

PROGRAMA INSTITUCIONAL DE BOLSAS DE INICIAÇÃO CIENTÍFICA E TECNOLÓGICA
UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE DARCY RIBEIRO
CENTRO CCT
LABORATÓRIO LCMAT

Relatório do período: XX/2020 a XX/2021

RELATÓRIO ANUAL

Nome do Bolsista: João Vítor Fernandes Dias

Curso: Ciência da Computação

Nº Matrícula: 00119110377

Orientador: Fermín Alfredo Tang Montané

Título do Projeto: Desenvolvimento e Controle de Dispositivos para Internet das Coisas

Título do Plano de Trabalho: Estudo sobre a Integração de Plataformas Microcontroladas para Internet das Coisas

Fonte financiadora da Bolsa: PIBIC / CNPq

1 Etapas propostas no plano de trabalho

O plano de trabalho contempla as seguintes propostas:

- a) Melhorias no aplicativo Android levando em consideração o design, a funcionalidade e a interação com o usuário
- b) Melhorias na estrutura do braço robótico e no circuito associado a ele, levando em consideração a estabilidade do braço, angulação máxima dos servomotores, alimentação de energia sendo possível pesquisar sobre o uso de capacitores para evitar o “jittering” (efeito que causa tremor durante a operação dos servomotores); conectividade remota, sendo possível a integração de ambas tecnologias Bluetooth e Wi-Fi no protótipo. Pesquisar sobre o uso do Módulo PCA9685 I2C, que permite o uso da comunicação serial I2C, para servomotores como alternativa ao módulo Sensor Shield. Documentação do estudo.
- c) Melhorias no código Arduino que realiza a movimentação do braço, visando uma movimentação mais suave e segura. Documentação.
- d) Realização de testes de avaliação e desempenho da interface de controle e do braço robótico.
- e) Elaboração de relatório técnico.

Este plano de trabalho foi muito interessante, pois foi muito denso. Embora possua menos tópicos do que o plano de trabalho anterior, ele apresenta tarefas muito mais completas relacionadas a diversas áreas diferentes.

O *item a* considera técnicas de desenvolvimento de software como a modularização de componentes e, portanto, a necessidade da abstração, levando em conta UX (User eXperience – Experiência do Usuário) e UI (User Interface – Interface do Usuário) (Teixeira, 2014), bem como o aumento da quantidade de suas capacidades técnicas, o que busca um refinamento maior durante o desenvolvimento, havendo então um grande aprimoramento em relação aos anteriores.

Em relação ao *item b*, vemos diversos tópicos relacionados a hardware, muitos deles envolvendo eletrônica. Temos a questão física das forças atuantes no braço, mas principalmente a questão de engenharia elétrica que envolve o cálculo das correntes e tensões necessárias para que todo o circuito eletrônico do braço robótico possa funcionar utilizando de todo seu potencial, um exemplo claro da não aplicação desse cálculo é justamente esse efeito de tremulação chamado de “jittering”. Além disso, inclui também a inclusão da integração da tecnologia Bluetooth e Wi-Fi no protótipo, e a pesquisa sobre o uso de novas alternativas de gerenciamento do servo através de novos dispositivos eletrônicos.

Já no *item c*, visa a parte do desenvolvimento de software para lidar com a tremulação, além de evitar que o uso da potência total do novo braço seja danoso para seu próprio equilíbrio e bom funcionamento.

No *item d*, se refere aos testes necessários para conferir se o transporte de dados à distância utilizando das tecnologias selecionadas está ocorrendo de maneira devida, e também para averiguar quão bem o braço robótico performa com as informações recebidas.

Por fim, no *item e*, temos a elaboração deste relatório técnico, com o qual visa-se uma abordagem de fato mais técnica em relação aos anteriores, que eram mais teóricos, tendo em vista a mudança para a bolsa de Iniciação Tecnológica.

2 Introdução

Com as revoluções industriais, foram surgindo diversos novos mecanismos para aumentar a produção, tanto mecanismos “sociais”, como metodologias de gerência dos operários, quanto mecanismos de fato mecânicos como esteiras e máquinas de grande porte (Livro sobre Revolução industrial, 2004). Em dado momento, até mesmo os funcionários que operavam as máquinas foram sendo substituídos por outros mecanismos como sistemas automatizados que fossem mais precisos, incansáveis e que não precisariam de horas extras para exercer suas funções.

Com esse contexto inicial, podemos analisar duas situações: a primeira onde o trabalho é mais preciso e específico e demandaria ações repetitivas e com correções de erros em sua execução. A segunda onde as diversas máquinas da indústria precisam ser gerenciadas, mensuradas e reguladas constantemente para evitar falhas catastróficas.

