



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE ENSINO SUPERIOR DO SERIDÓ
DEPARTAMENTO DE COMPUTAÇÃO E TECNOLOGIA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO
MICROCONTROLADORES

**RELATÓRIO DA SEGUNDA UNIDADE:
BRAÇO MECÂNICO**

**ALCINDO LOPES DE FARIAS JUNIOR
BRUNO DANTAS DE ARAÚJO
JACKSON SILVA PEREIRA SOBRINHO
JORGE JÚNIO DA SILVA
ROBERTO DE SOUSA ROCHA
ROBSON AGRIPINO DA SILVA**

Caicó - RN

2018

SUMÁRIO

	Lista de Figuras	ii
1	INTRODUÇÃO	1
1.1	Objetivos Gerais	1
1.2	Objetivos Específicos	1
2	TECNOLOGIAS UTILIZADAS	2
2.1	Arduino	2
2.2	Módulo I2C PCA9685 16-Channel 12-bit PWM	3
2.3	Braço Mecânico de Acrílico	4
3	PROCEDIMENTOS	6
3.1	Instalando bibliotecas necessárias	6
3.1.1	meArm Adafruit	6
3.1.2	Adafruit PWM servo Driver Library	6
3.2	Montagem	6
3.3	Código Fonte	7
4	RESULTADOS OBTIDOS	8
4.1	Funcionamento	8
4.2	Dificuldades	9
	REFERÊNCIAS	11

Lista de Figuras

Figura 2.1 – Anatomia do Arduino	2
Figura 2.2 – Módulo I2C PCA9685	4
Figura 2.3 – Módulo I2C PCA9685 visto de cima	4
Figura 2.4 – Braço robótico em acrílico	5
Figura 3.1 – Montagem do projeto	7
Figura 4.1 – Funcionamento 1	8
Figura 4.2 – Funcionamento 2	9
Figura 4.3 – Funcionamento 3	9

1 Introdução

Com o avanço tecnológico dos últimos tempos, tornou-se notório a introdução progressiva da robótica em nosso cotidiano. Vários casos são vistos no dia a dia como por exemplo: sistemas, casas, robôs, esteiras, todos automatizados, entre outros. Em diversos casos são utilizados plataformas de prototipagem como por exemplo os Arduínos.

A partir disso, os usuários estão se motivando a aprenderem mais sobre essas plataformas, fazendo com que contribua demasiadamente para o aumento dos usuários de microcontroladores, componentes analógicos e digitais, principalmente pelo fato de que sua arquitetura é de fácil compreensão e manuseio, auxiliando na transmissão dos conhecimentos teóricos do tema com a realização de experiências práticas.

Este relatório apresentará um dos projetos realizados na segunda unidade da disciplina de Microcontroladores, que consiste em montar um braço mecânico de acrílico, calibrá-lo e fazer o mesmo funcionar de maneira satisfatória usando um Arduino.

1.1 Objetivos Gerais

Fazer o braço mecânico de acrílico funcionar de maneira satisfatória de forma que o mesmo se mova como o usuário desejar dentro dos seus limites.

1.2 Objetivos Específicos

- Estudar manuais de montagem do braço mecânico de acrílico;
- Realizar a montagem do braço mecânico de acrílico com base no manual;
- Pesquisar e estudar bibliotecas úteis para o projeto;
- Realizar a montagem de hardware necessária;
- Enviar comandos para o braço mecânico de acrílico via Serial;
- Tratar a chegada de interrupção.

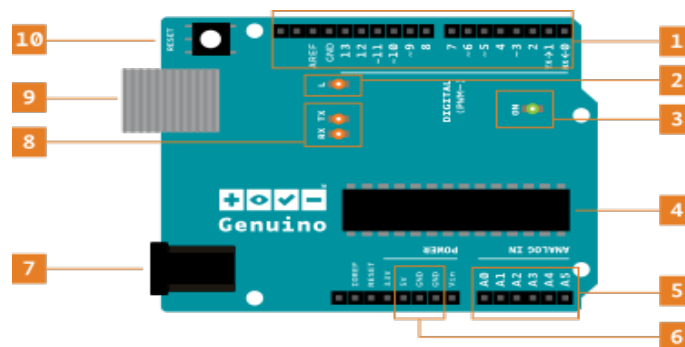
2 Tecnologias utilizadas

2.1 Arduino

O arduino é um dispositivo cujo objetivo é ser barato, funcional e de fácil programação, facilitando a vida de estudantes e projetistas amadores. Seu conceito de *hardware* livre favorece à criatividade dos "*makers*", permitindo a combinação entre diversos componentes para viabilizar a construção de projetos de diferentes níveis de utilidade. (ARDUINO.CC, 2018).

