



UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
FACULDADE DE ENGENHARIA ELÉTRICA E BIOMÉDICA

Wederson Medeiros Silva

Sistema de Aquisição e Transmissão de Dados Usando Sensor de Flexão Bioinspirado

Belém – Pará

2018

Wederson Medeiros Silva

Sistema de Aquisição e Transmissão de Dados Usando Sensor de Flexão Bioinspirado

Trabalho de Conclusão de Curso apresentado
para obtenção do título de Bacharel em
Engenharia da Computação. Instituto de
Tecnologia. Faculdade de Engenharia da
Computação e Telecomunicações. Universi-
dade Federal do Pará.

Orientador: Prof. Dr. Roberto Menezes Rodriguez

Coorientador: Prof. Dr. João Crisóstomo Weyl A. Costa

Belém – Pará

2018

Wederson Medeiros Silva

Sistema de Aquisição e Transmissão de Dados Usando Sensor de Flexão Bioinspirado/ Wederson Medeiros Silva. – Belém – Pará, 2018-

36 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Roberto Menezes Rodriguez

Coorientador: Prof. Dr. João Crisóstomo Weyl A. Costa

Trabalho de Conclusão de Curso – Universidade Federal do Pará – UFPa
Instituto de Tecnologia – ITEC

Faculdade de Engenharia da Computação e Telecomunicações – FCT, 2018.

1. Modo fantasma de segunda camada. 2. Taxa agregada. 3. *Vectoring*. 4. EVM. I.
Prof. Dr. Roberto Menezes Rodriguez. II. Universidade Federal do Pará. III. Faculdade
de Engenharia Elétrica e Biomédica. IV. Sistema de Aquisição e Transmissão de
Dados Usando Sensor de Flexão Bioinspirado

Wederson Medeiros Silva

Sistema de Aquisição e Transmissão de Dados Usando Sensor de Flexão Bioinspirado

Trabalho de Conclusão de Curso apresentado
para obtenção do título de Bacharel em Enge-
nharia da Computação. Instituto de Tecnolo-
gia. Faculdade de Engenharia da Computação
e Telecomunicações. Universidade Federal do
Pará.

Trabalho aprovado. Belém – Pará, 24 de dezembro de 2018:

Conceito: .

Prof. Dr. Roberto Menezes Rodriguez
Orientador

**Prof. Dr. João Crisóstomo Weyl A.
Costa**
Coorientador

Prof. Dr. Gilvan Borges
Convidado 1

Belém – Pará
2018

Este trabalho é dedicado...

Agradecimentos

Agradeço ao Professor Dr. ...

“Não vos amoldeis às estruturas deste mundo, ...)

Resumo

O modo *fronthaul* de redes 5G ...

Palavras-chaves: Modo fantasma.

Abstract

Phantom mode has...

Keywords: Phantom mode.

Lista de ilustrações

Figura 1 – Movimento do dedo através do tendão.	16
Figura 2 – Sensor de Flexão.	17
Figura 3 – Funcionamento do potenciômetro linear.	17
Figura 4 – Placa Arduino modelo Nano.	18
Figura 5 – Módulos RF transmissor (esq.) e receptor (dir.).	18
Figura 6 – (a) Sistema biomecânico e (b) sistema desenvolvido.	20
Figura 7 – (a) Mão em posição inicial e (b) Mão após flexão.	20
Figura 8 – Movimentos da luva de (a) flexão e (b) extensão.	21
Figura 9 – Comandos a partir de flexões e extensões para controle do carrinho . . .	25
Figura 10 – Costura do transdutor em um dedo.	28
Figura 11 – (a) Potenciômetro adaptado e (b) bateria.	29
Figura 12 – Dimensões aproximadas da PCI e seus componentes.	29
Figura 13 – Desenhos projetados no Kicad do (a) esquemático e (b) PCI.	30
Figura 14 – (a) Placa de fenolite cobreada e (b) com os componentes soldados. . . .	31
Figura 15 – Luva montada.	31
Figura 16 – (a) Transmissão do sinal. (b) Recepção do sinal.	32

Lista de tabelas

Tabela 1 – Valores de A para B	23
Tabela 2 – Valores de A para B e remapeados (R).	24
Tabela 3 – Valores recebidos dos dedos e comandos de movimento	27

Lista de abreviaturas e siglas

AM	<i>Amplitude Modulation</i>
I	Corrente
IDE	<i>Integrated Development Interface</i>
$k\Omega$	<i>Kilohm</i>
LED	<i>Light-Emitting Diode</i>
LiPo	<i>Lithium-ion Polymer</i>
mAh	<i>Milliampere Hour</i>
Mhz	<i>Megahertz</i>
mm	<i>Millimeter</i>
PCI	Placa de Circuito Impresso
PVC	<i>Polyvinyl Chloride</i>
R	Resistência
RF	<i>Radio Frequency</i>
V	Tensão

Listas de símbolos

α	Contante de atenuação
β	Contante de fase
γ	Contante de propagação
Γ_L	Coeficiente de reflexão na carga
δ	Constante da restrição de potência transmissão
Δ_f	Subcanais ou tons em Hz
Γ	Gap de <i>RSIR</i>
Λ	Matriz que contém os elementos da diagonal principal de H
ρ	Máscara espectral utilizada pelo sistema DSL
σ^2	Densidade espectral de potência do ruído Gaussiano branco aditivo

Sumário

1	INTRODUÇÃO	15
2	REFERENCIAL TEÓRICO	16
2.1	Movimentação dos dedos	16
2.2	Sensor Flex	16
2.3	Potenciômetro	16
2.4	Arduino	17
2.5	Módulo RF 433 Mhz	18
3	TRABALHO PROPRIAMENTE DITO	19
3.1	Introdução	19
3.2	Mecânica Bioinspirada	19
3.2.1	Transdutor de Flexão	19
3.2.2	Adaptação na Luva	19
3.3	Programação do microcontrolador	21
3.3.1	Digitalização do sinal	21
3.3.2	VirtualWire	22
3.4	Ajustes	23
3.4.1	Introdução	23
3.4.2	Deslocamento e número de posições	23
3.4.3	Remapeamento	24
3.5	Protocolo de comunicação	24
3.5.1	Introdução	24
3.5.2	Posições e movimentos	25
3.5.3	Software de recepção	25
3.6	Assembling	27
3.6.1	Transdutor de Flexão	27
3.6.2	Escolha de componentes	27
3.6.3	Posicionamento de componentes	29
3.6.4	Placa Embarcada	30
4	BUFFER	32
4.1	Transmissão e Recepção de Dados	32
5	ANÁLISES E RESULTADOS	34
5.1	Configurações	34