Neste primeiro caso, inicialmente usava-se (e ainda se usa) esteiras onde em suas laterais estão dispostos diversos funcionários que repetidamente executam a mesma tarefa. Manter seres humanos nessas condições, além de ser dispendioso para as empresas, é também um possível ponto de falha. Como pode ser comprovado pelo teste palográfico (Godoy & Noronha, 2005), mesmo ações simples, mas repetitivas, acabam estando suscetíveis a erros ou anomalias durante sua execução. Havendo a substituição dessas pessoas por braços robóticos, esse ponto de falha, embora ainda existente, se torna menos provável e menos dispendioso a longo prazo, embora seu investimento inicial seja maior.

No segundo caso, para haver o constante monitoramento, é necessária a instalação e manutenção de sensores que estariam fisicamente conectados a diversos cabos conectados a uma central de gerenciamento com diversos painéis. Como forma de aprimorar essa dinâmica, esses sensores poderiam estar conectados à internet em uma rede segura, e então, sem a necessidade de tantos cabos, envia e recebe informações à distância, podendo ser controlado em qualquer lugar do mundo por qualquer dispositivo com acesso à internet.

Pode então ser feita uma interação entre ambos os casos, onde dispositivos com acesso à internet, como smartphones, possam fiscalizar, averiguar a execução, ou até mesmo controlar os braços robóticos que podem estar trabalhando dia e noite dentro de uma indústria.

3 Objetivos

O objetivo principal é o de conseguir integrar a tecnologia Wi-Fi, vista como viável no plano de trabalho anterior, ao braço robótico de forma funcional; além disso, vê-se como positivo que seja feito uma melhoria na modularização dos componentes do código do app, permitindo que suas alterações futuras se tornem mais fáceis.

4 Metodologia

Diversas metodologias foram utilizadas visando alcançar o objetivo. Dentre elas podemos listar algumas mais diretas que já foram utilizadas

- a) A reconfiguração do ambiente de desenvolvimento;
- b) Aumento do uso de conceitos de pensamento computacional como abstração e decomposição (nesse contexto chamado de modularização) (Azevedo & Maltempi, 2020);
- c) Revisão dos códigos anteriores;
- d) Melhor organização dos códigos recentes.

Abaixo serão explicadas outras metodologias de forma mais aprofundada:

4.1 Estudo de documentação oficial do dispositivo utilizado

Durante o desenvolvimento viu-se necessário o estudo mais aprofundado do objeto de trabalho, nesse caso, o microcontrolador NodeMCU Amica. Para isso, houve extensa leitura, principalmente de sua [documentação oficial](#) (Grokhotkov, 2017), quanto a sua completa capacidade, parte da qual não se viu necessária de aplicar na pesquisa, mas que contribuiu para elucidar quanto à algumas questões que ficaram pouco explicadas no projeto anterior, principalmente a aparente dificuldade da conexão via Wi-Fi em relação ao Bluetooth sobre a qual será descrita mais a diante neste relatório na seção de resultados.

4.2 Testes com novos ambientes de desenvolvimento

Foram testadas duas IDEs diferentes para o desenvolvimento dos códigos para o Arduino: [Arduino IDE 2.0](#) e [VS Code](#). Ambos pareceram muito similares entre si, mas mais complicados e menos intuitivos do que o Arduino IDE. Mesmo com essas desvantagens, seria de bom proveito ter todo o ambiente de desenvolvimento ao redor do VS Code.

4.3 Programação nos ambientes escolhidos utilizando o TDD

Foi mantido o uso do React Native como framework para desenvolvimento do aplicativo, porém, agora utilizando uma nova biblioteca, necessária para a comunicação via Wi-Fi utilizando o protocolo UDP. Este desenvolvimento foi mais uma vez executado seguindo o método TDD (Test Driven Development, Desenvolvimento Guiado por Testes em inglês) pois este método tem se mostrado bem eficaz durante o desenvolvimento.

Neste método, utiliza-se de constantes testes do que foi desenvolvido para checar se o software está funcionando devidamente, e, caso não esteja, ser corrigido o quanto antes. De uma forma resumida, baseia-se em escrever uma lógica inicial que faça sentido para o objetivo, mesmo que esteja incorreta. Após ter a lógica concluída, modifica-se o código para que ele funcione, assim que funcionar, modifica-o novamente para aprimorar a sua escrita e polir a sua estrutura.

4.4 Utilização de ferramentas para teste de transmissão de dados

Para saber se os dados estão sendo enviados pelo Smartphone e recebidos pelo NodeMCU, se viu necessário o uso de uma ferramenta externa que fosse capaz de enviar e

receber esses dados. A ferramenta utilizada foi o “UDP - Sender/Receiver” ilustrado pela Figura 1, disponível na [Microsoft store](#).

