroyWindow( "Undistort" );
cvDestroyWindow( "video" );
cvReleaseCapture( &capture );
return 1;
cvReleaseImage( &gray );
cvReleaseImage( &out );
cvReleaseImage( &frame );
}

```

ANEXO 6

Módulo do VBA para chamada da DLL e do formulário de controlo da MMC

Inventor - Módulo VBA

```
' VBA2MMC 1.0
' Author Ricardo Fernandes <rrccff@gmail.com>
' PG15168 - Eng. Mecatrónica 2º Ciclo
' Version 1.0 last updated on 20/10/2012
' License Copyright
'
' V0.1 Call of DLL
' V0.2 Return data from a pointer
' V0.3 Test data values
' V0.4 Transform Pixel data to mm
' V0.5 Draw lines of the area coordinates from the MMC
' V0.6 Draw lines of the object coordinates;
' V0.7 Create Userform to control MMC
' V0.8 Connect, send data, read data and disconnect with MMC
' integrating module of Serial interface
' V0.9 Create X,Y and Z movement control
' V1.0 Create MMC initialization process call

Option Explicit
Declare Sub Sleep Lib "kernel32" _
    (ByVal dwMilliseconds As Long) _
'-----
' BEGIN OPENCV4VBA DECLARATIONS
'-----
    Type My_VarUDT
        x(30) As Double
        y(30) As Double
    End Type

'C:\dissertacao\excel-VBA\
Declare Function o4v_cam Lib _
"C:\dissertacao\excel-VBA\opencv4VBA.dll" _
(coordinates As My_VarUDT) As Double
'-----
' END OPENCV4VBA DECLARATIONS
'-----

'Nesta zona está um código desenvolvido por alguém de nickname brumbarchris
'que permite a comunicação série e que pode ser descarregado em:
'http://dev.emcelettronica.com/serial-port-communication-excel-vba

'-----
' BEGIN OPENCV4VBA ROTINES
'-----

Sub OpenCV4VBA()
Dim coordinates As My_VarUDT
Dim coordinatesPeca As My_VarUDT
Dim r As Double
Dim i As Integer, auxx As Double, auxy As Double
Dim PauseTime, Start
Dim Xpixel As Double
Dim Ypixel As Double
'Array to hold received data from OPENCV DLL
For i = 0 To 30
    coordinates.x(i) = 0
    coordinates.y(i) = 0
Next i
'Array to put piece coordinates
For i = 0 To 30
    coordinatesPeca.x(i) = 0
    coordinatesPeca.y(i) = 0
Next i
'Call DLL function
r = o4v_cam(coordinates)
'Get some time to receive pointer
PauseTime = 1 ' Set duration.
Start = Timer ' Set start time.
Do While Timer < Start + PauseTime
    DoEvents ' Yield to other processes.
Loop

If Not TypeOf ThisApplication.ActiveEditObject Is PlanarSketch Then
    MsgBox "A sketch must be active."
    Exit Sub
End If

' Set a reference to the active sketch.
Dim oSketch As PlanarSketch
```

```

Set oSketch = ThisApplication.ActiveEditObject

' Set a reference to the transient geometry collection.
Dim oTransGeom As TransientGeometry
Set oTransGeom = ThisApplication.TransientGeometry

Ypixel = 186.5 / (coordinates.x(3) - coordinates.x(1))
Xpixel = 136.5 / (coordinates.y(2) - coordinates.y(0))

Dim oTrans As Transaction
Set oTrans = ThisApplication.TransactionManager.StartTransaction( _
    ThisApplication.ActiveDocument, _
    "Piece from OpenCV")
Dim oLines(6 To 30) As SketchLine

coordinatesPeca.y(0) = (coordinates.x(6) - coordinates.x(1)) * Ypixel
coordinatesPeca.x(0) = (coordinates.y(5) - coordinates.y(0)) * Xpixel

'Draw piece lines on Inventor Sketch
For i = 7 To 30
    If coordinates.x(i) <> 0 Then
        coordinatesPeca.y(i - 6) = (coordinates.x(i) - coordinates.x(1)) * Ypixel
        coordinatesPeca.x(i - 6) = (coordinates.y(i - 1) - coordinates.y(0)) * Xpixel
        Set oLines(i) = oSketch.SketchLines.AddByTwoPoints( _
            oTransGeom.CreatePoint2d(coordinatesPeca.x(i - 7), coordinatesPeca.y(i - 7)), _
            oTransGeom.CreatePoint2d(coordinatesPeca.x(i - 6), coordinatesPeca.y(i - 6)))
        Else
            i = 31
        End If
    Next i
oTrans.End