5.2	Testes	34
5.3	Resultados	34
6	CONCLUSÃO	35
6.1	Conclusões	35
6.2	Trabalhos Futuros	35
REFERÊNCIAS		36

1 Introdução

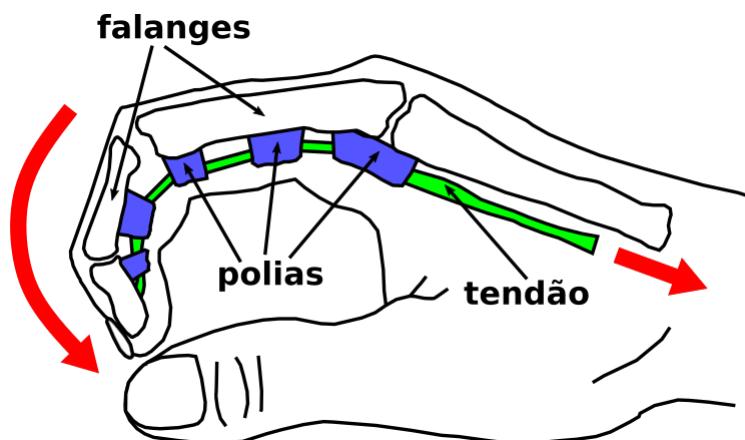
Contextualização; Estado da arte (se tiver); Motivação; O que vai fazer; Metodologia; O que terá no resto do documento;

2 Referencial Teórico

2.1 Movimentação dos dedos

Na mão, os tendões funcionam como cordas que conectam os músculos do antebraço aos ossos da mão. Nos dedos, os tendões passam por dentro de uma série de polias, que formam uma espécie de túnel. Isso permite manter os tendões próximos aos ossos da mão, aumentando a força nos dedos e diminuindo o gasto de energia. Ao movimentar o dedo, o músculo se contrai para que o tendão deslize por entre as polias. (OLIVEIRA, 2018)

Figura 1 – Movimento do dedo através do tendão.



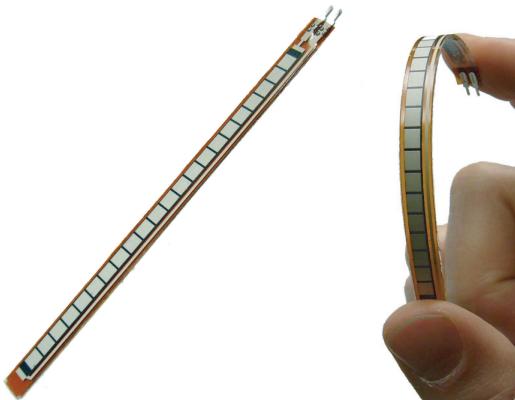
2.2 Sensor Flex

Sensores de flexão, mais conhecidos como sensores flex, são resistores analógicos que trabalham como divisores de tensão analógicos. Dentro desses sensores existem elementos resistivos de carbono junto a um fino substrato flexível. Mais carbono significa menos resistência. Quando o substrato é torcido o sensor produz uma resistência relativa ao raio da torção. (M, 2013).

2.3 Potenciômetro

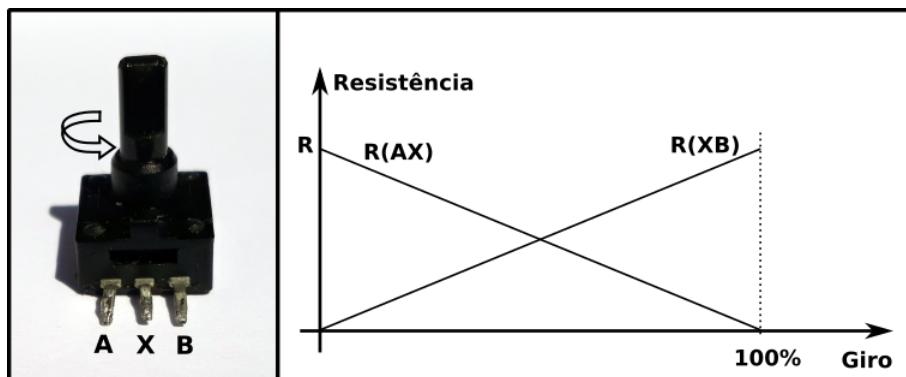
O potenciômetro é um componente eletrônico que permite, através do giro do seu eixo, a variação da resistência entre seus terminais. Eles são constituídos por um elemento de resistência, que pode ser de carbono ou fio de nicromo, sobre o qual corre uma lingueta, denominada cursor. Dentre as características do potenciômetro estão o valor máximo de sua resistência, seu número de voltas, seu grau máximo de giro (aproximado) e se ele é do tipo linear ou logarítmico (BRAGA, 2012).

Figura 2 – Sensor de Flexão.



Fonte – Site Sparkfun.

Figura 3 – Funcionamento do potenciômetro linear.



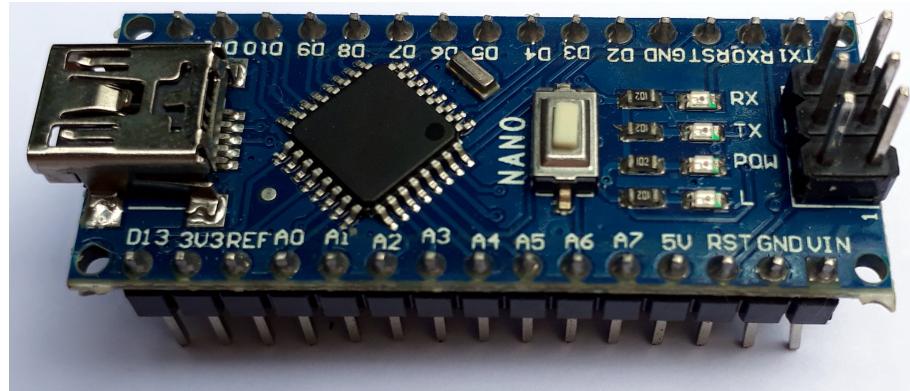
Fonte – Modificado de (BRAGA, 2012).

Segundo a lei de Ohm ($V = R \cdot I$), dada uma corrente constante, ao variar a resistência teremos uma variação da tensão. Sendo assim, ao girar o eixo do potenciômetro, dependendo do sentido do giro, perceberemos um aumento ou diminuição da tensão naquele ponto. Partindo de um ponto extremo com resistência mínima até o outro ponto extremo no qual a resistência deverá ser a máxima característica do componente.

2.4 Arduino

O Arduino é uma plataforma eletrônica de código aberto que é baseada em *hardware* e *software* fáceis de usar. As placas Arduino são capazes de ler entradas como o acionamento de um sensor, o pressionamento de um botão, ou uma mensagem do Twitter. Pode transformar essas entradas em saídas como a ativação de um motor, o acendimento de um LED ou até a publicação de algo online. O comportamento dessa placa pode ser programado usando sua IDE (*Integrated Development Interface*), que por sua vez, envia as instruções necessárias para o microcontrolador instalado na placa.(LICENSE, 2018)

Figura 4 – Placa Arduino modelo Nano.