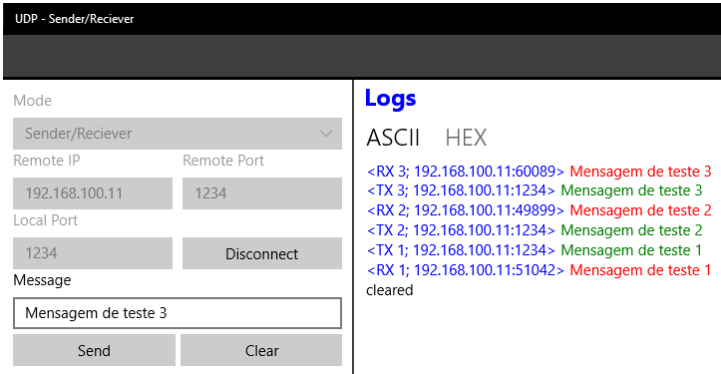


Figura 1: Wi-Fi - Software para envio e leitura de pacotes - Fonte: o autor

Com ele foi possível ter uma ferramenta na qual se basear em relação à expectativa de como o dado é enviado e recebido, de que forma ele deveria se portar e como deveria se conectar à rede. Porém ainda faltava a confirmação de que o pacote de fato está sendo enviado, e para isso foi utilizado o [Wireshark](#) que é um analisador de protocolos de rede (Equipe do Wireshark, 2022). Com este software, pode-se ver precisamente quais são os dados que estão trafegando na rede em que se está conectado.

Como pode ser visto na Figura 2, vemos quais são os pacotes que estão passando pela rede. Vemos sua origem e seu destino, seu protocolo, tamanho e informação, como por exemplo a porta de origem, a porta de destino e seu tamanho interno real.

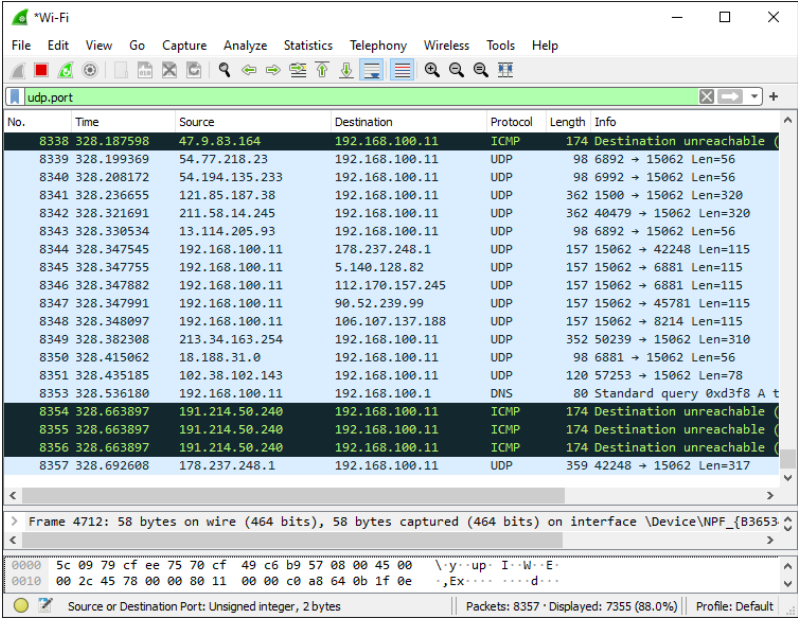


Figura 2: Wi-Fi - Software para análise de pacotes na rede Wi-Fi - Fonte: o autor

Com essas ferramentas preparadas, pôde-se testar o envio e recebimento de dados.

5 Resultados e Discussão

Ao decorrer do projeto, foram obtidos diversos resultados e avanços consideráveis que podem ser vistos como diretamente e indiretamente relacionados ao objetivo do projeto:

5.1 Resultados Indiretos

Os resultados indiretos, dizem respeito a avanços obtidos que auxiliaram de uma forma mais geral ao desenvolvimento do projeto e de sua continuação por bolsistas futuros que almejem trilhar os mesmos passos deste plano de trabalho.

5.1.1 Reconfiguração do ambiente

A reconfiguração do ambiente de desenvolvimento se mostrou necessário e complexo toda vez que surgisse necessidade de desenvolvimento do projeto em um computador diferente, por isso foi desenvolvido um documento que descreve o passo a passo necessário para a instalação do ambiente para futuros bolsistas.

5.1.2 Revisão de códigos anteriores e documentação

Com o passar do tempo e aquisição de experiência em desenvolvimento de software para o projeto, passou-se a ter maior entendimento sobre fluxos saudáveis de desenvolvimento e estruturas de organização que promovam uma forma mais intuitiva de se navegar nos diversos arquivos gerados, tanto para quem o estiver desenvolvendo, quanto para quem desejar analisar mais profundamente este projeto. Dessa forma, foi feita uma reorganização dos arquivos relacionados aos projetos anteriores.

Com isso, foi feita também a revisão dos diversos códigos desenvolvidos. Além de perceber diversos aprimoramentos que os códigos anteriores poderiam receber, principalmente em seu conceito de abstração e modularização, percebeu-se a falta que faz a inexistência de uma documentação bem estruturada, ou no mínimo alguns comentários dentro do código que expliquem a sua finalidade e lógica utilizada.

Assim, os arquivos anteriores foram reorganizados e aprimorados. Tanto os arquivos relacionados ao Arduino quanto ao app desenvolvido em React Native, passaram por este processo, visto que eles seriam a base para o desenvolvimento do projeto atual, então, havendo uma base firme, pode-se construir uma forte estrutura.