Ainda é permitido escolher entre diversos modelos de placa, desde o Arduino Nano até o Mega. A placa usada para os conhecimentos iniciais do projeto foi a Arduino Duemilanove. Segue abaixo alguns detalhes específicos sobre ela.

Figura 2.1 – Anatomia do Arduino



1. **Pinos digitais:** Usa-se esses pinos com *digitalRead()*, *digitalWrite()* e *analogWrite()*. *analogWrite()* funciona apenas nos pinos com o símbolo PWM.
2. **Pino 13 LED:** O único atuador embutido na placa. Além de ser um alvo útil para o primeiro exemplo de "pisca-pisca", este LED é muito útil para a depuração.
3. **LED de energia:** Indica que a placa está recebendo energia. Também é útil para depuração.
4. **Microcontrolador ATmega:** O coração da placa.
5. **Entradas analógicas:** Usa-se estes pinos com *analogRead()*.
6. **Pinos de 5V e GND:** Esses pinos são utilizados para fornecer energia de +5V e aterramento aos seus circuitos respectivamente.
7. **Conector de energia:** É assim que se liga o Genuino quando não está conectado a uma porta USB para energia. Pode aceitar tensões entre 7-12V.

8. **LEDs TX e RX:** Estes LEDs indicam a comunicação entre o Genuino e o computador. Espera-se que eles pisquem rapidamente durante o *upload* do *sketch*, bem como durante a comunicação serial. Útil para depuração.
9. **Porta USB:** Usada para alimentar o Genuino Uno, carregando seus esboços na placa, e para se comunicar-se com ele (via Serial. *Println()* etc.).
10. **Botão redefinir:** Redefine o microcontrolador ATmega.

2.2 Módulo I2C PCA9685 16-Channel 12-bit PWM

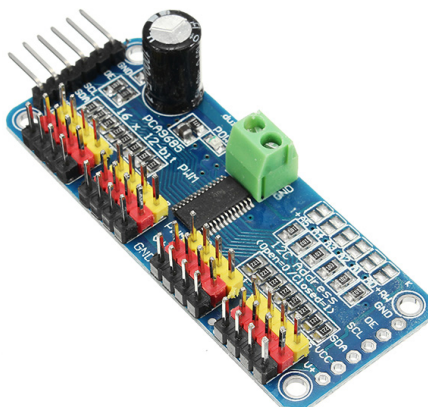
O Módulo de comunicação I2C possui 16 canais, e cada canal controla 1 servo motor, sendo possível assim controlar 16 servos motores PWM de 12 bit com resolução que vai de 0 a 4096. Pode interligar até 62 módulos e controlar até 992 saídas PWM utilizando apenas 2 pinos do microcontrolador (SCL e SDA). O módulo é compatível com microcontroladores 3.3 e 5V.

Cada canal pode gerar uma saída de duas formas distintas: a primeira utiliza o canal com sinal em estado alto ou baixo (ligado ou desligado) e a segunda utiliza o canal controlado com valores PWM, ou seja, cada um tem tensão de 5v e corrente de 25mA.