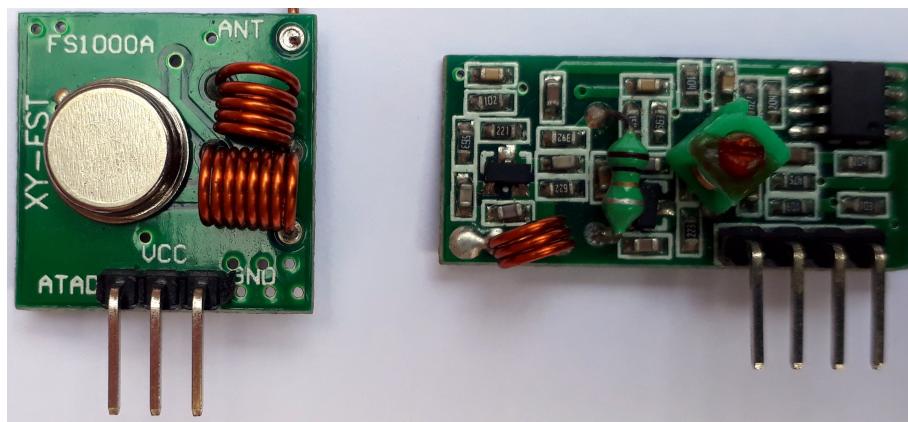


Fonte – Produzido pelo autor.

2.5 Módulo RF 433 Mhz

O módulo de RF (Rádio Frequênciā) 433 Mhz é composto por um par que contém um transmissor e um receptor, opera com modulação AM (*Amplitude Modulation*) e é uma alternativa para projetos de baixo custo que queiram usar comunicação sem fio entre microcontroladores Arduino ou outros. O par de módulos pode alcançar até 200 metros sem obstáculos, usando antenas e dependendo da tensão aplicada.(DIGITAL, 2018)

Figura 5 – Módulos RF transmissor (esq.) e receptor (dir.).



Fonte – Produzido pelo autor.

3 Trabalho Propriamente Dito

3.1 Introdução

O capítulo a seguir está organizado em 5 seções. Na primeira delas explica-se a movimentação mecânica do sistema e como as posições de flexão e extensão dos dedos são obtidas pelo microcontrolador. A segunda aborda a programação do microcontrolador, mais precisamente o tratamento dos dados recebidos a partir dos potenciômetros. A terceira seção explica os ajustes realizados em software. A quarta explica o protocolo criado para controlar um carrinho via rádio frequência. A quinta e última seção explica como o sistema foi montado.

3.2 Mecânica Bioinspirada

3.2.1 Transdutor de Flexão

Como foi explicitado previamente, através dos tendões, têm-se a movimentação dos dedos na mão. Baseado nessa biomecânica, foi desenvolvido um sistema mecânico semelhante, com o intuito de criar um transdutor de flexão de dedos atrelado a um microcontrolador.

No sistema biomecânico, resumidamente, uma das extremidades do tendão está presa nas falanges do dedo enquanto sua extensão desliza por dentro de polias até o músculo. No sistema desenvolvido, cada transdutor é composto por uma linha de náilon presa ao dedo da luva, guiada através de pequenos segmentos plásticos até o cursor de um potenciômetro. A figura 6 demonstra um pouco dos dois sistemas.

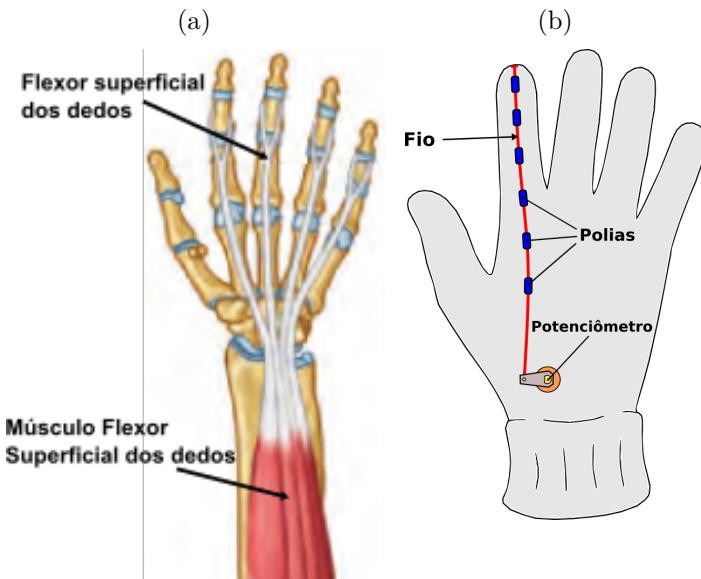
Para verificar se no sistema criado haveria deslocamento do fio durante a flexão dos dedos, com a mão inicialmente estendida com o dorso voltado para cima, uma das pontas de um fio foi presa na extremidade de um dos dedos. O local da outra ponta do fio foi marcada no dorso mão. Como é demonstrado na figura 7 (a).

Após a flexão dos dedos, o fio se movimentou em uma direção e criou um deslocamento (d) do fio em relação ao ponto marcado, mostrado na figura 7 (b).

3.2.2 Adaptação na Luva

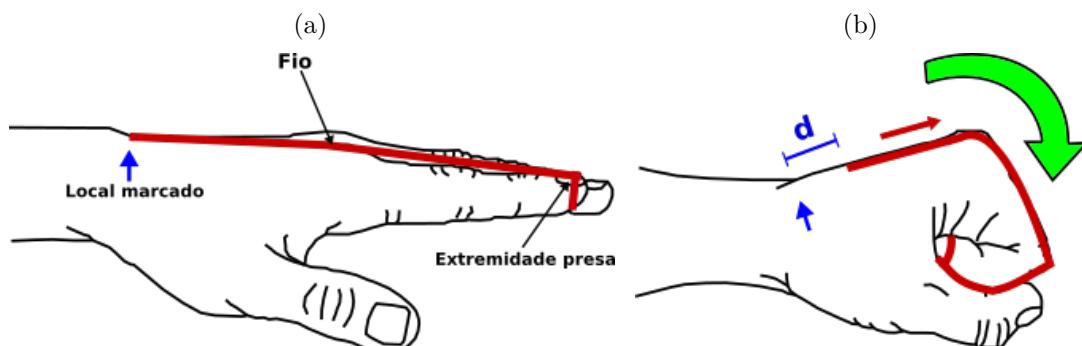
Após o sistema mecânico ser embarcado na luva, com a mão estendida e mantendo o fio de náilon tensionado, ao flexionar o dedo, o fio é puxado pela ponta do dedo e com isso o cursor do potênciometro é variado. Porém, após a flexionar os dedos, ao realizar o

Figura 6 – (a) Sistema biomecânico e (b) sistema desenvolvido.



Fonte – (a) Adaptado de (MOORE; DALLEY; AGUR, 2013) e (b) produzido pelo autor.

Figura 7 – (a) Mão em posição inicial e (b) Mão após flexão.



