

Comunicação Wireless

Transmissão e recepção de sinal sonoro utilizando os módulos 433RF e Arduino Uno

Dayana dos Santos Vieira
Departamento de Engenharia Elétrica
Universidade Federal do Pará
Tucuruí, Brasil
dayanavieira.eletr@gmail.com

Alexandre de Araújo Pereira
Departamento de Engenharia Elétrica
Universidade Federal do Pará
Tucuruí, Brasil
alexdeap@gmail.com

Resumo— Radiofrequência (RF) é qualquer uma das frequências de ondas eletromagnéticas que se situam na faixa de 3kHz a 300GHz, que incluem as frequências usadas para comunicações ou sinais de radar. Para receber sinais de rádio, uma antena deve ser usada. No entanto, como a antena captará milhares de sinais de rádio por vez, um sintonizador de rádio será necessário para sintonizar uma frequência específica (ou faixa de frequência). A distância sobre a qual as comunicações de rádio são úteis depende significativamente de outras coisas além do comprimento de onda, como potência do transmissor, qualidade do receptor, tipo, tamanho e altura da antena, modo de transmissão, ruído e sinais de interferência. As ondas terrestres, a dispersão troposférica e as ondas celestes podem alcançar intervalos maiores do que a propagação na linha de visão. O estudo da propagação de rádio permite estimativas de alcance útil a ser feito. Nosso objetivo é projetar e implementar uma comunicação entre dois arduinos através da qual transmitimos e realizamos a recepção de um sinal sonoro, de forma a demonstrar a eficiência da comunicação através dos módulos RFs via Arduino.

Palavra-chave— transmissão; recepção; módulo RF; arduino; ASK.

I. INTRODUÇÃO

É indiscutível a importância das comunicações sem fio na vida do homem moderno. Até se chegar aos sistemas de telefonia celular de última geração, e a tantos outros sistemas de comunicações sem fio que se fazem presentes nos mais diversos setores, alguns fatos históricos merecem ser destacados. A primeira rede de comunicação foi estabelecida na era pré-industrial. Os sistemas de comunicação de longas distâncias transmitiam informação considerando sinais de fumaça, tochas e espelhos refletores de luz. Um sistema complexo e bem elaborado de combinação de sinais tornava possível a transmissão de informação através desses sistemas rudimentares de comunicação. A observação dos sinais provindos dos espelhos, tochas, bandeiras, podiam ser vistos no altos das colinas e nas torres de grandes castelos e muralhas.

Nas tecnologias atuais os conversores A/D e D/A são a base de todo o interfaceamento eletrônico entre o mundo analógico e o mundo digital. Estão presentes na grande maioria dos instrumentos de medida atuais e são os responsáveis pelo aumento significativo nos níveis de precisão e exatidão assim como o barateamento e popularização de instrumentos de

medida digitais. E os conversores D/A convertem uma palavra digital em um sinal analógico sob a forma de uma tensão ou corrente de saída. São formados por elementos passivos, fontes de referência, chaves e AMPOP's. A conversão é, em geral, paralela e o tempo de conversão depende essencialmente da velocidade dos componentes utilizados [1].

A plataforma Arduino vem sendo utilizada de forma acentuada em projetos de sistemas embarcados, seja em de automatização de sistemas ou para aplicações comerciais. As placas arduino vem evoluindo constante e atualmente deixou de ser uma plataforma simples baseada apenas em microcontroladores de 8 bits. Atualmente há no mercado uma vasta variedade de placas arduino desde microcontroladores de 8 bits a microcontroladores de 32 bits. Com o desenvolvimento desta tecnologia barata e com grande aplicações o conhecimento dessa plataforma de hardware e software é um diferencial importante para o profissional da área de sistemas embarcado.

Pensando nisso, para a execução deste projeto serão utilizadas duas placas do Tipo Arduino Uno. Este micro controlador foi escolhido devido ao seu baixo custo e a possibilidade de sua prévia simulação no Proteus. Para este projeto, em especial, utilizou-se o software Proteus 8 Professional, devido a interface atualizada desta placa.