O PCA9685 possui alguns recursos exclusivos que o tornam mais adequado para aplicativos como a luz de fundo do LCD e Ambilight (SEMICONDUCTORS, 2009):

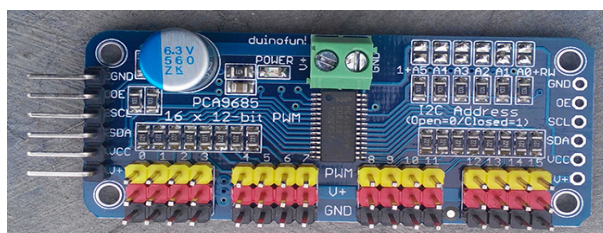
- O PCA9685 permite que os tempos de ativação e desativação dos servos escalonados funcionem em tempos diferentes. O atraso de tempo ligado e desligado é programável de forma independente para cada um dos 16 canais.
- O PCA9685 possui 4096 etapas (PWM de 12 bits) de controle de PWM individual.
- O PCA9685 tem um prescaler programável para ajustar as larguras de pulso PWM de vários dispositivos.
- O PCA9685 possui um pino de entrada de clock externo que aceita o clock fornecido pelo usuário (50 MHz máx.) No lugar do oscilador interno de 25 MHz. Esse recurso permite sincronização de vários dispositivos.
- O PCA9685 possui um oscilador interno para o controle PWM. A frequência usada para o controle PWM é ajustável cerca de 40 Hz a 1000 Hz. Isso permite o uso do PCA9685 com controladores de fonte de alimentação externos. Todos bits são definidos na mesma frequência.
- O estado padrão de Power-On Reset (POR) dos pinos de saída LEDn é LOW no caso de PCA9685.

Figura 2.2 – Módulo I2C PCA9685



Para este projeto o módulo I2C PCA9685 é usado para controlar os servomotores do braço robótico. Ele possui pinos de entrada e saída como vemos na Figura 2.3. Destes pinos apenas os seguintes serão usados:

Figura 2.3 – Módulo I2C PCA9685 visto de cima



- 16 canais com 3 pinos cada (GND - preto, VCC - vermelho e PWM - amarelo)
- SCL - entrada de relógio serial. Este pino é a entrada do relógio para a interface serial I2C e é usado para sincronizar dados movimento na interface serial;
- SDA - Entrada/Saída de Dados Seriais. Este pino é a entrada / saída de dados para a interface serial I2C;
- VCC - pino de alimentação CC para fonte de alimentação primária;
- GND - Ground.

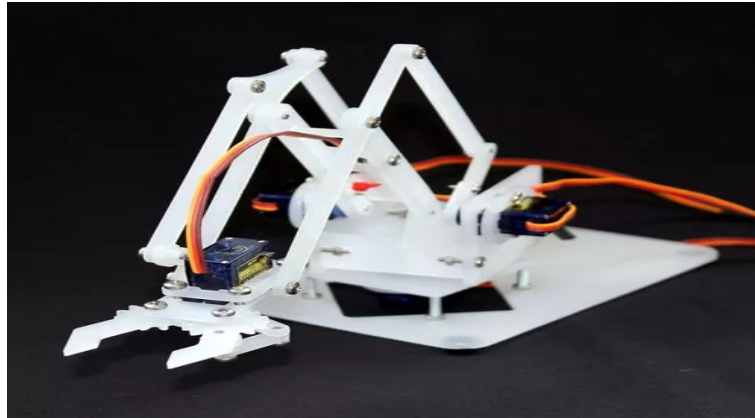
2.3 Braço Mecânico de Acrílico

Este kit é composto por 1 braço robótico em acrílico com garra, 4 servos motor 9g e parafusos para montagem.

O braço robótico tem como objetivo auxiliar na aplicação em projetos que trabalham com automação de robôs, proporcionando maiores possibilidades de movimentos e desenvolvimento

de tarefas. Através de plataformas de prototipagem (Arduíno, Raspberry Pi, etc), ele é capaz de cumprir variadas funções. Os 4 servos motores são utilizados em conjunto para que o braço realize suas articulações, movimentos de até 180°, de acordo com os comando indicados, podendo o usuário controlar o seu giro e a posição, além de possuir uma garra com uma abertura de 55mm, como mostra a figura abaixo:

Figura 2.4 – Braço robótico em acrílico



3 Procedimentos

3.1 Instalando bibliotecas necessárias

Para o correto funcionamento do projeto se faz necessário que as seguintes bibliotecas estejam devidamente instaladas.