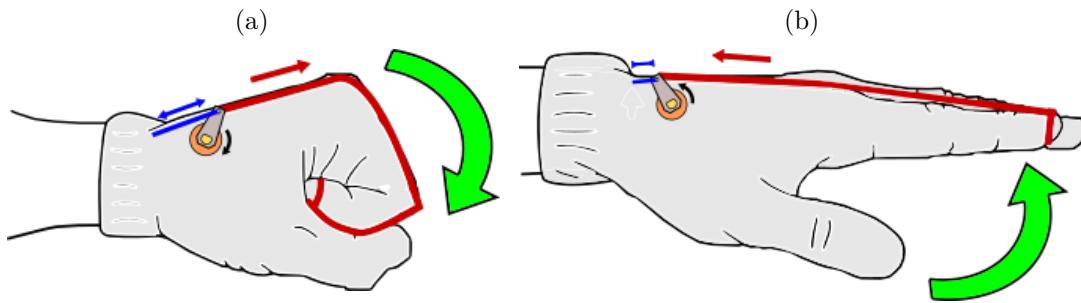
Fonte – Produzido pelo autor.

movimento inverso (extenção), não há força que movimente o fio de náilon e nem o cursor do potenciômetro de volta às suas posições iniciais.

Para possibilitar que o sistema retorne à sua posição inicial, um pequeno elástico foi instalado junto ao cursor do potenciômetro. Usando o elástico durante o movimento de flexão, o cursor gira e estica o elástico. Estando esticado, o elástico busca retomar sua posição de equilíbrio realizando uma força contrária para girar o cursor de volta à sua posição inicial. Figura 8 (a).

Quando há o movimento de extensão do dedo, o elástico puxa o cursor do potenciômetro girando-o em sentido inverso ao que ocorreu durante a flexão. Isso acontece enquanto o elástico estiver esticado o suficiente para exercer força sobre o cursor do potenciômetro. Demonstrado na figura 8 (b)

Figura 8 – Movimentos da luva de (a) flexão e (b) extensão.



Fonte – Produzido pelo autor.

3.3 Programação do microcontrolador

3.3.1 Digitalização do sinal

Na luva, além do transdutor, há um microcontrolador que capta os movimentos do potenciômetro e os digitaliza. O sinal chega em portas analógicas de um Arduino modelo Nano que converte a variação de tensão do potenciômetro em valores inteiros entre 0 e 1023.

Para facilitar a abordagem do trabalho, cada dedo da luva e seu respectivo potenciômetro é representado por um número de 1 a 5, começando pelo dedo mínimo (1) até o polegar (5). Também foram definidas duas posições principais para analisar o movimento dos dedos. Posição A: Quando todos os dedos estiverem estendidos. Posição B: Quando todos os dedos estiverem flexionados. Algo semelhante à figura 8.

Para cada posição, o potenciômetro apresenta uma média de valores digitais. Por exemplo, quando o dedo mínimo (1) está estendido (posição A), seu respectivo potenciômetro apresenta valores em torno de 191. Quando esse dedo é totalmente flexionado (posição B) seus valores ficam em torno de 616.

Porteriormente, para diminuir o tamanho da mensagem a ser enviada, o valor digital apresentado pelo potenciômetro é remapeado para uma faixa de valores inteiros que representam posições inteiras entre 0 e 9. Quando os valores de todos os potenciômetros são remapeados, eles são concatenados em um único número de 5 dígitos, no qual cada dígito representa cada um dos cinco dedos da esquerda para a direita.

Por exemplo, em uma mensagem "69071" o dedo 1 (mínimo) está na posição 6, o dedo 2 (anelar) está na posição 9 e assim sucessivamente até o dedo 5 (polegar) que está na posição 1.

3.3.2 VirtualWire

VirtualWire é uma biblioteca de comunicação para Arduino que possibilita vários Arduinos se comunicarem usando transmissores e receptores RF de baixo custo. Essa biblioteca permite o envio de mensagens curtas, sem endereçamento, retransmissão ou confirmação, como se fosse uma espécie de protocolo UDP (*User Datagram Protocol*), usando modulação ASK (*Amplitude Shift Keying*) (MCCAULEY, 2013).

O uso dessa biblioteca nos códigos da IDE do Arduino, permite abstrair tratamentos de envio e recebimento de dados tais como, sincronização de padrões, balanceamento de bits 0 e 1, e checagem de erros (PAUL, 2018). Dessa forma, para enviar e receber mensagens, basta seguir os padrões de entrada e saída das funções descritas na biblioteca.

No código embarcado no microcontrolador da luva foram definidos, segundo os parâmetros da biblioteca, o pino do transmissor RF, a taxa de transmissão e o tamanho da mensagem a ser enviada. Porém, antes de efetivamente enviar a mensagem é preciso convertê-la em uma string que logo em seguida será transformada em um vetor de char, que é o formato de dado aceito pela função de envio.

Uma das formas de converter valores inteiros em string é concatenar o valor inteiro com uma string (SM, 2018). Por isso, no código desenvolvido, uma string vazia é somada aos valores inteiros remapeados dos potenciômetros. Com isso, os valores serão concatenados e transformados em uma string ao mesmo tempo. Essa string então é transformada em um vetor de char usando a função "toCharArray" antes de ser enviada. Este processo está descrito resumidamente no pseudocódigo abaixo.

```
// Inicia variaveis
string vazia = ""
string mensagem = ""

// Recebe as posicoes de cada dedo
int dedo1 = posicao_dedo1;
int dedo2 = posicao_dedo2;
int dedo3 = posicao_dedo3;
int dedo4 = posicao_dedo4;
int dedo5 = posicao_dedo5;

// Concatena os inteiros e os transforma em string
mensagem = vazia + dedo1 + dedo2 + dedo3 + dedo4 + dedo5;

// Converte a string em um vetor de char
mensagem = mensagem.toCharArray();

// Envia o vetor de char
vw_send(mensagem);
```

3.4 Ajustes

3.4.1 Introdução

Quando os dedos estão em um mesmo grau de flexão, na grande maioria das vezes, os valores detectados para cada dedo são diferentes. Quando a luva está em posição A (extendida), por exemplo, o valor detectado no dedo mínimo (1) fica em torno de 192, enquanto que o anelar (2) apresenta, ao mesmo tempo, um valor em torno de 141.

Além do mais, devido à posição em que cada potenciômetro foi soldado na placa embarcada, alguns potênciômetros apresentam deslocamentos positivos ou negativos para um mesmo movimento.

3.4.2 Deslocamento e número de posições

Para descobrir os valores de deslocamento, cada potenciômetro teve seu valor digital anotado para as posições A e B, que posteriormente foram usados na equação 3.1:

$$\Delta Pos = PosB - PosA \quad (3.1)$$

Usando o módulo de ΔPos é possível calcular o número de posições detectáveis partindo da Posição A até a Posição B, através da equação 3.2:

$$NPos = |\Delta Pos| + 1 \quad (3.2)$$

A tabela 1 mostra os valores obtidos e calculados para cada um dos cinco potenciômetro no deslocamento da posição A para a posição B.

Tabela 1 – Valores de A para B

Dedo	PosA	PosB	ΔPos	NPos
1	191	616	+425	426
2	140	609	+469	470
3	774	360	-414	415
4	728	475	-253	254
5	670	367	-303	304

Fonte – Produzido pelo autor

Quanto maior for o valor de NPos na tabela 1, mais preciso será o sensoriamento do dedo. Sendo assim, o anelar (2) é o dedo com maior potencial de sensoriamento, enquanto que o indicador (4) possui menor potencial.

