**UNIVERSIDADE FEDERAL DO TOCANTINS**

**DESENVOLVIMENTO DE UM BRAÇO ROBOTICO UTILIZANDO SERVOMOTERES E GYROSCOPIO CONTROLADOS PELA PLATAFORMA DE PROTOTIPAGEM ARDUINO**

IVAN VIEIRA ([ivanvieira.10@hotmail.com](mailto:ivanvieira.10@hotmail.com))

JUCELINO MENEZEZ ([jussamenezes@gmail.com](mailto:jussamenezes@gmail.com))

WILLIAN LUNARDI ([willlunardi2@hotmail.com](mailto:willlunardi2@hotmail.com))

RODRIGO DANTAS ([zeh\_b@hotmail.com](mailto:zeh_b@hotmail.com))

**PALMAS – TOCANTINS – 2013**

Sumário

[Introdução 3](#_Toc353923704)

[Objetivo 3](#_Toc353923705)

[Hardware 3](#_Toc353923706)

[Arduino 4](#_Toc353923707)

[A placa Arduino 4](#_Toc353923708)

[5](#_Toc353923709)

[O software Arduino 5](#_Toc353923710)

[Servo Motor 5](#_Toc353923711)

[Módulo GY-521 (MPU6050) 7](#_Toc353923712)

[Chassis 8](#_Toc353923713)

[Instalação e Configuração do Arduino 8](#_Toc353923714)

[Instalação de drives sem a assinatura digital 9](#_Toc353923715)

[Instalar o Arduino Uno no Windows 8 sem desativar a assinatura de driver 11](#_Toc353923716)

[Ferramentas Auxiliares 12](#_Toc353923717)

[Prototipagem e Desenvolvimento 13](#_Toc353923718)

[Código-Fonte Desenvolvido no Projeto 15](#_Toc353923719)

[Bibliografia 17](#_Toc353923720)

# Introdução

A área de robótica é relativamente nova e foi criada para proporcionar soluções adequadas a algumas dualidades técnicas, onde a atuação humana é difícil ou ate mesmo impossível. A base da robótica consiste na junção dos conceitos científicos da mecânica, eletrônica e programação, onde a grande necessidade por inovação técnica no mundo moderno impulsiona de forma rápida o avanço tecnológico nessa área. Hoje se encontram disponíveis no mercado muitos modelos de computadores e dispositivos específicos para a robótica, com motores, sensores, ligas metálicas especiais, que são utilizados para diversos benefícios em algumas áreas de atuação humana. A robótica exerce certo fascínio entre os jovens e, justamente por conta disso, um novo campo de aplicação, que busca usar a robótica como instrumento auxiliar no processo de ensino-aprendizagem, tem despontado na atualidade. Muitos resultados satisfatórios têm sido obtidos na Educação quando se tem incluído o robô como uma ferramenta pedagógica. Devido a este fato, muitas escolas, com o apoio de universidades, estão utilizando a robótica como um meio dinâmico de aprendizagem transdisciplinar, chamado por alguns autores de robótica pedagógica ou robótica educacional e consiste basicamente na aprendizagem por meio da montagem de sistemas constituídos por robôs.

## Objetivo

Apresentaremos uma alternativa para o desenvolvimento de um braço robótico utilizando servomotores e giroscópio, ambos controlados pela plataforma de prototipagem Arduino.

O projeto apresentado também visa o desenvolvimento de um kit básico de robótica para que o mesmo possa ser utilizado como apoio em processos educacionais dos cursos tecnológicos de computação.

O kit utilizará os mais novos recursos computacionais didáticos, mas também reaproveitará ferramentas há muito tempo conhecidas pela robótica.

# Hardware

A parte física do projeto será constituída de:

* 1 Arduino Uno R3 com microcontrolador ATmega328
* 1 Kit de robótica para construção do chassis do braços robótico
* 3 servo motores com liberdade de 180º
* 1 módulo GY-521 (MPU5060 - Giroscópio + Acelerômetro)

## 

## Arduino

O Arduino foi desenvolvido para ser uma plataforma de fácil utilização onde seja possível que pessoas com nenhum conhecimento de hardware e programação possam desenvolver projetos onde objetos possam responder a diversos sinais como de luz, toque e movimento. O Arduino é utilizado em vários programas educacionais pelo mundo, particularmente por desenvolvedores e artistas que querem criar facilmente protótipos, mas que não necessitem um profundo entendimento sobre detalhes técnicos por trás de suas criações (Margolis, 2011).

