# C:\Users\DELL\Desktop\logo uenf.jpg

## **RELATÓRIO DE PROJETO**

MICROCONTROLADORES

PROF. ÁTILA CARVALHO

Braço robótico com 6 graus de liberdade controlado por Bluetooth

Daniel Brito dos Santos;

João Vítor Fernandes Dias;

José Lucio Castro de Azevedo.

## Introdução

Este documento foi desenvolvido para a disciplina optativa eletiva de microcontroladores. Ele apresenta um relatório sobre o desenvolvimento de um braço mecânico utilizando tecnologias de envio de dados a distância. Para tal, precisaremos nos basear em alguns conceitos teóricos:

O primeiro deles será sobre o envio e recebimento de dados. As formas mais usuais de troca de dados utilizada pelo Arduino UNO são através dos protocolos de transmissão de dados USB e UART, embora ele também disponibilize portas que possam utilizar os protocolos SPI e I²C. Os que serão utilizados nesse projeto são os mais usuais. O [USB](https://www.iperiusbackup.net/pt-br/conhecendo-mais-sobre-o-novo-modelo-usb-c-e-vantagens-na-sua-utilizacao/) para o upload do código do computador para o Arduino e UART para receber os dados dos módulos bluetooth e wi-fi.

Já o bluetooth e o wi-fi são tecnologias para transferência de dados através de radiofrequência, portanto, sem a utilização de fios. A diferença entre eles se encontra em seu objetivo: enquanto o bluetooth foca na transferência de dados em uma curta distância, o wi-fi foca em estabelecer conexões de rede em longas distâncias.

Além dos conceitos de transferência de dados, é necessário também entender sobre os servomotores. Servomotor é um atuador eletromecânico que realiza o movimento de seu eixo em determinada posição angular a partir da sua entrada. O ângulo do eixo é determinado através da duração da largura do pulso enviado à entrada do servomotor. Este sinal varia de 0 a 5 volts. Dessa forma, o circuito de controle do servomotor monitora esses sinais com intervalos de 20 milissegundos e se dentro desse intervalo houver alguma alteração do sinal, ele altera a posição do seu eixo conforme o sinal recebido.

## Parte Experimental

### Materiais utilizados:

1. Arduino UNO
2. Kit de servomotor (7)
3. Cabo USB
4. Jumpers
5. Sensor Shield V5.0
6. Módulo Bluetooth HC-05
7. Smartphone com bluetooth
8. Carregador de celular adaptado
9. Protoboard
10. Kit de peças de montagem em acrílico
11. Conjunto de porcas e parafusos
12. Chave de fenda e Philips

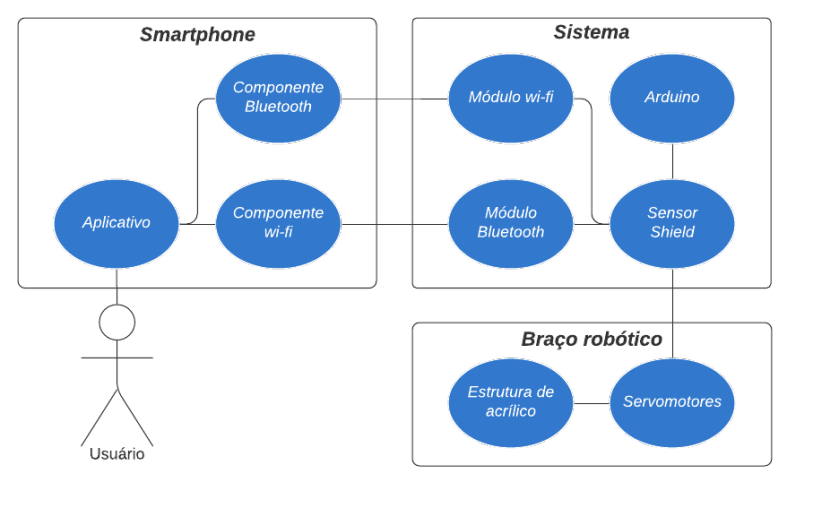
### Etapas dos experimentos:

1. Montagem da estrutura em acrílico utilizando porcas, parafusos e as chaves de fenda e Philips e também os servomotores
2. Associação do circuito utilizando o Arduino, Sensor Shield e Jumpers, conectando aos servomotores
3. Programação do Arduino para movimentar os servos
4. Programação do Arduino para receber dados via bluetooth
5. Programação do App Android para envio de dados via bluetooth
6. Correção de bugs e melhorias gerais

### Estrutura de execução do projeto:

Como pode ser observado na Figura 1, o usuário irá utilizar de um aplicativo em seu Smartphone para enviar as angulações dos servomotores através de seus componentes Bluetooth e wi-fi para os módulos bluetooth e wi-fi conectados ao Sensor Shield que está acoplado ao Arduino. Através do código presente no Arduino, o Sensor Shield irá alterar a angulação dos servomotores que estão anexados à estrutura de acrílico do braço robótico.

Figura 1: Diagrama de uso



## Resultados e Discussão

### Resultados

Como resultado vê-se de forma sucinta a montagem do braço robótico e do aprendizado sobre tecnologias de comunicação remota.

##### Braço montado

A montagem da estrutura e componentes do braço robótico é uma tarefa manual, que exige atenção aos detalhes, e o uso de ferramentas. Foram utilizados uma chave Philips e o manual disponibilizado pelo fornecedor das peças do braço robótico, para montar o novo braço robótico com 5 graus de liberdade, sendo este braço feito em acrílico.

Após o desenvolvimento dos códigos no Arduino IDE, e o aprendizado sobre programação de aplicativos seguido de seu desenvolvimento, ambos os códigos foram integrados e permitiram a transmissão de dados entre um dispositivo Android e um Arduino através de conexão Bluetooth. Posteriormente com a montagem do braço robótico, a transmissão de dados, já funcional, foi utilizada para movimentar o braço robótico.

##### Código final do Arduino

Com o objetivo final de controlar o braço robótico através de tecnologias de comunicação remota, foram desenvolvidos programas no Arduino IDE para se alcançar este resultado.

###### Bluetooth

O recebimento de dados via *Bluetooth* é bem simples. Como exemplificado pelo código a seguir (Código 1).

Na função *setup ()* ele apenas inicia a comunicação serial na velocidade 9600 bauds, ou seja, 9600 bits por segundo. Na função *loop ()* se o valor estiver disponível na entrada serial, ele irá fazer a leitura e imprimir o valor lido.

Código 1: Recepção Bluetooth simplificada



###### Wi-Fi.

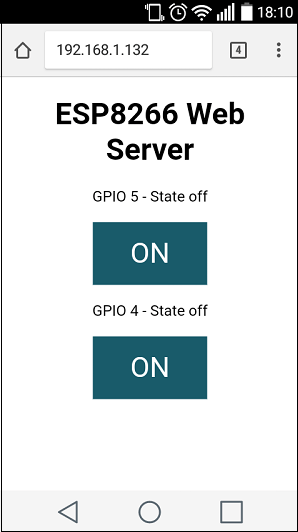
O recebimento de dados via wi-fi apresenta uma complexidade muito superior à do envio de dados via Bluetooth. Foram realizados testes com o módulo NodeMCU Amica que possui wi-fi integrado. Nesses testes pode-se observar demora superior quando comparado ao Arduino, na configuração inicial do microcontrolador, assim como para fazer o upload dos códigos para o microcontrolador. O código utilizado no teste é uma junção de várias linguagens de programação, o que pode gerar certa confusão ao se misturar comandos em C com as tags do HTML. Mas apesar dessas dificuldades foi possível realizar a transmissão dos dados.

Inicialmente foi utilizado como base um servidor que disponibilizava dois botões (como pode ser visto na Figura 2), e que, quando apertados, executavam duas tarefas diferentes. Posteriormente modificou-se o código para que ao invés de transmitir os dados a partir dos botões, fosse transmitido após a mudança de posição de um *slider* (Controle deslizante). Entretanto, a informação só é enviada após soltar o *slider* na nova posição, o que reduz a quantidade de dados que é transmitida.

Mesmo com esses pontos negativos, é esperado um estudo mais aprofundado em relação à aplicação wi-fi e que as características ruins sejam amenizadas.

O código em si é muito extenso, e, portanto, não será apresentado neste documento.