3.4.3 Remapeamento

Com o intuito de diminuir o tamanho da mensagem a ser enviada pelo transmissor de rádio frequência, foi definido que cada potenciômetro teria apenas 10 níveis de deslocamento representados durante a transmissão. Dessa forma, em uma mensagem numérica, bastaria 1 dígito para representar a posição atual do dedo.

Junto a isso, para diminuir a complexidade do protocolo de comando, foi decidido que não haveriam deslocamentos negativos. Algo que acontece nos dedos 3, 4 e 5 durante o deslocamento de A para B.

Para se adequar aos requisitos descritos, no código embarcado foi usada a função "map", que permite remapear uma faixa de valores em outra menor. Usando tal função, a faixa de entrada que inicialmente poderia variar entre 0 e 1023, foi remapeada para variar entre 0 e 9 valores inteiros.

Além do mais, a função map também permite inverter a saída, de uma forma que os valores de entrada que variam de 1023 a 0 (decrescente) passarão a variar de 0 a 9 (crescente). Este último requisito, permite eliminar os deslocamentos negativos.

Usando as equações 3.1 e 3.2, os valores digitais remapeados (R) foram adicionados à tabela 1, criando a tabela 2:

Tabela 2 – Valores de A para B e remapeados (R).

Dedo	PosA (RA)	PosB (RB)	Δ Pos (Δ R)	NPos (NR)
1	191 (1)	616 (5)	+425 (+4)	425 (5)
2	140 (1)	609 (5)	+469 (+4)	469 (5)
3	774 (3)	360 (6)	-414 (+3)	414 (4)
4	728 (3)	475 (5)	-253 (+2)	253 (3)
5	670 (4)	367 (6)	-303 (+2)	303 (3)

Fonte – Produzido pelo autor

3.5 Protocolo de comunicação

3.5.1 Introdução

Através de um transmissor de rádio frequência, o sistema embarcado na luva envia constantemente mensagens de 5 dígitos que indicam a posição de cada um dos cinco dedos. Para receber esse sinal, é preciso usar um módulo receptor de rádio frequência compatível com o transmissor.

Um sistema receptor foi embarcado em um carrinho usando um segundo microcontrolador Arduino modelo Nano e um módulo receptor RF. Esse carrinho funciona com

motores DC (*Direct Current*) e todo o sistema é alimentado por uma bateria LiPo.

Para possibilitar o controle, conjuntos de posições dos dedos pré definidas indicam movimentos a serem seguidos pelo carrinho.

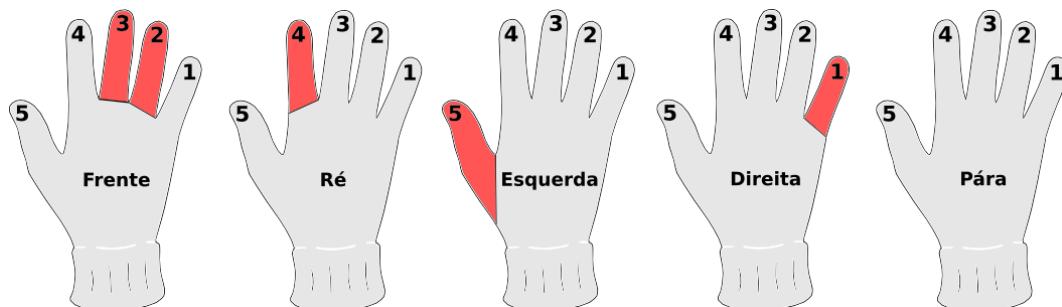
3.5.2 Posições e movimentos

Como foi explicado anteriormente, a mensagem transmitida pela luva consiste em conjunto de 5 dígitos no qual cada um deles indica posições de cada dedo. Na mensagem, o primeiro dígito da esquerda indica a posição do dedo 1, que é o dedo mínimo. O segundo dígito indica a posição do dedo 2, ou seja, o dedo anelar. Essa lógica segue até o quinto dígito que representa a posição do dedo polegar.

5 comandos foram definidos para o carrinho, ir para frente, ir para trás, girar para a esquerda, girar para a direita e parar. Com o intuito de controlar o carrinho através de combinações de flexões e extensões de dedos, 5 conjuntos de posições foram definidas, flexionar os dedos 2 e 3 simultaneamente, flexionar somente o dedo 4, flexionar somente o dedo 5, flexionar o dedo 1 e não flexionar nenhum dedo.

A figura 9 mostra a correspondência entre cada combinação de flexão dos dedos e seu respectivo comando para o carrinho. Apenas os dedos indicados em vermelho devem ser flexionados para validar o comando. Qualquer outra combinação faz o carrinho parar.

Figura 9 – Comandos a partir de flexões e extensões para controle do carrinho



Fonte – Produzido pelo autor.

3.5.3 Software de recepção

A biblioteca VirtualWire também deve estar presente no código do microcontrolador embarcado no carrinho. Para receber a mensagem adequadamente, é necessário definir o pino do receptor RF e taxa de transmissão do transmissor RF, também deve-se criar variáveis para guardar a mensagem recebida e o seu tamanho. As mensagens são recebidas byte a byte.

Uma função da biblioteca verifica se chegou alguma mensagem, ela indica *true* quando isso acontece. Então para ler a mensagem recebida, é necessário criar um *for* que

recebe como parâmetro o tamanho da mensagem recebida. Dentro da função será lida *byte* a *byte* a mensagem enquanto a função *for* for verdadeira.

O pseudocódigo abaixo demonstra o fluxo descrito:

```
// Define as variaveis
int pino_receptor = 8;
int taxa_transmissao = 2000;

// Recebe a mensagem e o seu tamanho
byte msg = get_mensagem;
byte tamanho_msg = get_tamanho_msg;

// Aguarda receber alguma mensagem
chegou_mensagem = vw_wait_rx();

// Verifica se ha mensagem, inicia o for para ler cada byte
if(chegou_mensagem){
    for (int i = 0; enquanto i < tamanho_msg; i++)
        byte_recebido[ i ] = msg[ i ];
}
```

Usando o pseudocódigo apresentado e sabendo que na mensagem recebida o primeiro dígito indica a posição atual do dedo 1, então a função *for* irá rodar pelo menos 5 vezes, armazenando em cada posição do vetor *byte_recebido*[] as informações de cada dedo. Dessa maneira, a posição recebida do dedo 1 estará em *byte_recebido*[0], enquanto que a do dedo 2 está em *byte_recebido*[1]. Sendo assim até o dedo 5 que estará em *byte_recebido*[4].

Cinco funções foram criadas para comandar a ponte H e dessa forma movimentar os motores do carrinho, FRENTE(), TRAS(), ESQUERDA(), DIREITA() e PARA(). Ao chamar qualquer uma dessas funções, o carrinho passa a se movimentar na direção e sentido definidas. Para chamar a função correta baseada na mensagem recebida, foram criadas cadeias de *Ifs* e *Elses* nas quais, quando satisfeitas certas condições, a respectiva função é chamada.