### A placa Arduino

A placa Arduino é uma interface onde podem ser conectados diversos tipos de componentes como sensores, LEDS, LCD's, motores, etc. Esta placa recebe sinais elétricos dos componentes conectados e atua de acordo com o programa o qual ela esta executando. Como esta placa é desenvolvida livremente, existem diversos modelos o ciais, dentre estes o mais comum são os encontrados com porta USB apesar de existirem versões com portas paralelas.

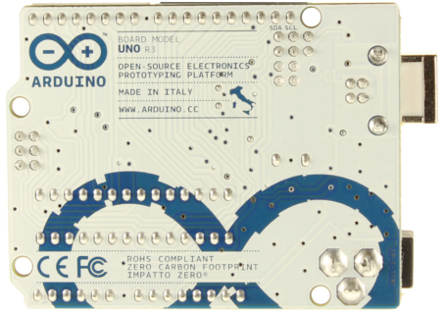
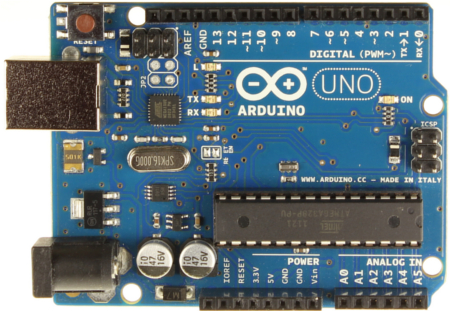


Figura 1: Arduino Uno R3 - Frente e Fundo

|  |  |
| --- | --- |
| **Microcontroller** | ATmega328 |
| **Operating Voltage** | 5V |
| **Input Voltage (recommended)** | 7-12V |
| **Input Voltage (limits)** | 6-20V |
| **Digital I/O Pins** | 14 (of which 6 provide PWM output) |
| **Analog Input Pins** | 6 |
| **DC Current per I/O Pin** | 40 mA |
| **DC Current for 3.3V Pin** | 50 mA |
| **Flash Memory** | 32 KB (ATmega328) of which 0.5 KB used by bootloader |
| **SRAM** | 2 KB (ATmega328) |
| **EEPROM** | 1 KB (ATmega328) |
| **Clock Speed** | 16 MHz |

### 

Tabela 1: Especificações Técnicas Arduino Uno R3 (Arduino)

### O software Arduino

Para sua utilização, a plataforma Arduino possui um software chamado Arduino no qual podem ser criados projetos simples os quais são chamados de sketches. Além disto, o software Arduino também envia o programa para o hardware no qual o programa será executado.

## Servo Motor

Servo motores são dispositivos de malha fechada, ou seja, recebem um sinal de controle; verificam a posição atual; atuam no sistema indo para a posição desejada. Em contraste com os motores contínuos que giram indefinidamente, o eixo dos servo motores possui a liberdade de até cerca de 180º graus mas são precisos quanto a posição.



Figura 2: Servo Motor

Os servo motores são divididos por três componentes básicos: sistema atuador, sensor e circuito de controle.

* **Sistema Atuador** é constituído por um motor elétrico, na maioria dos casos com motores de corrente contínua, embora também possamos encontrá-lo de corrente alternada. Onde também está presente um conjunto de engrenagens que forma uma caixa de redução com uma relação bem longa o que ajuda a amplificar o torque.
* **Sensor** normalmente é um potenciômetro acoplado ao eixo do servo, pois com o valor de sua resistência elétrica determinaremos a posição angular do eixo.
* **Circuito de Controle** - O circuito de controle é formado por componentes eletrônicos discretos ou circuitos integrados e geralmente é composto por um oscilador e um controlador que recebe um sinal do sensor (posição do eixo) e o sinal de controle e aciona o motor no sentido necessário para posicionar o eixo na posição desejada.

Servos possuem três fios de interface, dois para alimentação e um para o sinal de controle. O sinal de controle utiliza a modulação por largura de pulso que possui três características básicas: Largura mínima, largura máxima e taxa de repetição.

A largura do pulso de controle determinará a posição do eixo:

* **Largura máxima** equivale ao deslocamento do eixo em +45º da posição central;
* **Largura mínima** equivale ao deslocamento do eixo em -45º;
* Outras larguras determinam sua posição proporcionalmente;

## Módulo GY-521 (MPU6050)

A família ™ MPU-6000/MPU-6050 integra dispositivos de detecção de movimento em 6 eixos projetadas para baixo consumo de energia, baixo custo de produção e proporcionam alto desempenho em smartphones, tablets e dispositivos que utilizam navegação inercial. (Playground)