3.1.1 meArm Adafruit

Para facilitar o uso do braço mecânico com o módulo I2C PCA9685, foi utilizado a biblioteca meArm_Adafruit desenvolvido por RorschachUK. A mesma pode ser baixada no link: https://github.com/RorschachUK/meArm_Adafruit.

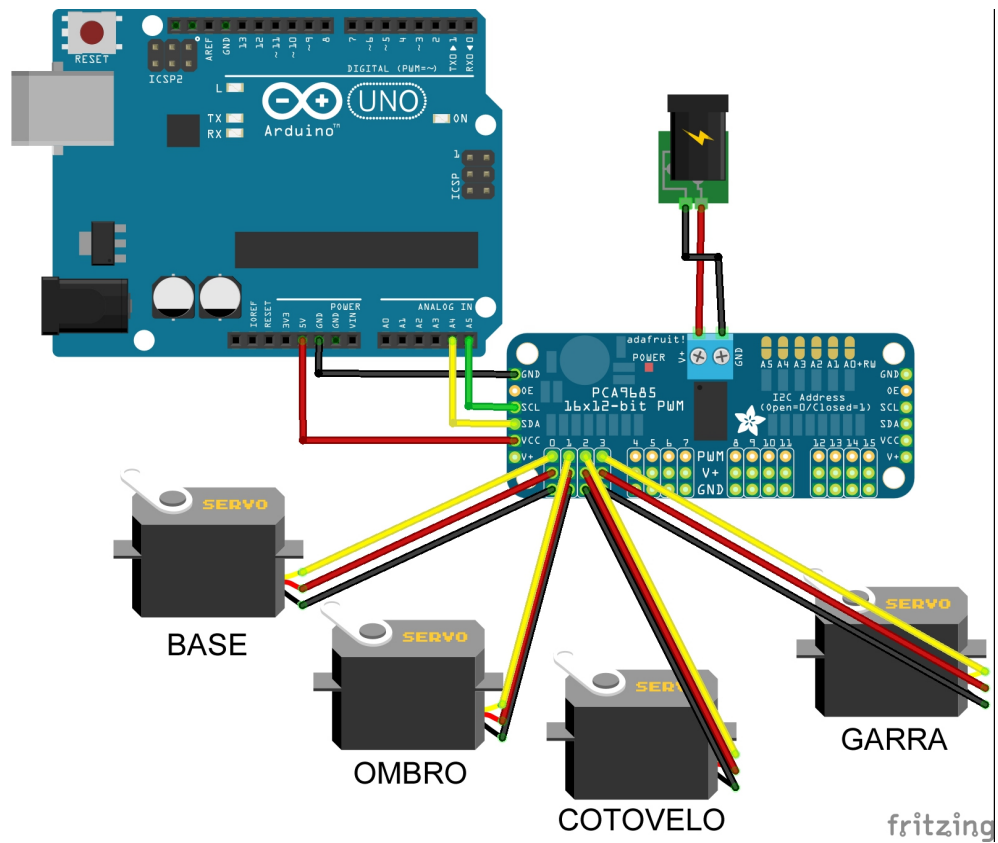
Faça o download em formato .Zip, descompacte o arquivo e copie-o para a pasta *libraries* que fica na *home* do seu usuário.

3.1.2 Adafruit PWM servo Driver Library

Visto o uso da biblioteca citada anteriormente, é necessário a utilização da biblioteca Adafruit PWM servo Driver Library para controlar o Módulo I2C PCA9685. A mesma pode ser baixada pela IDE do Software Arduíno.

3.2 Montagem

Figura 3.1 – Montagem do projeto



3.3 Código Fonte

Devido o código ser extenso e ocupar espaço demasiado no relatório, foi optado por colocar o código fonte em um repositório do GitHub. Segue o link para acesso: https://github.com/jrrf/projeto_2_meArm.

4 Resultados Obtidos

Como resultados foi obtido o controle da garra por coordenadas passadas via Serial da IDE do Arduíno. Leitura por meio da mesma Serial dos valores da coordenada após serem filtrados pela biblioteca "meArm_Adfruit".