5.1.3 Melhorias da estrutura interna do aplicativo Android

Cumprindo a primeira etapa do projeto, foi revisitado os códigos anteriores do aplicativo desenvolvido, como foi listado no subtópico 8.3 “Software App” do tópico 8 “Perspectivas de continuidade do trabalho”, haviam diversos existentes e aprimoramentos esperados. Como o aplicativo para controle via Bluetooth já havia sido desenvolvido e funcionava, optou-se por desenvolver um novo aplicativo, tentando seguir o mesmo padrão visual que o anterior, para poder fazer a conexão via Wi-Fi com o microcontrolador.

Este comportamento de visitar o código, alterar modificar os nomes de variáveis e funções para deixá-los mais descritivos, evitar repetição de código, dentre outras práticas, está fortemente associado ao conceito de *Clean Code* (código limpo” em inglês) (Latte, Henning, & Wojcieszak, 2019), uma boa prática propagada pela comunidade de desenvolvimento de

software. Para seguir este conceito, foram utilizados os conceitos de abstração e modularização, para aprimorar o desenvolvimento do app desde seu início. Abaixo serão descritas algumas mudanças feitas na estrutura de alguns componentes do código:

Sliders

Neste caso, anteriormente o código apresentava extensa repetição. Mudando apenas os números utilizados para sua identificação. Na forma recente, o código está modularizado e abstraído de tal forma que, caso futuramente seja utilizado um novo braço robótico com ainda mais graus de liberdade, basta alterar um número, e o número correspondente de sliders serão criados. Além disso, se torna mais esteticamente agradável o código, visto que a grande quantidade de repetição foi removida.

Funções Wi-Fi

Uma boa prática para desenvolvimento de aplicativos é a de separar os componentes das funções utilizadas, importando as funções necessárias sob demanda, podendo elas serem utilizadas em diversos componentes diferentes. Além disso, auxilia na manutenção do código, visto que todas elas se encontram em um arquivo específico, então, por exemplo, caso o app passe a utilizar outra biblioteca para conexão Wi-Fi, apenas o arquivo de funções precisará ser alterado, visto que a parte visual do código, de uma forma geral, independe da biblioteca atual.

Componentes

Justamente para manter esse conceito de praticidade de manutenção e tornar o código mais denso e fechado em pequenos pacotes, os diversos componentes foram separados em arquivos e pastas distintas de tal forma que cada um mantém sua lógica independente do resto do aplicativo, assim facilitando que sejam utilizados em outros apps, ou que sejam utilizados de formas diferentes pela tecnologia de envio de dados utilizada.

Estilos

Outra boa prática é a de manter todos os estilos referentes ao app em um mesmo arquivo e/ou numa mesma pasta. Além disso, foram utilizadas algumas funções para remover a quantidade excessiva de repetição através de componentes que tinham a mesma estrutura estilística, coisa que seria ainda mais extensa caso os estilos estivessem separados.

App desenvolvido

O app final apresenta uma estrutura responsiva aos comandos do usuário e consegue enviar os valores dos slides tanto via Bluetooth, quanto via Wi-Fi.

5.2 Resultados diretos

Além dos resultados indiretos, apresentam-se aqui os resultados diretos, ou seja, aqueles que estão diretamente relacionados às etapas propostas pelo plano de trabalho.

5.2.1 Novas capacidades do aplicativo Android

Após a leitura da documentação oficial do módulo Wi-Fi, foram analisadas algumas possibilidades para o desenvolvimento da transferência de dados via Bluetooth. Uma delas

sendo o protocolo UDP que foi utilizado e apresenta vantagens diferentes do Bluetooth. Durante este documento o termo Wi-Fi e UDP poderão ser utilizados de forma intercambiável, mas com o intuito de expressar o envio de pacotes de dados através de uma rede Wi-Fi utilizando do protocolo UDP da camada de transporte (Issariyakul, 2011).

Comparativo entre Bluetooth e Wi-Fi

No caso do Bluetooth, o que ocorre é o seguinte: O dispositivo a ser controlado apresenta um sinal constante a espera de um sinal de pareamento, o dispositivo controlador envia um sinal de pareamento, ambos se pareiam e a partir daí é estabelecida uma conexão por onde eles mantêm a troca de dados.

Já no caso do Wi-Fi, o que ocorre é o seguinte: O dispositivo a ser controlado se conecta a uma rede Wi-Fi que o controlador também se conectará (ou já está conectado), a partir daí, ele se configura para ficar sempre a espera de pacotes de dados em uma porta da rede arbitrariamente escolhida. Já o dispositivo controlador, também se conecta nesta mesma rede (ou já está conectado) e envia pacotes de dados para esta porta. Assim sendo recebidos pelo controlado.