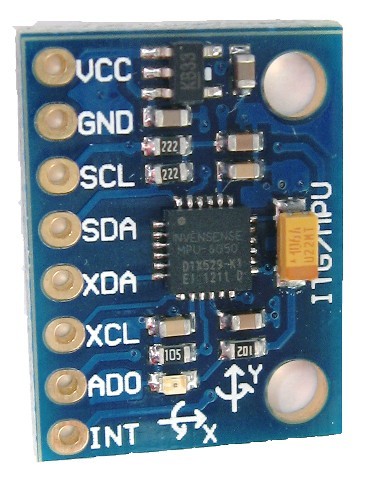


Figura 3: Módulo GY-521 (MPU6050)

Os dispositivos MPU-6000/6050 combinam um giroscópio de 3 eixos e um acelerómetro de 3 eixos no mesmo CI juntamente com um processador de Movimento Integrado Digital ™ (DMP ™) com capacidade de processamento de algoritmos complexos de 9 eixos (“MotionFusion” - com acesso a magnetômetros externos ou outros sensores através de uma comunicação auxiliar - Protocolo I²C -, permitindo que os dispositivos se comuniquem em um conjunto completo de dados sem a intervenção do processador do sistema). (InvenSense)

A precisão de ambos os sensores pode ser calibrada para movimentos rápidos ou lentos, os sensores disponibilizam essa mudança no nível de programação para o usuário. Para o giroscópio as escalas são de ± 250, ± 500, ± 1000 e 2000 ± ° / seg (PD) e para o acelerômetro as escalas são de intervalo de ± 2 g, ± 4 g, ± 8 g, e ± 16 g (InvenSense).

## Chassis

O Robix RCS-6 é um dos diversos kits de robótica comercializados e voltado ao ensino em escolas e universidades. Começou a ser desenvolvido pela empresa norte americana Robix (Advanced Design Inc.) no ano de 1992.

Através das peças disponíveis no kit, é possível montar 11 protótipos robóticos diferentes, sendo que para cada modelo é fornecido o respectivo manual de montagem e configuração. o kit Robix RCS-6 é composto por:

* Perfil de alumínio para sustentação;
* Base de mármore para sustentar a "torre" na qual o braço é fixado;
* Servo motores responsáveis pela movimentação das articulações;
* Mecanismo de pinça, que serve como mão para pegar objetos;
* Circuito controlador, que promove a comunicação entre o robô e o computador;
* Manual com informação de como montar os protótipos;
* Componentes menores, como fios, parafusos, molas, etc.

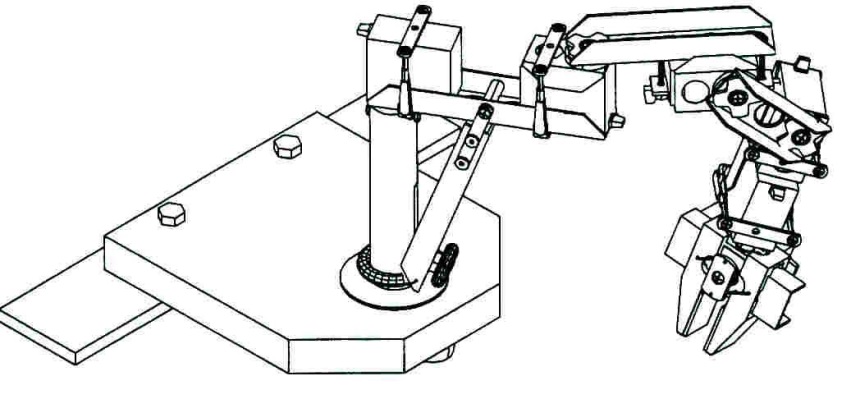
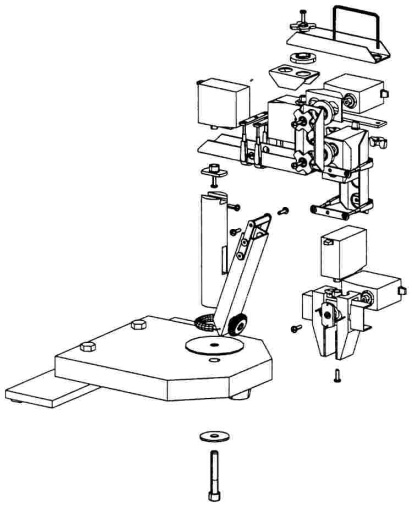


Figura 4: Protótipo de um braço robótico utilizando o kit Robix

# Instalação e Configuração do Arduino

O grande problema da instalação do Arduino é a quantidade de Sistemas Operacionais existentes. Temos Windows XP, Vista, 7, 8, Mac OS e finalmente uma enorme quantidade de distribuições Linux. Além de tudo isso, temos versões 32 e 64 bits. Isso sem falar que existem várias distribuições do Arduino, como UNO, Duemilanove, Decimila, Mega etc.