Atualmente, existem algumas maneiras de comunicar o Arduino com o mundo “externo”, como por exemplo: usando Bluetooth com Arduino ou um Arduino Ethernet Shield para verificar o estado de sensores e enviar comandos. No entanto, há uma outra alternativa eficiente e barata para transmissão de dados, que é o uso de módulos transmissores RF (Rádio Frequência), os quais serão utilizados neste projeto.

O modulo RF opera com frequências de rádio. A frequência correspondente varia entre 30 kHz e 300 GHz. Neste sistema RF os dados digitais são representados como variações das amplitudes da onde portadora. Esse tipo de modulação é conhecida como Chaveamento de deslocamento de amplitude - ASK (Amplitude Shift Keying). A transmissão por RF é melhor que IR (infravermelho) por vários motivos. Em primeiro lugar, os sinais através de RF podem percorrer distâncias maiores, tornando-o adequado para aplicações de longo alcance. Além disso, enquanto o IR opera principalmente no modo de linha de visão, os sinais de RF podem viajar

mesmo quando há uma obstrução entre o transmissor e o receptor [3].

A transmissão de RF é mais forte e confiável do que a transmissão de infravermelho. A comunicação RF usa uma frequência específica, ao contrário dos sinais IR, que são afetados por outras fontes emissoras de infravermelho. Este módulo de RF é composto por um transmissor de RF e um receptor de RF. O par transmissor / receptor (Tx / Rx) opera a uma frequência de 434 MHz. Um transmissor de RF recebe dados seriais e transmite sem fio através de RF através de sua antena conectada no pino 4. A transmissão ocorre a uma taxa de 1Kbps - 10Kbps. Os dados transmitidos são recebidos por um receptor RF operando na mesma frequência do transmissor [2]. O módulo RF é frequentemente usado junto com um par de codificador/decodificador. O codificador é usado para codificar dados paralelos para alimentação de transmissão enquanto a recepção é decodificada por um decodificador. [3].

Como dito anteriormente, a codificação utilizada nestes módulos é a ASK/OOK, onde eles modulam a amplitude do sinal, em outras palavras a intensidade, conforme o nível em que se encontra a porta de dados. Portanto se for nível 1 (5V) é emitido sinal na frequência de 433 Mhz, sendo zero não é emitido nenhum sinal naquela frequência, e é por essa modulação/codificação que este tipo de rádio está tão sujeito a interferência, assim é preciso inserir uma segunda codificação sobre a codificação OOK para aumentar sua confiabilidade.

Visto tudo isso, este projeto visa o envio de uma onda senoidal, a qual na recepção deve atender às limitações da audição humana. Por isso, é importante destacar que, as ondas sonoras utilizadas possuem frequência entre 20Hz e 20kHz, para que assim influenciem diretamente na altura do som.

II. DESENVOLVIMENTO

Este artigo tem como objetivo transmitir um sinal de áudio (onda senoidal) de frequência de 490Hz utilizando o transmissor Tx e receber este sinal através o receptor Rx do módulo RF 433MHz utilizando o arduino para a conversão do sinal e envio deste para os módulos. Como no diagrama de blocos na Fig. 1.

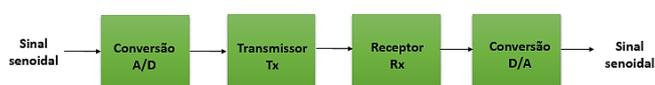


Fig. 1. Funcionamento do circuito em blocos

O sinal aplicado no arduino foi uma onda senoidal de 490Hz aplicada na entrada analógica do arduino, esse sinal analógico foi convertido para um sinal digital e a partir deste sinal foi realizada as leituras da tensão da onda senoidal em espaços de tempo e para esses valores de tempo analisados foram guardados na memória. Utilizando os valores de tempo foi possível encontrar os valores de frequência da onda senoidal, como na Fig. 2.