4.1 Funcionamento

O projeto funciona da seguinte maneira. Ao conectar uma fonte de energia para alimentar a placa Arduíno o processador começa a rodar o código e aciona(nos momentos certos) os componentes eletrônicos que estão conectados a placa. Também é necessário uma fonte externa de 5V para alimentar o módulo I2C PCA9685(placa utilizada para controlar os servos motor). Quando ambas as fontes estão conectadas o braço mecânico de acrílico se move.

Primeiramente é setada uma posição inicial de repouso para o braço, daí em frente o mesmo fica aguardando o envio de comandos externos via Serial da IDE do Arduíno para poder se mover novamente. Quando o usuário seta uma coordenada para o braço a biblioteca meArm_Adfruit verifica se estas coordenadas são válidas para a área de movimento do braço e logo em seguida adapta esses valores passados pelo usuário para valores que não firam os limites físicos do braço.

Enquanto o circuito estiver ligado é possível ficar monitorando os valores das coordenadas que a biblioteca passou para o braço.

Figura 4.1 – Funcionamento 1

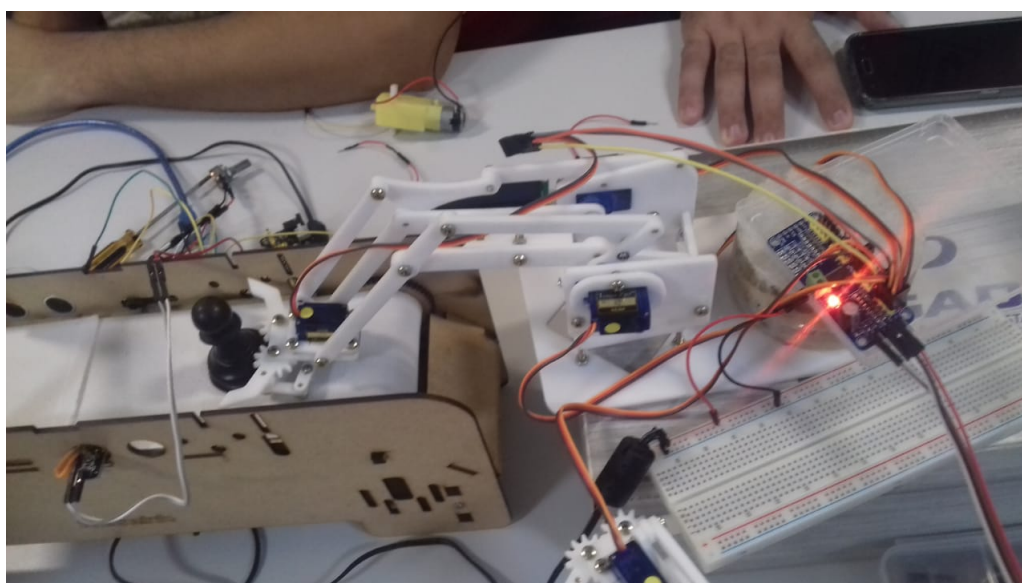


Figura 4.2 – Funcionamento 2

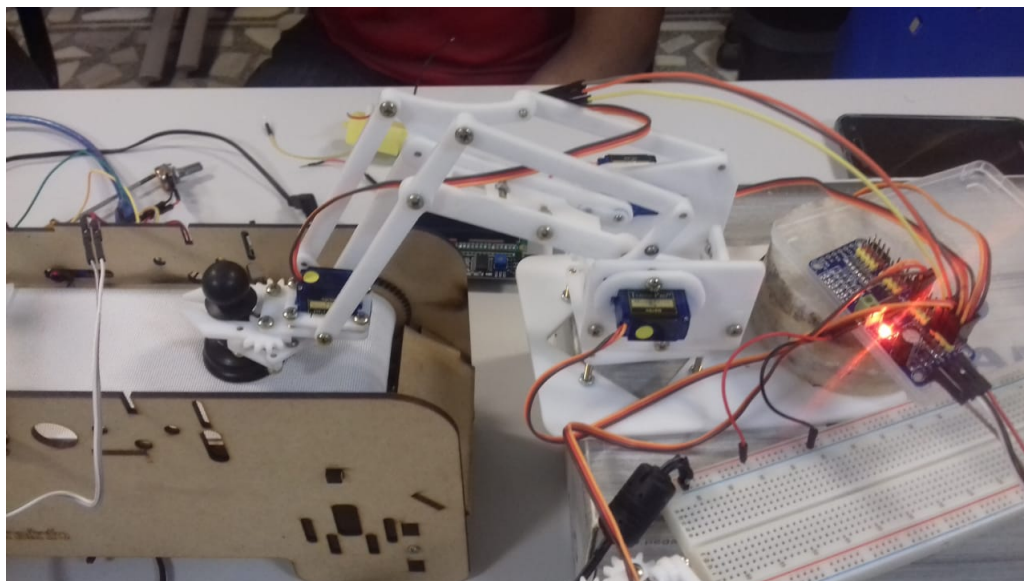
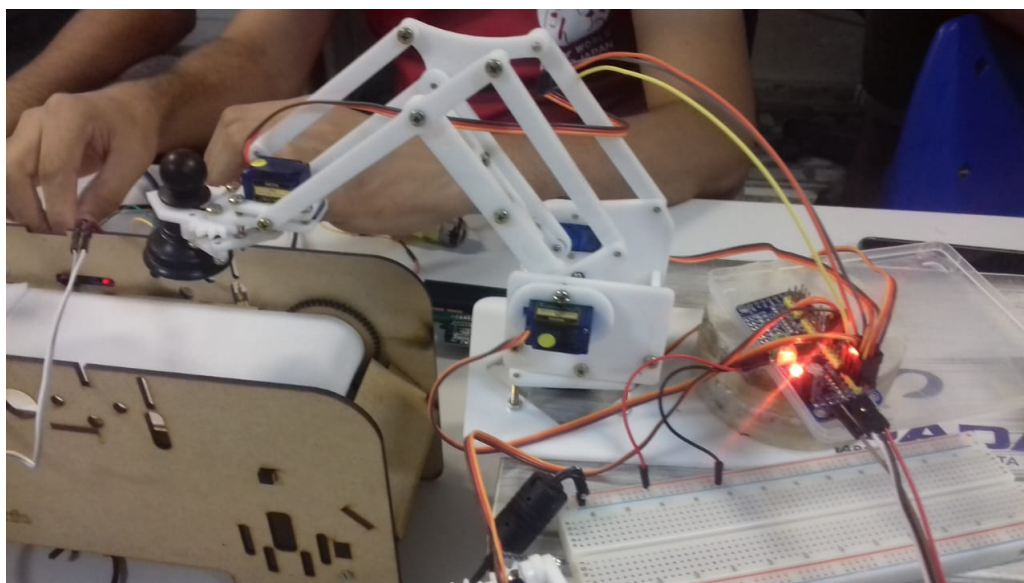