O projeto foi desenvolvido na plataforma Windows. No Windows 8 encontraremos uma grande dificuldade para instalar drives de terceiros que não possuem assinatura digital. Para resolver esse problema estão disponíveis dois métodos: *Habilitar a instalação de drivers no Windows 8 sem a assinatura digital* ou *instalar o Arduino Uno no Windows 8 sem desativar a assinatura de driver ou reiniciar*.

Vamos então partir para a instalação do Arduino no computador. Precisamos de duas coisas:

* Instalar o programa de desenvolvimento (conhecido por IDE).
* Instalar o driver USB

Baixe e instala a IDE (Ambiente Integrado de Desenvolvimento). Para isso, basta acessar o seguinte endereço e fazer o download específico para seu sistema operacional: [*http://arduino.cc/en/Main/Software*](http://arduino.cc/en/Main/Software)

Na página do link acima, basta baixar a versão denominada apenas "Windows". Não existe instalador, o que você vai baixar é um arquivo compactado, do tipo ZIP. Para abrir este arquivo e descompactar a IDE do Arduino, você precisará de um programa para manipulação de arquivos comprimidos . Algumas versões de Windows lidam com arquivos ZIP de forma transparente.

Descompacte esse arquivo chamado "*arduino-1.0.3-windows*" (é a versão atual utilizada no desenvolvimento do trabalho) em uma pasta com o mesmo nome, de preferência no drive "*C:*", para que o caminho fique assim: "*C:\arduino-1.0.3-windows\arduino-1.0.3*". Dentro dessa pasta, você verá que existe um arquivo executável chamado "*arduino.exe*". Este é o ponto de entrada do programa principal do Arduino, ou seja, a IDE ("*C:\arduino-1.0.3-windows\arduino-1.0.3\arduino.exe).*

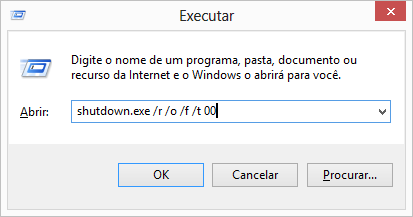
## Instalação de drives sem a assinatura digital

### 

Este método desabilita algumas opções de segurança do Windows 8 através dos passos abaixo:

**ATENÇÃO**: Antes de prosseguir, certifique-se que você salvou todos os documentos que estão abertos em sua máquina. Pois sua máquina será reiniciada.

1. Pressione as teclas "*windows*" e "*R*" simultaneamente.
2. Copie e cole o seguinte comando: *shutdown.exe /r /o /f /t 00*



1. Uma série de telas azuis será apresentada, na primeira, selecione “*Solução de Problemas*”.
2. A seguir, selecione “*Opções Avançadas*”.
3. Selecione “*Configurações de Inicialização*”.
4. Clique em “*Reiniciar*”.
5. O computador será reinicializado. Na tela apresentada, pressione o botão “*7*” do seu teclado, referente a “*Desabilitar Imposição de Assinatura de Driver*”.
6. Faça agora o procedimento normal de instalação do driver.
7. Conecte a placa via cabo USB no computador. O assistente de instalação deve aparecer. Clique em "*Avançar*" até que o Windows diga que não conseguiu instalar o driver (não se preocupe, é assim mesmo). Clique em "*Concluir*" para dispensar o assistente.
8. Vá para "*Painel de Controle*".
9. No painel de controle, clique no ícone "*Sistema e Segurança*". Na tela que aparecer, selecione "*Hardware e Sons*" e depois procure por "*Gerenciador de Dispositivos*".
10. Procure pela opção "*Portas (COM & LPT)".* Clique no sinal de adição (+) correspondente a essa opção e uma entrada chamada "*Arduino UNO (COMxx)*" deve aparecer. Pode ser que você não encontre essa entrada na opção "*Portas*", se for esse o caso, procure por um "*Dispositivo Desconhecido*" na opção "*Outros Dispositivos*". Se você tiver mais de um dispositivo desconhecido, ou tiver dúvidas nesse ponto, desligue e religue o cabo do Arduino e preste atenção qual a entrada que some quando você desliga o Arduino e que reaparece quando você o reconecta.
11. Clique então na entrada descrita no passo 12 com o botão direito do mouse para aparecer o menu de contexto. Selecione a opção "*Atualizar driver...*".
12. Na tela que aparecer, selecione a opção "*Instalar de uma lista ou local específico (Avançado)*" e clique em "*Avançar*".
13. Na próxima tela, verifique que a opção selecionada é a "*Procurar melhor driver nestes locais*". E nas caixas de seleção logo abaixo essa opção, desmarque a primeira, denominada "*Pesquisar mídia removível (disquete, CD-ROM...)*" e marque a segunda "*Incluir este local na pesquisa:*". Quando você fizer isso, o campo de pesquisa e o botão "*Pesquisar*" ficarão disponíveis.
14. Clique então no botão "*Pesquisar*" e localize a pasta "*Drivers*" que se encontra no local " *C:\arduino-1.0.3-windows\arduino-1.0.3\drivers\*" (se você seguiu os passos de instalação desse tutorial). Atenção: NÃO escolha a sub-pasta "*FTDI USB Drivers*". Clique em "*OK*" para concluir a escolha da pasta.
15. Agora clique em "*Avançar*" até concluir a instalação do driver. Pode ser que o Windows apresente uma tela dizendo que o driver pode não ser confiável.
16. Uma mensagem de erro será apresentada. Clique em “*Instalar este software de driver mesmo assim*”.