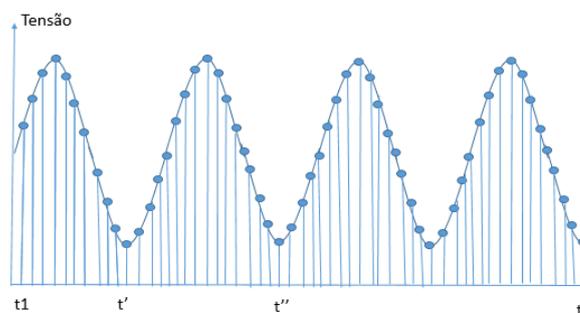


Fig. 2. Leitura da tensão e os tempos de leitura

Com estes valores de tempo discretizados é possível calcular a distância entre dois picos da onda senoidal e deste modo calcular o período da onda utilizando a relação seguir.

$$T = t'' - t' \quad (1)$$

Estes valores de período foram utilizados para calcular a frequência da onda senoidal, utilizando a expressão da frequência a seguir:

$$f = \frac{1}{T} \quad (2)$$

Os valores de frequência calculados são transmitidos pelo transmissor Tx do módulo RF utilizado. O sinal RF gerado pelo arduino é modulado em ASK/OOK (Amplitude-Shift Keying ou On-Off keying), sendo na verdade a modulação OOK o tipo de modulação ASK mais simples possível. Este consiste basicamente do chaveamento de uma portadora para representar um sinal digital, resultando em um sinal RF intermitente que pode ser um pouco difícil de ser analisado.

Com os valores recebidos pelo receptor RF realizamos a conversão digital/analógica para recuperar o sinal mensagem enviada.

Utilizando um gerador de sinal analógico configuramos a tensão senoidal para 4Vpp e Offset de 2V e frequência de 490Hz, este sinal gerado é o sinal mensagem que aplicaremos ao circuito para a transmissão. Antes aplicar o sinal a entrada analógica do arduino utilizamos um divisor de tensão, pois nas portas do arduino apenas podem ser aplicadas tensões de amplitude de até 5V, deste modo este divisor de tensão dividirá pela metade e assim será aplicada apenas a metade da tensão de pico. Para o divisor de tensão utilizamos 2 resistores de 10kΩ. Como a tensão de pico da onda aplicada no circuito é de 6V utilizando o divisor de tensão a tensão na entrada da porta analógica do arduino é de 3Vp.

O circuito do projeto envolveu a utilização de dois *Arduinos Uno R3* juntos de dois módulos de comunicação, um transmissor e um receptor, ao qual foram conectados aos *Arduinos* de modo a estabelecer um canal de comunicação entre eles. Com isso foi possível implementar tarefas específicas para da arduino.

Ao primeiro Arduino (Fig.1) coube a tarefa de ler e estudar uma onda senoidal de amplitude unitária e frequência variável de 260 á 490 Hz gerada por um gerador de sinais conectado a primeira porta analógica do Arduino (A0). A onda senoidal utilizada foi configurada de modo a ter uma tensão variante somente entre valores positivos, pois como indica o manual do arduino, as portas analógicas desses dispositivos estão condicionadas somente a interpretar valores de tensão analógica entre 0 e 5V com uma corrente máxima de 40mA, qualquer valor de tensão ou corrente que desobedeça a esses critérios poderá danificar permanentemente a porta utilizada ou em casos mais graves danifica toda a estrutura da placa Arduino.

Enquanto estuda as amostras da onda senoidal o primeiro arduino interpreta a frequência dessa onda através de técnicas e cálculos escritos em seu algoritmo interno. Tal frequência ao ser encontrada é transmitida ao segundo Arduino através do modulo transmissor conectado a portal digital número quatro. Essa porta leva os dados da frequência lida naquele momento até o transmissor que se encarrega de transmitir a informação ao receptor a espera delas. A todo o momento esse processo se repete, assim qualquer mudança na frequência dessa onda será detectada imediatamente e enviada ao receptor dessas informações.