'Draw XY MMC area on Inventor Sketch
Dim oRectangleLines As SketchEntitiesEnumerator
Set oRectangleLines = oSketch.SketchLines.AddAsTwoPointRectangle( _
    oTransGeom.CreatePoint2d(0, 0), _
    oTransGeom.CreatePoint2d(136.5, 186.5))

End Sub
'-----
' END OPENCV4VBA ROTINES
'-----

'-----
' BEGIN CALL USERFORM
'-----

Sub showArduinoControl()
    UserForm1.Show
End Sub

'-----
' END CALL USERFORM
'-----

```

Inventor - Controlo do formulário

```

Private Sub CommandButton1_Click()
'Make connection to MMC
Dim intPortID As Integer ' Ex. 1, 2, 3, 4 for COM1 - COM4
Dim lngStatus As Long

intPortID = 1
' Open COM port
lngStatus = CommOpen(intPortID, "COM" & CStr(intPortID), _
    "baud=9600 parity=N data=8 stop=1")
TextBox3.Value = lngStatus & " - " & "connected"
End Sub

Private Sub CommandButton10_Click()
'-Z direction movement
Dim intPortID As Integer ' Ex. 1, 2, 3, 4 for COM1 - COM4
Dim lngStatus As Long
Dim strData As String

intPortID = 1
TextBox5.Value = "-" & Right(TextBox5.Value, 5)
strDataaux = "G00X+00000Y+00000Z" & TextBox5.Value
TextBox3.Value = ""
For i = 1 To 24
    strData = Mid(strDataaux, i, 1)
    'Application.Wait (Now + 0.000001)'EXCEL ONLY

```

```

        Sleep 10
        lngStatus = CommWrite(intPortID, strData)
        TextBox3.Value = TextBox3.Value & strData
    Next i
    'Application.Wait (Now + 0.00001)'EXCEL ONLY
    Sleep 10
    TextBox4.Value = ""
    lngStatus = CommRead(intPortID, strData, 80)
    TextBox4.Value = strData
End Sub

Private Sub CommandButton2_Click()
    'Read incoming data from MMC
    Dim intPortID As Integer ' Ex. 1, 2, 3, 4 for COM1 - COM4
    Dim lngStatus As Long
    Dim strData As String

    intPortID = 1
    TextBox4.Value = ""
    lngStatus = CommRead(intPortID, strData, 80)
    TextBox4.Value = strData
End Sub

Private Sub CommandButton4_Click()
    '+X direction movement
    Dim intPortID As Integer ' Ex. 1, 2, 3, 4 for COM1 - COM4
    Dim lngStatus As Long
    Dim strData As String

    intPortID = 1
    TextBox1.Value = "+" & Right(TextBox1.Value, 5)
    strDataaux = "G00X" & TextBox1.Value & "Y+00000Z+00000"
    TextBox3.Value = ""
    'lngStatus = CommWrite(intPortID, strData)
    For i = 1 To 24
        strData = Mid(strDataaux, i, 1)
        'Application.Wait (Now + 0.00001)'EXCEL ONLY
        Sleep 10
        lngStatus = CommWrite(intPortID, strData)
        TextBox3.Value = TextBox3.Value & strData
    Next i
    'Application.Wait (Now + 0.00001)'EXCEL ONLY
    Sleep 10
    TextBox4.Value = ""
    lngStatus = CommRead(intPortID, strData, 80)
    TextBox4.Value = strData
End Sub

Private Sub CommandButton3_Click()
    'Disconnect from MMC
    Dim intPortID As Integer ' Ex. 1, 2, 3, 4 for COM1 - COM4

    intPortID = 1
    Call CommClose(intPortID)
    TextBox3.Value = "disconnected"
End Sub