1. Clique em "*Concluir*" para finalizar a instalação.

## Instalar o Arduino Uno no Windows 8 sem desativar a assinatura de driver

Uma forma alternativa e que foi utilizada no desenvolvimento do projeto foi a instalação do Arduino no Windows 8 sem a necessidade de desativar a assinatura de driver. Ele funciona porque o Arduino Uno usa o driver de dispositivo *Usbser.sys* e é mesmo que acontece com alguns drivers incluídos no Windows 8.

1. Abra o “*Gerenciador de Dispositivos*”.
2. Entre na opção "*Portas*", se for esse o caso, procure por um "*Dispositivo Desconhecido*" na opção "*Outros Dispositivos*".
3. Clique no sinal de adição (+) correspondente a essa opção e uma entrada chamada "*Arduino UNO*" deve aparecer.
4. Clique então na entrada descrita no passo 3 com o botão direito do mouse para aparecer o menu de contexto. Selecione a opção "*Atualizar driver...*" e na tela que aparecer clique em “*Procurar software de drive no computador*”.
5. Na tela seguinte, selecione “*Permitir que eu escolha em uma lista de dispositivos no computador*”.
6. Na tela “Atualizar driver – Arduino Uno” espere ate que uma lista de dispositivos seja carregada. Então rola a barra vertical ate localizar a opção “*Modens*”
7. Na tela “Instalar novo modem” espere outra lista ser carregada. No lado esquerdo selecione “*Compaq*”, depois no lado direito selecione o modelo “*Ricochet Wireless USB Modem*” e em seguida clique em “*Avançar*”. Clique em “*Sim*”.
8. Feche a tela de confirmação da instalação dos drives e volte para o “*Gerenciador de Dispositivos*”. Acesse a opção “*Modems*”. Localize o dispositivo recentemente instalado “*Ricochet Wireless USB Modem*” clique com o botão direito do mouse sobre ele e selecione “Propriedades”.
9. Na aba “*Modem*” selecione a opção “*Velocidade Máxima da Porta*” e atualize a velocidade para *9600.*
10. Em Seguida acesse a aba “*Avançado*” e clique em *“Configurações Avançadas da Porta”*.
11. Desmarque a opção “Usar buffer FIFO” e em “*Numero da Porta COM*” deixe selecionado a porta “COM5”. De “*OK*” e em seguida “*OK*” novamente.

Pronto. Agora podemos programar na IDE Arduino e realizar a gravação do código responsável pelo controle dos diapositivos conectados ao Arduino.

# Ferramentas Auxiliares

Ferramentas auxiliares foram de grande importância para o inicio do projeto. Com elas conseguimos avaliar custos, definir os componentes corretos para o projeto e principalmente criar protótipos e executar testes com eles, com isso não existe riscos de queimar um componente ou de fazer aquisição de hardware desnecessário para o projeto.

As ferramentas auxiliares utilizadas foram:

* Ferramenta Eagle
* Ferramenta Fritzing

As duas ferramentas são muito semelhantes, tem como objetivo apoiar designers, artistas e pesquisadores de gerar protótipos físicos e traze-los para o produto real.

Fritzing é uma iniciativa open-source enquanto Eagle é uma ferramenta proprietária. Fritzing permite aos usuários documentar os projetos Arduino e outros eletrônicos baseados em protótipos, enquanto o Eagle fornece recursos mais avançados para usuários mais experientes.