A tabela 3 mostra quais são os conjuntos de valores esperados para todos os dedos simultaneamente em cada comando. Ou seja, para saber qual a combinação de valores para movimentar o carrinho para frente, é só observar a coluna "Frente" da tabela. Qualquer outra combinação além das especificadas na tabela, fará o carrinho ficar parado. Os dedos flexionados estão destacados em vermelho.

Tabela 3 – Valores recebidos dos dedos e comandos de movimento

Dedo	Frente	Tras	Esquerda	Direita	Para
1	< 2	< 2	< 2	> 1	-
2	> 1	< 2	< 2	< 2	-
3	> 3	< 4	< 4	< 4	-
4	< 4	> 3	< 4	< 4	-
5	< 5	< 5	> 4	< 5	-

Fonte – Produzido pelo autor

3.6 Assembling

3.6.1 Transdutor de Flexão

A escolha de materiais foi baseada no menor custo e maior disponibilidade local. Prevendo a possibilidade de costuras no projeto, um par de luvas de algodão foi escolhida. O sistema foi embarcado na luva direita, enquanto que a esquerda poderia ser usada em caso de dano da primeira.

As polias plásticas são feitas a partir de pequenos pedaços de hastes flexíveis com algodão nas pontas. Sendo que apenas a parte plástica foi aproveitada enquanto que o algodão foi descartado.

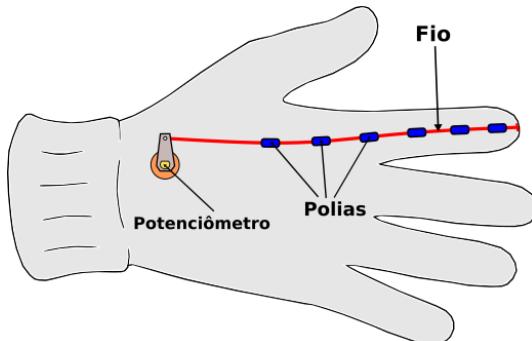
O material da linha que movimenta os cursores dos potenciômetro é náilon. Isso porque, nos testes, ela apresentou boa resistência e durabilidade, além de aparentar ter pouca ou nenhuma elasticidade para a aplicação.

Utilizando a luva escolhida com o dorso voltado para cima, as polias plásticas foram costuradas ao longo dos dedos. Uma das extremidades de cada fio de náilon foi presa na ponta de cada um dos dedos, enquanto que o restante do fio passava por dentro das polias plásticas. A outra extremidade do fio de náilon seria amarrada ao cursor do potenciômetro quando a PCB fosse incluída na luva. A figura 10 demostra o resultado para um dos dedos da luva.

3.6.2 Escolha de componentes

O sistema processamento e transmissão de sinal embarcado na luva, foi projetado para ser móvel, leve, alimentado por uma bateria, caber no dorso da mão e ter custo relativamente menor em relação à soluções com sensores flex tradicionais. O sistema dever ser de fácil reprodução e o mais adaptável possível à outras formas de transmissão além da rádio frequência, caso estas sejam necessárias futuramente.

Figura 10 – Costura do transdutor em um dedo.



Fonte – Produzido pelo autor.

O primeiro desafio foi escolher, dentre os componentes disponíveis, quais seriam utilizados para compor a eletrônica presente na placa de circuito impresso que estaria embarcada na luva.

O potenciômetro foi o primeiro componente a ser definido. Isso porque, este seria o componente que estaria em maior número na placa. O modelo escolhido deveria ser pequeno suficiente para manter uma distância adequada para outros potenciômetros e componentes. Seu cursor deveria ser de fácil giro, para que pudesse ser acionado apenas pelo deslocamento do fio. Seu ângulo de giro total precisava ser mínimo, para que o menor grau de giro correspondesse à maior variação de resistência possível, facilitando assim a percepção pelo microcontrolador.

O modelo de potenciômetro que mais se aproximou das especificações acima foi retirado de servomotores modelo MG996R da marca TowerPro. O potenciômetro encontrado possui dimensão aproximada de $13mm \times 13mm$, resistência máxima de $5k\Omega$, giro máximo de 200° (aproximado) e oferece pouca resistência para girar seu cursor.

Um pequeno pedaço de PVC (*Polyvinyl Chloride*) expandido foi adaptado ao cursor do potenciômetro para facilitar o seu giro, amarrar uma das extremidades do fio de náilon e para prender um elástico. Essa adaptação pode ser vista na figura 11 (a).

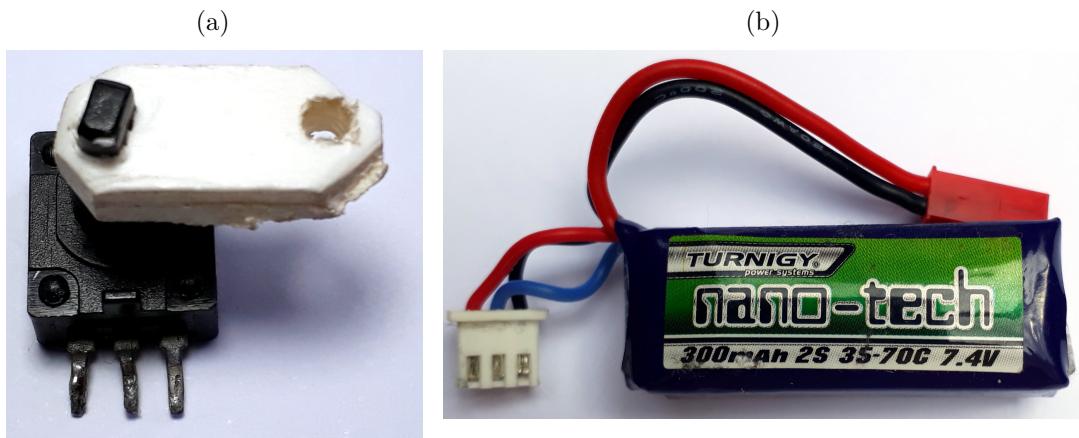
O microcontrolador escolhido para processar os dados recebidos de cada potenciômetro foi o Arduíno modelo Nano. Isso porque ele é leve, ocupa uma área de apenas $45mm \times 17mm$, possui vasta documentação e disponibilidade no mercado, além de ser compatível com diversos módulos externos e possuir custo menor do que outros modelos da família Arduino.

Para transmitir e receber o sinal, foi escolhido o par de RF $433Mhz$, que é leve e de baixo custo comparado à outras soluções de transmissão de dados.

Finalmente, para alimentar esse aparato eletrônico, foi escolhida uma pequena bateria LiPo (*Lithium-ion Polymer*) que possui capacidade de $300mAh$, $7.4V$ de tensão

nominal e ocupa um espaço de $45mm \times 12.5mm$. Esse modelo é mostrado na figura 11 (b).

Figura 11 – (a) Potenciômetro adaptado e (b) bateria.