Private Sub CommandButton5_Click()
    '-Y direction movement
    Dim intPortID As Integer ' Ex. 1, 2, 3, 4 for COM1 - COM4
    Dim lngStatus As Long
    Dim strData As String

    intPortID = 1
    TextBox2.Value = "+" & Right(TextBox2.Value, 5)
    strDataaux = "G00X+00000Y" & TextBox2.Value & "Z+00000"
    TextBox3.Value = ""
    For i = 1 To 24
        strData = Mid(strDataaux, i, 1)
        'Application.Wait (Now + 0.00001)'EXCEL ONLY
        Sleep 10
        lngStatus = CommWrite(intPortID, strData)
        TextBox3.Value = TextBox3.Value & strData
    Next i
    'Application.Wait (Now + 0.00001)'EXCEL ONLY
    Sleep 10
    TextBox4.Value = ""
    lngStatus = CommRead(intPortID, strData, 80)
    TextBox4.Value = strData
End Sub

```

```

Private Sub CommandButton6_Click()
'Send command for MMC initialization
Dim intPortID As Integer ' Ex. 1, 2, 3, 4 for COM1 - COM4
Dim lngStatus As Long
Dim strData As String

intPortID = 1
strDataaux = "INIX+00000Y+00000Z+00000"
TextBox3.Value = "INIX+00000Y+00000Z+00000"
For i = 1 To 24
    strData = Mid(strDataaux, i, 1)
    'Application.Wait (Now + 0.000001)'EXCEL ONLY
    Sleep 10
    lngStatus = CommWrite(intPortID, strData)
    TextBox3.Value = TextBox3.Value & strData
Next i
'Application.Wait (Now + 0.000002)'EXCEL ONLY
Sleep 10
TextBox4.Value = ""
lngStatus = CommRead(intPortID, strData, 80)
TextBox4.Value = strData
End Sub

Private Sub CommandButton7_Click()
'+Z direction movement
Dim intPortID As Integer ' Ex. 1, 2, 3, 4 for COM1 - COM4
Dim lngStatus As Long
Dim strData As String

intPortID = 1
TextBox5.Value = "+" & Right(TextBox5.Value, 5)
strDataaux = "G00X+00000Y+00000Z" & TextBox5.Value
TextBox3.Value = ""
For i = 1 To 24
    strData = Mid(strDataaux, i, 1)
    'Application.Wait (Now + 0.000001)'EXCEL ONLY
    Sleep 10
    lngStatus = CommWrite(intPortID, strData)
    TextBox3.Value = TextBox3.Value & strData
Next i
'Application.Wait (Now + 0.00001)'EXCEL ONLY
Sleep 10
TextBox4.Value = ""
lngStatus = CommRead(intPortID, strData, 80)
TextBox4.Value = strData
End Sub

Private Sub CommandButton8_Click()
'-X direction movement
Dim intPortID As Integer ' Ex. 1, 2, 3, 4 for COM1 - COM4
Dim lngStatus As Long
Dim strData As String

intPortID = 1
TextBox1.Value = "-" & Right(TextBox1.Value, 5)
strDataaux = "G00X" & TextBox1.Value & "Y+00000Z+00000"
TextBox3.Value = ""
lngStatus = CommWrite(intPortID, strData)
For i = 1 To 24
    strData = Mid(strDataaux, i, 1)
    'Application.Wait (Now + 0.000001) 'EXCEL ONLY
    Sleep 10
    lngStatus = CommWrite(intPortID, strData)
    TextBox3.Value = TextBox3.Value & strData
Next i
'Application.Wait (Now + 0.00001) 'EXCEL ONLY
Sleep 10
TextBox4.Value = ""
lngStatus = CommRead(intPortID, strData, 80)
TextBox4.Value = strData
End Sub

Private Sub CommandButton9_Click()
'+Y direction movement
Dim intPortID As Integer ' Ex. 1, 2, 3, 4 for COM1 - COM4
Dim lngStatus As Long
Dim strData As String

intPortID = 1
TextBox2.Value = "-" & Right(TextBox2.Value, 5)
strDataaux = "G00X+00000Y" & TextBox2.Value & "Z+00000"

```



```
TextBox3.Value = ""
lngStatus = CommWrite(intPortID, strData)
For i = 1 To 24
    strData = Mid(strDataaux, i, 1)
    'Application.Wait (Now + 0.000001)'EXCEL ONLY
    Sleep 10
    lngStatus = CommWrite(intPortID, strData)
    TextBox3.Value = TextBox3.Value & strData
Next i
'Application.Wait (Now + 0.00001)'EXCEL ONLY
Sleep 10
TextBox4.Value = ""
lngStatus = CommRead(intPortID, strData, 80)
TextBox4.Value = strData
End Sub
```