# Prototipagem e Desenvolvimento

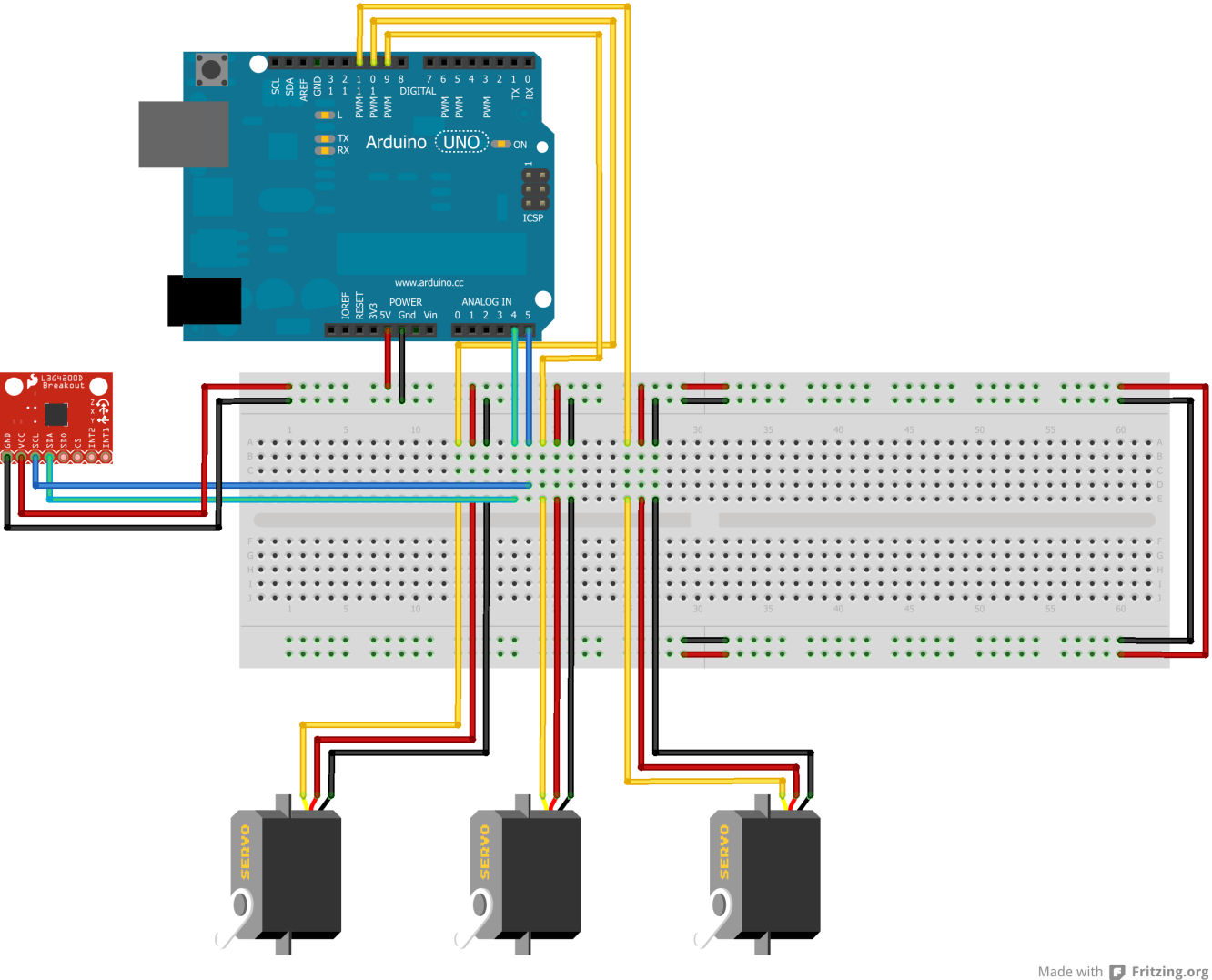
 A imagem a baixo ilustra o primeiro protótipo desenvolvido no Fritzing com o objetivo de estudar como os componentes se conectam.

Figura 5: Protótipo do projeto na Protoboard

Os servos 1, 2 e 3 são conectados nas saídas PWM 09, 10 e 11 respectivamente. São alimentos pela saída Vcc do próprio Arduino, sem a necessidade de alimentação extra.