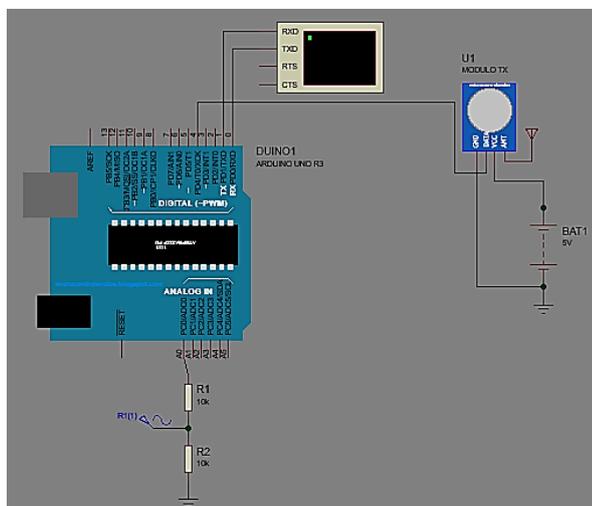


Fig. 3. Circuito do primeiro Arduino.

Ao receber essas frequências, o modulo receptor conectado ao segundo arduino (Fig.2), repassa essas informações a placa controladora através da porta digital número sete. A esse segundo Arduino é incumbido o trabalho de receber a frequência encontrada no primeiro e repassa-la ao microcomputador conectado a ele pela porta *USB* de ambos os dispositivos. Logo fica evidente que o segundo Arduino nesse projeto nada mais é do que uma unidade retransmissora de informações, pois seu único trabalho é coletar os dados transmitidos e envia-los pela porta serial *COM* até o microcomputador que estará conectado a ela.

Para coletar e trabalhar as frequências recebidas na porta serial, o microcomputador utiliza o software *MATLAB*®. Esse software integra análise numérica, cálculo com matrizes, processamento de sinais e construção de gráficos em ambiente fácil de usar, ao contrário da programação tradicional. Utilizado as ferramentas disponíveis nesse software, programou-se um algoritmo que faz a leitura das frequências enviadas à porta serial e amostra numericamente uma onda senoidal idêntica a gerada no gerador de sinais conectado ao primeiro Arduino. Cada frequência recebida é amostrada em frações de tempo durante um período no valor de um segundo. Assim esse algoritmo armazena essas amostras – durante um intervalo de tempo determinado – e cria um sinal senoidal com as diversas frequências coletadas, sendo ela igual ou não.

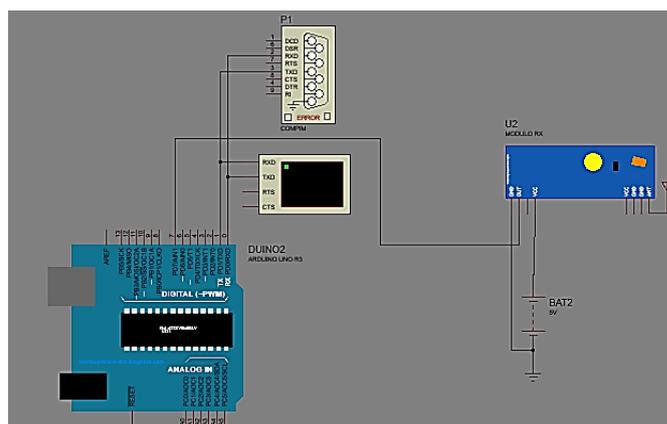


Fig. 4. Circuito do segundo Arduino.