Na *Tabela 1* abaixo são listadas algumas vantagens e desvantagens referentes ao envio de dados através de Bluetooth e UDP:

Tabela 1: Vantagens e desvantagens - Bluetooth X UDP

	Bluetooth	Wi-Fi (protocolo UDP)
Vantagens	Baixa dificuldade de utilização: Basta parear e o Arduino está pronto para receber os dados através de suas portas seriais.	A conexão é facilmente expansível: Como é necessário o uso de uma rede externa e esta rede pode ser continuamente expandida com o uso de repetidores de sinal, os dados podem ser enviados por distâncias bem mais longas.
Desvantagens	Há limitação de distância para o envio dos dados baseado no alcance que a antena do Bluetooth é capaz de conectar, sendo complicada a sua expansão.	Falta de segurança: qualquer dispositivo conectado a mesma rede é capaz de captar e até mesmo enviar sinais por conta própria o que pode causar problemas de execução no dispositivo que receberá esses dados.

Apesar dos pontos negativos de cada uma, isso não significa que não sejam amenizados por outros meios, por exemplo: a limitação de distância para Bluetooth pode ser amenizada ao utilizar antenas maiores e melhores que conseguem alcançar dispositivos a mais de 350 metros de distância (Woolley, 2017); já a desvantagem do Wi-Fi utilizando o protocolo UDP no quesito segurança pode ser aliviada ao utilizar de um ponto de acesso criado pelo próprio smartphone através do compartilhamento de dados móveis, assim havendo uma conexão direta entre o dispositivo controlador e o controlado.

Desenvolvimento do App: Bluetooth

Quanto ao desenvolvimento da parte Bluetooth do projeto, a modularização dos arquivos anteriores, citada no tópico “5.1.3 Melhorias da estrutura interna do aplicativo Android”, facilitou que diversos componentes fossem utilizados de formas mais criativas e agradáveis à utilização.

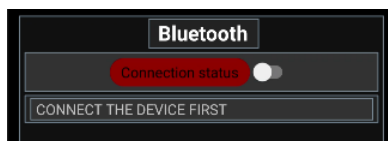


Figura 3: Bluetooth - Ligar o Bluetooth - Fonte: o autor

Inicialmente temos ilustrado pela Figura 3 a *Connection status* (status da conexão) em vermelho, informando abaixo a necessidade de conectar o Bluetooth do dispositivo. Para isso, basta clicar no switch localizado ao lado da Connection status.

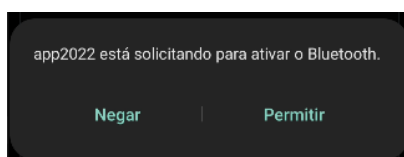


Figura 4: Bluetooth - Permitir acesso do app ao sistema para ativar o Bluetooth - Fonte: o autor

Com isso o App solicita ao usuário que permita o acesso ao Bluetooth (Figura 4).



Figura 5: Bluetooth - Seleção do dispositivo - Fonte: o autor

Após ter a solicitação permitida, a Connection status se torna verde, informando que a conexão foi bem sucedida e a *List of devices* (Lista de dispositivos) aparece, assim permitindo que o usuário selecione qual dos dispositivos gostaria de parear (Figura 5). Com isso, o usuário em seu Smartphone seleciona o módulo Bluetooth HC-05 e assim que o faz, tem o retorno visual, na cor azul, informando que seu toque foi lido (Figura 6). Caso haja falha de pareamento, o dispositivo selecionado passa a ter uma borda vermelha. Um detalhe a ressaltar são os 3 campos presentes em cada um dos dispositivos: name (nome, em inglês) diz respeito ao nome pré-definido para representar aquele dispositivo; [class \(classe, em inglês\) faz referência ao tipo do dispositivo lido](#) (Harold, 2004); address (endereço, em inglês) é o endereço MAC do dispositivo (Sari, Supiyandi, Siahaan, & S.Kom, 2017).

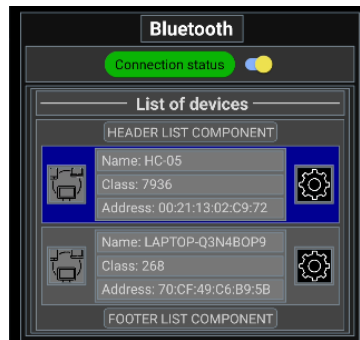


Figura 6: Bluetooth - Dispositivo selecionado - Fonte: o autor

Por fim temos o Smartphone pareado corretamente ao HC-05, ilustrado pela cor verde no dispositivo escolhido, como pode ser visto pela Figura 7, assim fazendo surgir a listagem de sliders que foram definidas previamente para teste. Vale ressaltar que embora na imagem apareça o termo “LED” ele apenas é usado durante os testes. O termo correto seria servomotor. Outro detalhe é que neste exemplo apenas aparecem 4 servos, porém, ao trocar um único valor, podemos fazer aparecerem quantos servos forem necessários, nesse caso, a quantidade correta seriam os 7 servomotores referentes aos 7 graus de liberdade.