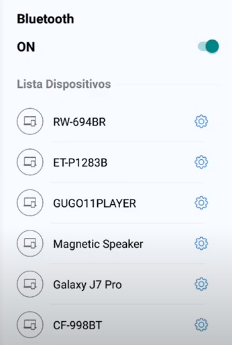
*Figura 2: ESP8266 Web Server*



##### App feito

Após desenvolver um app com conexão Bluetooth, foi possível desenvolver um aplicativo próprio que conseguisse se conectar ao módulo Bluetooth HC-05 conectado ao Arduino, fato confirmado pela mudança no padrão luminoso emitido pelo módulo. (Figura 3)**.**

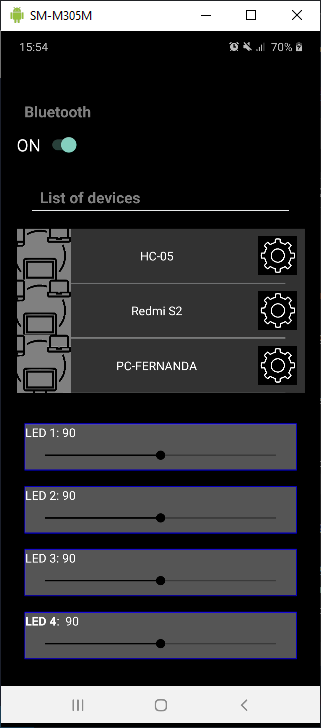
Figura 3: Aplicativo desenvolvido no WorkShop



Após pesquisas na documentação de bibliotecas de comunicação Bluetooth pôde-se entender e utilizar as funções necessárias para o envio de dados via Bluetooth.

Baseado no que foi aprendido com o app de treinamento, e na pesquisa das documentações, foi possível alterar algumas funções já existentes para poder enviar os dados.

*Figura 4: Um dos primeiros layouts do app de envio de dados por slider*



O app mostrado na Figura 4 permite ligar e desligar o Bluetooth no dispositivo, se conectar a um dispositivo disponível e enviar para ele os valores dos *sliders* quando eles eram movimentados. Com isso foi possível se conectar ao módulo HC-05 e enviar os dados apropriadamente.

### Discussões

Visto o montante de novos conhecimentos obtidos durante o desenvolvimento do projeto, pode-se olhar em retrospecto e analisar de uma forma mais crítica os resultados obtidos, algumas dessas críticas estão listadas logo abaixo.