Figura 4.3 – Funcionamento 3



4.2 Dificuldades

Inicialmente a dificuldade encontrada foi a montagem e calibração do braço mecânico, principalmente pelo fato de existir poucos manuais para auxiliar, sendo que os que existem tem didática dificultosa. A partir disso, os obstáculos foram na compreensão do uso das funções da biblioteca `meArm_Adafruit`.

Não foram grandes dificuldades, mas as mesmas fizeram requerer tempo em demasia, em buscas, estudos, e testes, que fizeram enriquecer o projeto.

Dificuldade em receber interrupções, provavelmente o circuito estava com algum ruído e o pino de interrupção ficava identificando interrupções todo instante... A solução foi colocar o arduino para se comunicar através da serial (TX e RX) para receber as tarefas que precisavam

ser realizadas soluções ainda melhores seriam fazer a comunicação sem fio, via bluetooth, wifi, etc... Pois não teria limitação do tamanho do fio.

Referências

ARDUINO.CC. *Arduino Uno Rev3*. 2018. Disponível em: <<https://store.arduino.cc/usa/arduino-uno-rev3>>. Acesso em: 30-05-2018.

SEMICONDUCTORS, N. *PCA9685*. 2009. Disponível em: <<https://html.alldatasheet.com/html-pdf/293576/NXP/PCA9685/54/1/PCA9685.html>>. Acesso em: 26-06-2018.