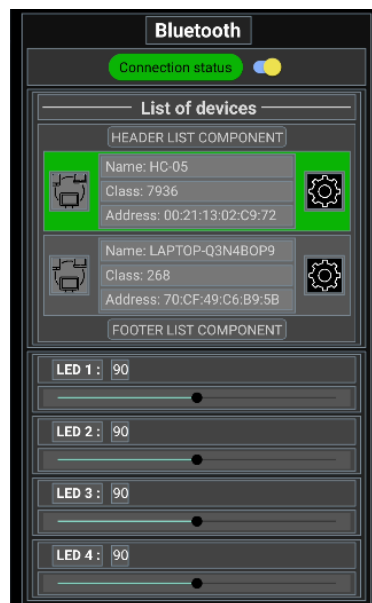


Figura 7: Bluetooth - Dispositivo conectado - Fonte: o autor

Outro avanço obtido no desenvolvimento do envio de dados via Bluetooth foi a superação do bug encontrado no projeto anterior. Recapitulando, o bug ocorria de forma que o HC-05, ao invés de ler os dados enviados, ele lia os seguintes valores: “192 5 0 0 250 192”, ou melhor:

“192

‘a quantidade de caracteres da mensagem enviada parte 1’

‘a quantidade de caracteres da mensagem enviada parte 2’

‘a quantidade de caracteres da mensagem enviada parte 3’

‘255 – a quantidade de caracteres da mensagem enviada parte 1’

192”

Para obtermos o movimento do braço robótico, é necessário um dígito equivalente a qual servo será movido e um número inteiro entre 0 e 180 que defina qual será a angulação final para a qual ele deverá seguir. Para contornar o bug, e fazer com que, por exemplo, o servo 3 se movimente 128 graus, anteriormente enviaríamos uma string contendo o valor 3128, mas para essa situação seriam enviados $256 \times 3 + 128$ caracteres, pois desta forma, a “parte 1” recebida seria o ângulo para o qual o servo deve rumar, e o dígito presente na “parte 2” seria o identificador de qual servo deve ser movido.

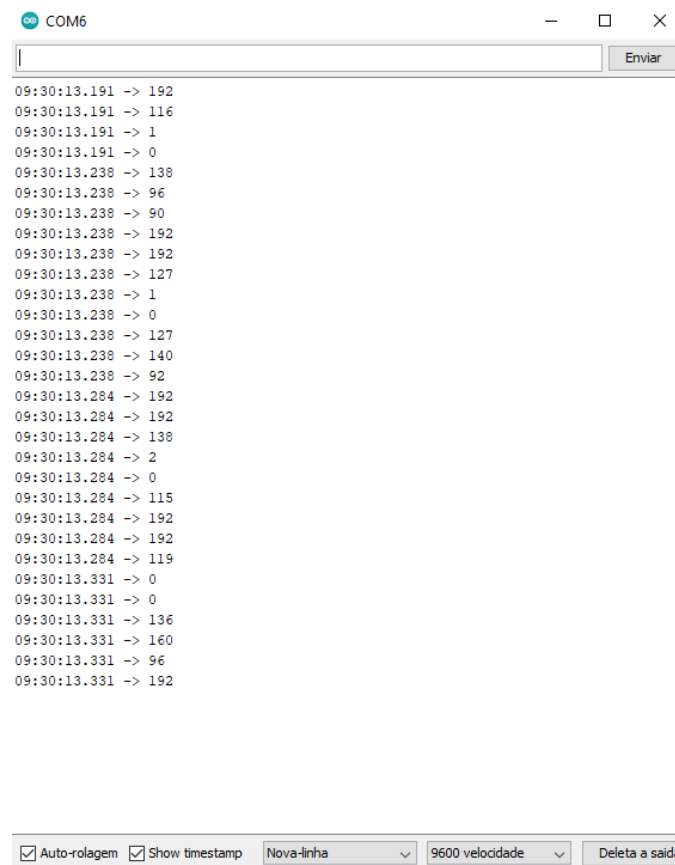


Figura 8: Bluetooth - Informações recebidas no monitor serial após contornar o bug - Fonte: o autor

Dessa forma, como ilustrado pela Figura 8, podemos ignorar os valores iguais a 192, selecionar o primeiro valor após o 192 como sendo o identificador do servo e o segundo valor como o ângulo, dessa forma, mesmo que limitados à uma forma não muito eficaz de lidar com o problema, este avanço não deixa de ser uma vitória em relação a um empecilho existente no projeto anterior.

Desenvolvimento do App: Wi-Fi

Nesse momento, o objetivo principal foi fazer com que o WireShark detectasse algum pacote de dados que tivesse sido enviado pelo smartphone. Após mais pesquisa e desenvolvimento, a etapa 2 do TDD foi atingida. Com isso o código foi modificado e aprimorado para se tornar mais modularizado e novas funções auxiliares foram criadas.