O modulo GY-521 (Acelerômetro+Giroscópio) esta conectado ao Arduino Uno no seguinte esquema:

|  |  |  |
| --- | --- | --- |
| **GY-521** |  | **Arduino** |
| VCC | >> | +5V |
| GND | >> | Gnd |
| SDA | >> | Entrada analógica pin A4 (SDA) |
| SCL | >> | Entrada analógica pin A5 (SCL) |

Esquema de conexão do modulo GY-521 ao Arduino Uno

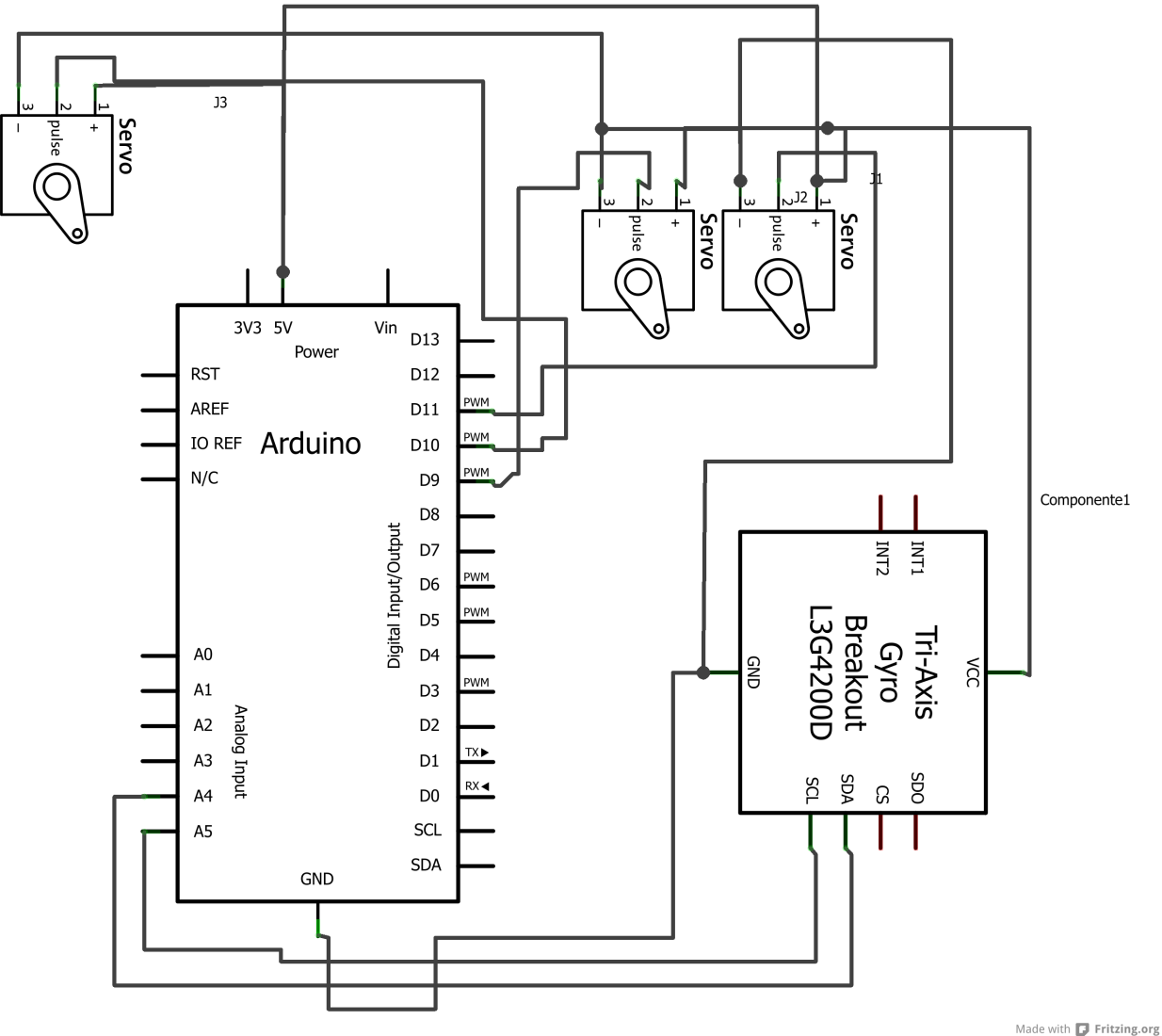


Figura 6: Esquema do Circuito Integrado do Protótipo

Devido à falta de boa documentação disponível publicamente sobre o funcionamento interno do modulo GY-521, todas as informações relacionadas com a DMP tem surgido de engenharia reversa e de análise de sinais I2C.  Este esforço de descoberta resultou em várias bibliotecas que auxiliam na conexão e aquisição de dados da MUP6050. Atualmente, o código-fonte disponível só fornece a configuração do dispositivo básico e a leituras de aceleração/gryo.