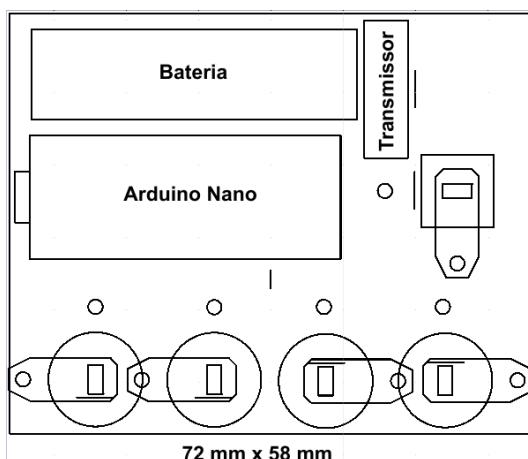


Fonte – Produzido pelo autor.

3.6.3 Posicionamento de componentes

Após medições realizadas na luva que serviu de modelo para o projeto, foi decidido que as dimensões máximas da PCI (Placa de Circuito Impresso) deveriam ser de aproximadamente $72mm \times 58mm$. Isso para que a placa não ficasse muito maior do que o dorso da luva. Sendo assim, usando o software QCAD, que é gratuito para os sistemas Linux, todos os componentes foram organizados dentro da placa baseados em suas dimensões aproximadas. Chegando ao *layout* mostrado na figura 12.

Figura 12 – Dimensões aproximadas da PCI e seus componentes.



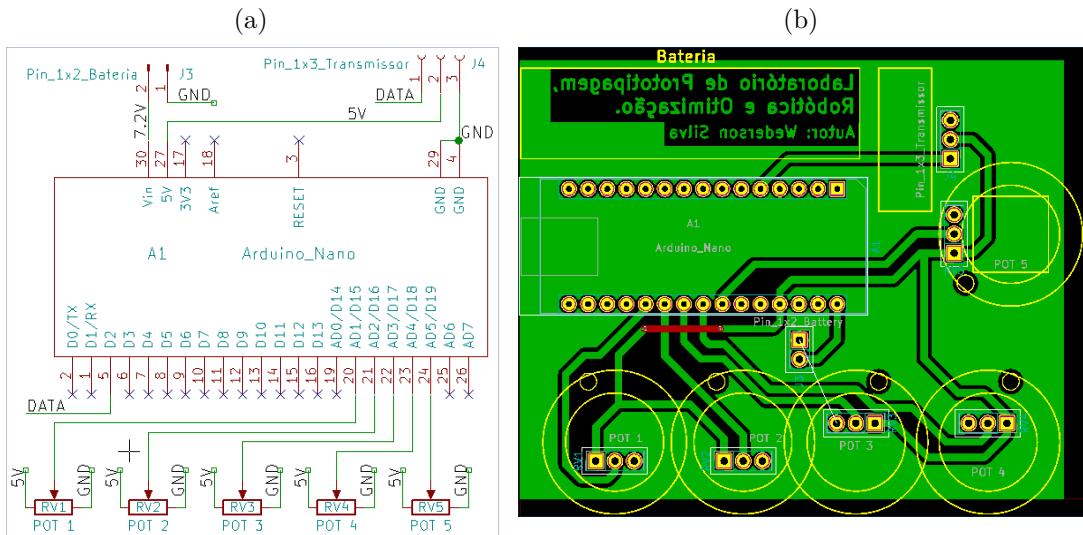
Fonte – Produzido pelo autor.

3.6.4 Placa Embarcada

Os primeiros experimentos de ligação e testes entre os componentes foram realizados ainda em protoboard. Inicialmente, um simples programa lia a variação de um resistor e mostrava na tela do computador. Este e outros programas foram usados para verificar como deveriam ser as conexões entre os pinos, potenciômetros, bateria, módulo transmissor e o microcontrolador Arduíno.

O passo seguinte foi utilizar o software gratuito Kicad para desenhar o esquemático da placa. Posteriormente, o Kicad ainda possibilita projetar uma placa de circuito impresso baseada no esquemático desenhado anteriormente. Através desse software foram criados os desenhos do esquemático e da PCI embarcada mostrados na figura 13.

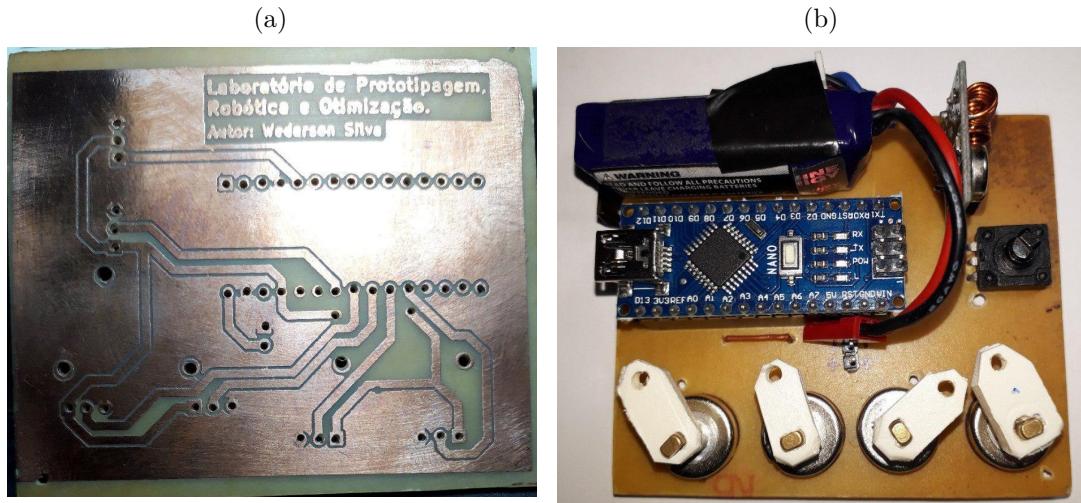
Figura 13 – Desenhos projetados no Kicad do (a) esquemático e (b) PCI.



Fonte – Produzido pelo autor.

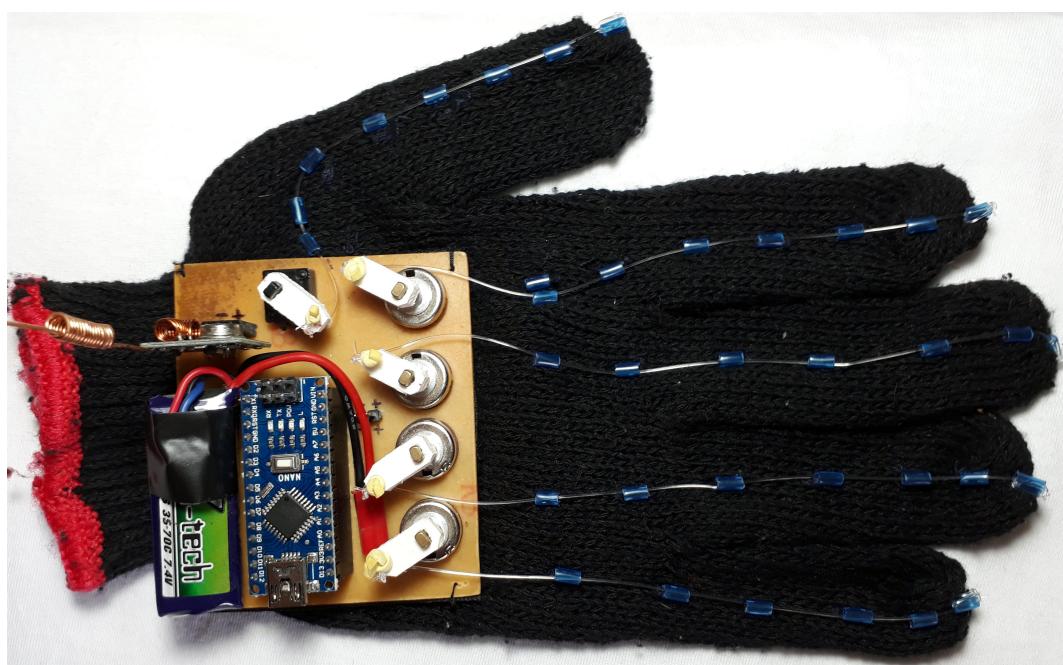
Após a conclusão do desenho da placa, foram efetuados os procedimentos para a confecção desta PCI em uma placa de fenolite cobreada, além de sua posterior soldagem de pinos e componentes. A figura 14 mostra os resultados após o procedimento descrito. Já a figura 15 mostra o resultado final após juntar a luva de algodão e a placa de circuito impresso recém concluída.