Para conseguir enviar e receber os dados pelo aplicativo, foi utilizada a API react-native-UDP que permitiu o envio de pacotes de dados via Wi-Fi para o NodeMCU Amica. A implementação dessa lógica foi feita em um arquivo de funções Wi-Fi é composto de basicamente 3 funções básicas: Obter o socket criado para o protocolo UDP utilizando do ipv4; configurar esse socket relacionando-o a uma porta local para a qual se quer ler os dados; E por fim a função de envio de dados que depende apenas de 3 parâmetros essenciais: a mensagem em si em formato de string que, diferente do Bluetooth, não precisa estar codificado em base 64.

Tendo alcançado o objetivo do envio de dados pelo app desenvolvido, o foco retorna ao desenvolvimento do código de recebimento de dados pelo Arduino, sobre o qual será comentado mais detalhadamente abaixo.

5.2.2 Melhorias do Hardware

Optou-se por deixar esta etapa em último plano devido a sua natureza mais arriscada, ou melhor, por apresentar maiores riscos de danos aos dispositivos utilizados devido ao uso de corrente e/ou voltagens indevidas. Para isto, seria preferível o acompanhamento presencial do orientador, necessidade que não foi cumprida devido à pandemia do COVID-19. Por isso optou-se pelo foco maior nas outras etapas que envolviam o desenvolvimento de software.

5.2.3 Melhorias do código Arduino

Já tendo anteriormente a lógica necessária para o recebimento de dados via bluetooth, faltava implementar a lógica de recebimento de dados via Wi-Fi. Com a já comentada leitura da documentação, foi bem direta a implementação dessa parte.

O código é composto pela parte de conexão à rede Wi-Fi existente, pela definição de qual porta ele utilizará, e então a constante checagem se algum pacote foi identificado nessa porta, assim que ele for identificado, a string recebida pode ser tratada de forma idêntica a quando ela é recebida pelo Bluetooth (antes de ocorrer o bug).

6 Conclusões

Como esclarecido nos tópicos superiores, as etapas do plano de trabalho foram em sua maioria bem sucedidas, apenas com ressalvas nos tópicos que apresentavam riscos aos equipamentos onde optou-se por evitar danos, visto que a pandemia do COVID-19 perdurou, e ainda perdura durante todo o período de realização deste projeto, assim impedindo reuniões presenciais do bolsista com seu orientador (Moraes, 2020).

Apesar disso, foi notável a evolução proporcionada pelo estudo necessário para o desenvolvimento da pesquisa acadêmica. Evolução essa que não só permitiu solucionar diversos tópicos abordados no projeto passado como pontos de melhoria, como também trouxe novas funcionalidades que não haviam sido pensadas e que foram viabilizadas pela refatoração do código anterior e nova construção do código já visando a abstração e modularização.

Como ilustrado pela Figura 7, foram feitas mudanças nas funcionalidades do aplicativo Android, adicionando a capacidade de envio de dados via Wi-Fi e também alterando a interação do usuário com componentes invisíveis que surgem à medida que o usuário configura a conexão.

Foram feitas melhorias no código Arduino de tal forma que os dados podem ser recebidos via Bluetooth e também via Wi-Fi, embora ainda não simultaneamente. Como as correções de funcionalidade física do braço robótico não ocorreram e eram necessárias para um desenvolvimento seguro sem danos para o braço, o desenvolvimento de códigos Arduino para suavização do movimento também acabaram sendo afetadas.

Entretanto, extensivos testes foram feitos para comprovar a eficácia tanto do recebimento quanto do envio de dados via Wi-Fi.

7 Referências

- Azevedo, G. T., & Maltempi, M. V. (2020). Processo de Aprendizagem de Matemática à luz das Metodologias Ativas e do Pensamento Computacional. *Ciência & Educação*. doi:<https://doi.org/10.1590/1516-731320200061>
- Dias, J. V. (2022). *Repositório da iniciação científica*. (GitHub) Acesso em 2022, disponível em https://github.com/jvfd3/IC-2019_2022-Robotic_Arm
- Dias, J. V., & Montané, F. A. (2021). Publicação Atena Editora. Em D. autores, *Coleção desafios das engenharias: engenharia de computação 3* (Vol. 3, pp. 141-161). Brasil: Atena Editora. doi:10.22533/at.ed.192212911
- Equipe do WireShark. (2022). *Descrições sobre o WireShark*. Fonte: WireShark: <https://www.wireshark.org/>
- Godoy, S. L., & Noronha, A. P. (Junho de 2005). Instrumentos psicológicos utilizados em seleção profissional. *Revista do Departamento de Psicologia. UFF*. doi:<https://doi.org/10.1590/S0104-80232005000100011>
- Grokhotkov, I. (14 de Maio de 2017). ESP8266 Arduino Core Documentation. 77. Acesso em 2022, disponível em https://buildmedia.readthedocs.org/media/pdf/arduino-esp8266/docs_to_readthedocs/arduino-esp8266.pdf
- Harold, E. R. (2004). O que são classes do Bluetooth? Em E. R. Harold, *Java I/O* (p. 244). Acesso em 2022
- Issariyakul, T. H. (2011). *Transport Control Protocols Part 1: An Overview and User Datagram Protocol Implementation*. In: *Introduction to Network Simulator NS2*. Boston, MA, Estados Unidos da América: Springer, Boston, MA. doi:https://doi.org/10.1007/978-1-4614-1406-3_9
- Latte, B., Henning, S., & Wojcieszak, M. (18 de Fevereiro de 2019). Clean Code: On the Use of Practices and Tools to Produce Maintainable Code for Long-Living. *Proceedings of the Workshops of the Software Engineering Conference*, pp. 96-99. Acesso em 2022, disponível em <https://oceanrep.geomar.de/id/eprint/45829/>
- Livro sobre Revolução industrial. (2004). *CAMINHOS DE GEOGRAFIA*(<https://seer.ufu.br/index.php/caminhosdegeografia/article/download/15327/8626/58099>), 96. Acesso em 2022