A biblioteca utilizada e responsável por se comunicar com dispositivos I2C é chamada de i2cdevlib. i2cdevlib é uma coleção de classes uniformes e bem documentada para fornecer interfaces simples e intuitivas para dispositivos I2C. Cada dispositivo é construído para fazer uso genérico da classe "I2Cdev", oferecendo uma maneira simples de modificar apenas uma classe e portar o código de comunicação I2C para diferentes plataformas (Arduino, PIC, etc.) As classes dos dispositivos são projetados para fornecer uma cobertura completa de todas as funcionalidades descritas por documentação de cada dispositivo (Library).

Baixe a versão da i2cdevlib para Arduino e GY-521 no link: [*https://github.com/jrowberg/i2cdevlib*](https://github.com/jrowberg/i2cdevlib)

Extraia as pastas “*MPU6050*” e “*I2Cdev*” para a pasta onde estão localizadas as bibliotecas utilizadas pelo Arduino (caminho padrão utilizado no projeto “*C:\arduino-1.0.3-windows\arduino-1.0.3\libraries*”).

## Código-Fonte Desenvolvido no Projeto

**#include** "Wire.h"

**#include** "I2Cdev.h"

**#include** "MPU6050.h"

**#include** <Servo.h>

Servo servo1;

Servo servo2;

Servo servo3;

MPU6050 accelgyro;

int16\_t ax, ay, az;

int16\_t gx, gy, gz;

//Usado para calcular o angulo - Variaveis do acelerometro

**double** accXangle;

**double** accYangle;

**double** accZangle;

//Usado para calcular o angulo - Variaveis do giroscopio

**double** gyroXangle = 180;

**double** gyroYangle = 180;

**double** gyroZangle = 180;

uint32\_t timer;

**void** **setup**() {

Wire.begin();

// Inicializando a comunicação serial

// funciona em 8MHz ou em 16MHz

Serial.begin(38400);

// Iniciando dispositivos

Serial.println("Inicializando cominicação I2C...");

accelgyro.initialize();

// Testando a conexão com a MPU6050

Serial.println("Testando a conexão com MPU6050...");

Serial.println(

accelgyro.testConnection() ?

"MPU6050 conectada com sucesso" :

"Falha na conexão com a MPU6050");

servo1.attach(11);

servo2.attach(10);

servo3.attach(9);

timer = micros();

}

**void** **loop**() {

// Fazendo a leitura de conexão com a MPU6050

accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

// Calcular os angulos com base nos sensores do acelerometro

accXangle = (atan2(ax, az) + PI) \* RAD\_TO\_DEG;

accYangle = (atan2(ay, az) + PI) \* RAD\_TO\_DEG;

accZangle = (atan2(ax, ay) + PI) \* RAD\_TO\_DEG;

**double** gyroXrate = ((**double**) gx / 131.0);

**double** gyroYrate = -((**double**) gy / 131.0);

**double** gyroZrate = -((**double**) gz / 131.0);

//###################### Calcular o ângulo de giro sem qualquer filtro #########################

gyroXangle += gyroXrate \* ((**double**) (micros() - timer) / 1000000);

gyroYangle += gyroYrate \* ((**double**) (micros() - timer) / 1000000);

gyroZangle += gyroZrate \* ((**double**) (micros() - timer) / 1000000);

servo1.write(gyroYangle);

servo2.write(gyroXangle);

servo3.write(gyroZangle);

timer = micros();

// A taxa de amostras máxima do acelerometro é de 1KHz

delay(1);

//Angulo Giroscopio x/y/z

Serial.print(gyroXangle);

Serial.print("\t");

Serial.print(gyroYangle);

Serial.print("\t");

Serial.print(gyroZangle);

Serial.print("\t");

Serial.print("\n");

}

Vídeo compartilhado no YouTube mostrando o resultado do projeto:

<http://www.youtube.com/watch?v=8wvEauCCvio&list=FLzhxRiZNCou9Nycyc7euhoQ>

# Bibliografia

Advanced Design Inc. (s.d.). *ROBIX*. Acesso em 02 de 2013, disponível em http://www.robix.com/

Arduino. (s.d.). *arduino.cc*. Acesso em http://arduino.cc/ de 02 de 2013

InvenSense. (s.d.). *InvenSense*. Acesso em 02 de 2013, disponível em http://www.invensense.com

Library, I. D. (s.d.). *I2C Device Library*. Acesso em 02 de 2013, disponível em i2cdevlib: http://www.i2cdevlib.com/

Margolis, M. (2011). *Arduino Cookbook.* O'Reilly Media; Second Edition edition (December 30, 2011).

Playground, A. (s.d.). *Arduino Playground*. Acesso em 02 de 2013, disponível em http://playground.arduino.cc

# 