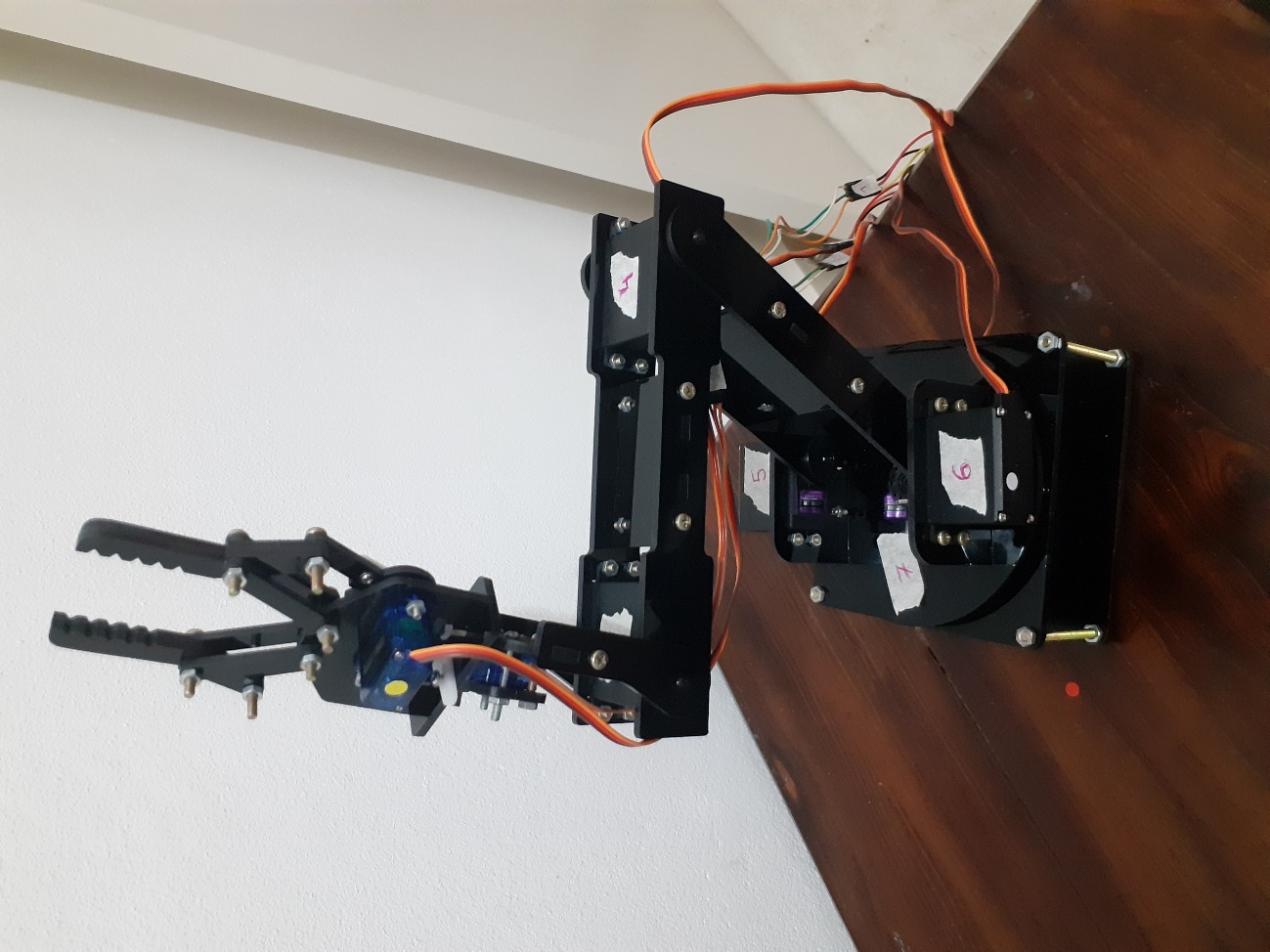
##### Braço robótico montado

A montagem do braço robótico foi bastante trabalhosa e durou em torno de 10 horas. De um modo geral não foi muito complicado, porém a falta de instruções claras presentes no manual reduziu o rendimento. Dessa forma, foi inevitável que acontecessem erros de montagem. Outros problemas foram a existência de peças que não encaixavam adequadamente e necessitaram de um pouco de força para se fixarem devidamente, e também alguns parafusos que eram mais longos do que necessário e outros que eram mais curtos do que o necessário.

Optou-se por um modelo de braço robótico com 5 graus de liberdade que tem 5 servomotores “*Servo Motor MG996R Tower Pro*” que são mais potentes e trazem maior resistência ao peso da estrutura do que os servomotores “*Micro Servo 9g SG90 TowerPro*”.

A Figura 5 mostra o braço robótico devidamente montado.

*Figura 5: Braço robótico montado*



###### Movimentação do braço robótico com controle via Bluetooth

Após o desenvolvimento do app que enviava os valores dos *sliders* apropriadamente para o HC-05 e da conclusão da montagem do braço robótico, os diferentes segmentos do projeto foram conectados, resultando na movimentação esperada do braço robótico, necessitando apenas de alguns ajustes em relação à angulação máxima de cada servo motor. Após esses ajustes, o braço robótico deveria estar completamente funcional.

Após algumas tentativas percebeu-se que após determinada angulação, os servos não eram capazes de suportar o peso da estrutura e acabavam deixando a estrutura descer lentamente.