- Moraes, R. F. (Setembro de 2020). Determinants of physical distancing during the covid-19 epidemic in Brazil: effects from mandatory rules, numbers of cases and duration of rules. *Scielo, Ciênc. saúde coletiva*(25). doi:<https://doi.org/10.1590/1413-81232020259.21892020>
- Sari, R. D., Supiyandi, S., Siahaan, A. P., & S.Kom, M. M. (Agosto de 2017). A Review of IP and MAC Address Filtering in Wireless Network Security. Acesso em 2022, disponível em https://www.researchgate.net/publication/319292293_A_Review_of_IP_and_MAC_Address_Filtering_in_Wireless_Network_Security
- Teixeira, F. (2014). Livro sobre UX. Em F. Teixeira, & C. d. Código (Ed.), *Introdução e Boas Práticas em Ux Design* (1 ed., p. 252). Fonte: <https://books.google.com.br/books?id=vWuCCwAAQBAJ&lpg=PP1&ots=kTVrpP834k&dq=ux&lr&hl=pt-BR&pg=PT107#v=onepage&q=ux&f=false>
- Woolley, M. (13 de Fevereiro de 2017). *Exploring Bluetooth 5* —. Fonte: Bluetooth: <https://www.bluetooth.com/blog/exploring-bluetooth-5-going-the-distance/>

8 Perspectivas de continuidade do trabalho

Espera-se que este projeto seja continuado em algum momento posterior. Para auxiliar o bolsista futuro, todo o material utilizado durante todos esses anos de desenvolvimento como bolsista e organizado se encontrará disponível no [GitHub do atual bolsista](#) (Dias, Repositório da iniciação científica, 2022).

Como possíveis vertentes de desenvolvimento do projeto vê-se parte do que foi disposto no projeto anterior, sendo que alguns dos tópicos abordados foram concluídos, outros removidos ou alterados.

8.1 Hardware

- Analisar o consumo de energia para averiguar se é isto que causa a insuficiência de força no braço robótico;
- Impedir que o braço robótico tombe com movimentos bruscos;
- Utilizar o módulo Bluetooth e o módulo wi-fi simultaneamente (melhoria operacional);
- Buscar maneiras de evitar o “*jittering*” (efeito que causa tremor durante a operação dos servomotores);
- 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);
- Pesquisar sobre as tecnologias 6LoWPAN e BLE.

8.2 Software Arduino

- Posição inicial dos Servomotores (Configurar para manterem na mesma posição ao ligar);

- b) Permitir que o braço seja controlado simultaneamente pelo Wi-Fi e pelo Bluetooth;
- c) Pesquisar, calcular e ilustrar o Envelope de Trabalho Robótico do braço robótico;
 - a. Impedir que o braço robótico encoste no chão (Limitar as angulações dos diversos servomotores);
- d) Tornar Servos mais suaves durante a movimentação (Pesquisar sobre o uso da Biblioteca ServoEasing e sobre o controlador PID);
- e) Garantir a proteção dos servos ante esforço excessivo (youtu.be/LKJLCJvyVdk);
- f) Mudança em relação a movimentação do braço (youtu.be/RkNgRe8X4iY)
 - a. Estudo sobre Cinemática Direta (Input: ângulos das articulações; output: coordenadas x, y, z);
 - b. Estudo sobre Cinemática inversa (Input: coordenadas x, y, z; Output: ângulos das articulações);
 - c. Se mover nos eixos X, Y e Z;
 - d. Se mover nas angulações yaw (guinada), pitch (arfagem) and roll (rolamento), mantendo a extremidade do braço centralizada.

8.3 Software App

- a) Finalizar a integração entre a conexão Bluetooth e Wi-Fi no mesmo App;
- b) Não mostrar na lista de dispositivos que estiverem desligados;
 - a. 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;
- c) Melhorar o visual de todo o aplicativo tornando mais intuitivo e atraente.

9 Participação em congressos e trabalhos publicados ou submetidos e outras atividades acadêmicas e de pesquisa

Foi publicado através da [Atena Editora](#) um [resumo do projeto anterior](#) (Dias & Montané, Publicação Atena Editora, 2021).

10 Data e assinatura do bolsista



2 de maio de 2021.

11 Data e assinatura do orientador



2 de maio de 2021.