Figura 14 – (a) Placa de fenolite cobreada e (b) com os componentes soldados.



Fonte – Produzido pelo autor.

Figura 15 – Luva montada.



Fonte – Produzido pelo autor.

4 BUFFER

Também é possível notar que alguns deslocamentos são positivos enquanto outros são negativos. Isso se dá porque os potênciômetros 1 e 2 foram instalados com seus cursores em sentido oposto aos potênciômetros 3 e 4. Com o intuito de minimizar o espaço entre eles na placa. Isso pode ser observado na figura 15.

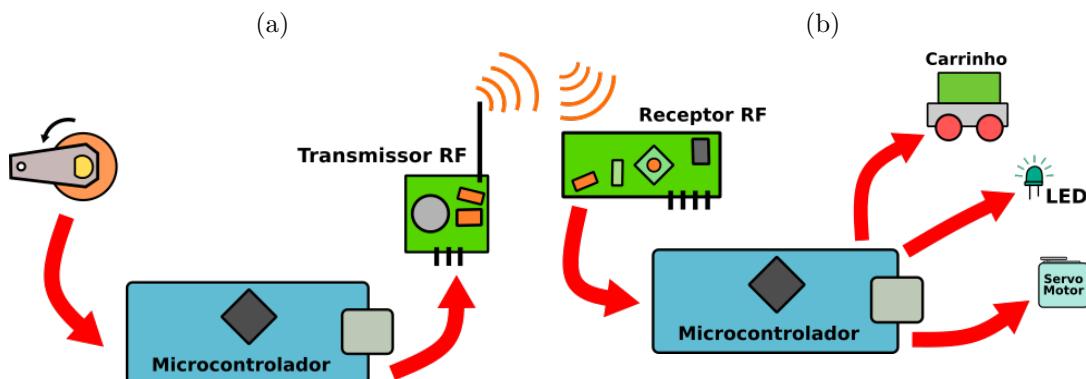
Por conta desse fato, durante o movimento de flexão, os potênciômetros 1 e 2 giram em sentido anti-horário e possuem deslocamento positivo, enquanto os demais giram em sentido horário com deslocamento negativo. No caso do potênciômetro 5, apesar de girar em sentido anti-horário, ele possui um deslocamento negativo. Isso porque foi soldado em posição reversa aos demais, com o propósito de maximizar sua aderência na placa.

4.1 Transmissão e Recepção de Dados

Como foi descrito, durante a movimentação dos dedos, a posição do cursor do potênciômetro é modificada. Sendo assim, para possibilitar o sensoriamento, a variação de cada potênciômetro é captada por um microcontrolador que processa esse sinal antes de despachá-lo para o transmissor. Este por sua vez envia mensagens por rádio frequência, em formato de números inteiros que representam a posição atual de cada dedo. Figura 16 (a).

O módulo receptor de rádio frequência, capta as mensagens e as envia a outro microcontrolador. Este por sua vez, processa a mensagem e transmite aos respectivos componentes e atuadores daquela aplicação. Figura 16 (b).

Figura 16 – (a) Transmissão do sinal. (b) Recepção do sinal.



Fonte – Produzido pelo autor.

Para este trabalho, um pequeno carrinho, foi o sistema escolhido para ser controlado pela luva. Para isso, um protocolo de transmissão foi desenvolvido para traduzir os

movimentos dos dedos da luva em direções para o carrinho.

5 Análises e Resultados

5.1 Configurações

5.2 Testes

5.3 Resultados

6 Conclusão

6.1 Conclusões

6.2 Trabalhos Futuros

Referências

BRAGA, N. C. *Curso de Eletrônica: Eletrônica Básica*. 1. ed. São Paulo - Brasil: Editora Newton C. Braga, 2012. v. 1. PAGES p. ISBN 8565050092, 9788565050098. Citado 2 vezes nas páginas 16 e 17.

DIGITAL, I. *Módulo RF Transmissor + Receptor 433mhz - Instituto Digital*. 2018. Acessado em: 12/08/2018. Disponível em: <<http://www.institutodigital.com.br/pd-f7460-modulo-rf-transmissor-receptor-433mhz.html?ct=76b8b&p=4&s=1>>. Citado na página 18.

LICENSE, C. C. A.-S. . *Arduino - Introduction*. 2018. Acessado em: 12/08/2018. Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>. Citado na página 17.

M, S. K. Indian sign languages using flex sensor glove. *IJETT*, v. 4, p. 2478–2480, 2013. ISSN 2231-5381. Disponível em: <<http://www.ijettjournal.org/volume-4/issue-6/IJETT-V4I6P149.pdf>>. Citado na página 16.

MCCAULEY, M. *VirtualWire, Documentation for the VirtualWire communications library for Arduino*. [S.l.], 2013. V. 1.27. Disponível em: <<http://www.airspayce.com/mikem/arduino/VirtualWire.pdf>>. Citado na página 22.

MOORE, K.; DALLEY, A.; AGUR, A. *Clinically Oriented Anatomy*. Wolters Kluwer Health/Lippincott Williams & Wilkins, 2013. (Clinically Oriented Anatomy). ISBN 9781451119459. Disponível em: <<https://books.google.com.br/books?id=-Le5bc5F0sYC>>. Citado na página 20.

OLIVEIRA, R. K. de. *Dedo em Gatilho - Dr. Ricardo Kaempf - Cirurgia de Mão e Microcirurgia*. 2018. Acessado em: 07/08/2018. Disponível em: <<http://www.ricardokaempf.com.br/services/dedo-em-gatilho/>>. Citado na página 16.

PAUL. *VirtualWire Library, for very cheap wireless communication*. 2018. Acessado em: 27/11/2018. Disponível em: <https://www.pjrc.com/teensy/td_libs_VirtualWire.html>. Citado na página 22.

SM. *Arduino - String Addition Operator*. 2018. Acessado em: 04/12/2018. Disponível em: <<https://www.arduino.cc/en/Tutorial/StringAdditionOperator>>. Citado na página 22.