##### Melhorias do app

Como o aplicativo desenvolvido foi programado do zero, diversas características que pareciam adequadas passaram a se mostrar incômodas ou aprimoráveis a medida em que se desenvolvia o app. Para tal, seria necessária uma completa reformulação de sua estrutura. Abaixo estão listadas algumas mudanças possíveis:

1. Não mostrar na lista de dispositivos aqueles que estiverem desligados;
   1. Após se conectar, não mostrar o componente de conexão até que perca a conexão.
   2. Quando um dispositivo estiver conectado mostrar um único botão com os dados do dispositivo conectado que possa ser tocado para desconectar e voltar a mostrar a lista de componentes disponíveis;
2. Atualizar a lista de dispositivos disponíveis sempre que algum deles mudar de estado;
3. Permitir a conexão via wi-fi no mesmo App;
4. Fazer com que os Sliders desapareçam enquanto estiver desconectado;
5. Tornar os componentes do app modulares;
6. Melhorar o visual de todo o aplicativo tornando mais intuitivo e atraente.

##### Melhorias do código do Arduino

Assim como analisado acima, o código presente no Arduino também pode ser aprimorado. Quanto ao recebimento de dados via bluetooth, não aparenta ter muita necessidade de alteração, entretanto, o recebimento via wi-fi e o código para controle dos servos apresentam várias melhorias possíveis, tais quais:

1. Posição inicial dos Servomotores (Configurar para manterem na mesma posição ao ligar);
2. Impedir que o braço robótico encoste no chão (Limitar as angulações dos diversos servomotores);
3. Tornar Servos mais suaves durante a movimentação (Pesquisar sobre o uso da Biblioteca ServoEasing e sobre o controlador PID);
4. Garantir a [proteção dos servos ante esforço excessivo](youtu.be/LKJLCJvyVdk)
5. Pesquisar sobre o [desenvolvimento de interface gráfica](youtu.be/2cjufbgOBYo) para [microcontrolador](remotexy.com)
6. Mudança em relação a movimentação do braço:
7. Se mover nas angulações yaw (guinada), pitch (arfagem) and roll (rolamento), mantendo a extremidade do braço centralizada;
8. Estudo sobre Cinemática Direta (Input: ângulos das articulações; output: coordenadas x, y, z) e Cinemática inversa (Input: coordenadas x, y, z; Output: ângulos das articulações)

##### Melhorias de Hardware

Outros fatores de melhoria relevantes são o aumento da resistência por parte do braço robótico para que consiga executar os movimentos com força, velocidade e precisão, além de corrigir o bug encontrado na comunicação bluetooth.

1. Analisar o consumo de energia para averiguar se é isto que causa a insuficiência de força no braço robótico;
2. Impedir que o braço robótico tombe (possivelmente utilizando pesos em sua base, aumentando a área da base, ou fixando sua base);
3. Utilizar o módulo Bluetooth e o módulo wi-fi simultaneamente (melhoria operacional);
4. Remover excessos dos parafusos e buscar parafusos mais apropriados (estética);
5. Pesquisar sobre o uso de capacitores para evitar o “*jittering*” (efeito que causa tremor durante a operação dos servomotores);
6. Pesquisar sobre o uso do Módulo I2C p/ Servo Motor - PCA9685 (alternativa para o módulo Sensor Shield, que permite o uso da comunicação serial I2C);
7. Pesquisar sobre as tecnologias 6loWPAN e BLE
8. Adquirir ESP32

Quanto à falta de força do braço robótico, algumas possibilidades foram ventiladas. Dentre elas, se encontram a falha elétrica e a falha mecânica. Caso seja uma falha elétrica, será necessário analisar se o problema é insuficiência de corrente ou tensão, se for corrente, pode-se utilizar indutores para estabilizá-la. Se for tensão, pode-se utilizar capacitores. Caso seja uma falha mecânica, será necessário analisar se o próprio servomotor não é forte o suficiente para suportar o peso, sendo esta a situação, pode-se utilizar de suportes elásticos, sendo eles molas ou de fato elásticos.

## Conclusão

Com esse projeto pôde-se aprender e ver na prática conceitos presentes na área de automação industrial através do uso de servomotores e o conceito de graus de liberdade.

Além disso, também foi estudado sobre o transporte de dados através dos [protocolos de comunicação serial](https://www.robocore.net/tutoriais/comparacao-entre-protocolos-de-comunicacao-serial.html) e as tecnologias de transmissão remota de dados bluetooth e wi-fi.

Também foi desenvolvido um aplicativo para controle do braço robótico via Bluetooth que permitiu controlar o braço robótico com precisão, e também um site que permitiu enviar informações via wi-fi para o NodeMCU.