Todas essas frequências têm que ser amostradas a uma taxa de pelo menos 1000 Hz, pois a frequência máxima que é lida nesse projeto é de quase 500 Hz, de acordo com o Teorema de Nyquist. Em vista disso e levando-se em conta que após ser construído esse sinal é reproduzido pelos alto falantes do microcomputador, tal sinal é gerado a uma taxa de amostragem igual a utilizada em sistemas baseados em *Compact Disc Audio (CD)*, ou seja é amostrado em 44100 Hz. A amostragem ocorre a essa taxa pois esse sistema leva em conta a frequência máxima que o ouvido humano pode captar, que é de cerca de 20000 Hz.

Os dois Arduino Uno R3 e microcomputador utilizado possuem um sistema de controle, análise e tratamento dos dados que passam por eles. Esses sistemas são elaborados em algoritmos transcritos na linguagem de programação C, no caso do *MATLAB*, ele também possui um sistema de script baseado nessa linguagem de programação.

O algoritmo do primeiro Arduino que é o arduino acoplado ao gerador de sinais e ao modulo transmissor, tem em suas primeiras quinze linhas de código (da linha de número 1 á 15) a declaração da biblioteca de comunicação dos módulos RF, a *VirtualWire.h* que é uma biblioteca de disponibiliza várias

funções úteis que facilitam a utilização dos módulos RF, além disso também são declaradas várias variáveis e matrizes inteiras, de ponto flutuante e strings. No meio dessas, as mais importantes a se destacar são o vetor que armazena as amostras de tensão (em número de bits) da porta analógica e o vetor de mesma dimensão deste que armazena o instante no tempo (em micros segundos) em que essa amostra é coletada, também é importante se destacar as variáveis que armazenam a frequência instantânea, o histórico de frequências e média dessas.

Das linhas 18 a 29 do algoritmo são definidas primeiramente a referência da porta analógica, isso é necessário porque o Arduino lê as tensões de entrada e as converte a um valor de amplitude entre 0 e 1023 que equivale a 0 e 5 volts respectivamente, logo no nosso caso a referência foi 5V, ou seja a padrão desse modelo de Arduino. Também são estabelecidas nessas linhas de código a velocidade de comunicação da porta serial que foi utilizada para monitorar os dados coletados, a resolução a partir da nossa referência, o pino utilizado na comunicação com o módulo e a velocidade dessa comunicação.

Dentro do loop onde acontece a repetição das instruções contínuas do Arduino, da linha 31 a 39 estão dispostas as instruções que coletam as amostras de tensão e tempo da onda senoidal que está sendo lida. Enquanto que nas linhas seguintes (40 a 63) existem controle de fluxos que tem como objetivo identificar entre as amostras coletadas em que células estão armazenadas dois máximos do sinal lido.

Sabendo onde se encontra os dois picos da onda senoidal, os códigos das linhas restantes (de 64 a 83) calculam a frequência da onda ao encontrar o período entre os dois picos coletados. Assim os códigos seguintes a esses cálculos se responsabilizam por converter a frequência que, é um número inteiro, para uma string que é um vetor de caracteres. Esse vetor de caracteres é então enviado através do módulo transmissor para o receptor utilizando as funções de comunicação disponíveis na biblioteca VirtualWire.h.

No algoritmo do segundo Arduino que está ligado ao módulo receptor são declaradas a biblioteca de comunicação já citada, duas variáveis que irão receber a frequência vinda do módulo receptor em formato de string que é então convertida em inteiro. Em seguida são estabelecidas algumas referências de comunicação na função setup, para então finalmente no loop desse código ser feitas a recepção em string da frequência transmitida, sua conversão para número inteiro e sua retransmissão para o microcomputador ligado a esse arduino por meio da porta serial USB desses dispositivos.

Através do software MATLAB, o algoritmo escrito na aba de scripts deste, estabelece uma conexão com a porta serial para onde o segundo arduino está enviando as frequências lidas, todas as configurações dos parâmetros necessários a essa comunicação são realizadas através das funções escritas nas primeiras linhas de código desse algoritmo (da linha 1 a 36).

Por fim, o algoritmo estabelece algumas variáveis de ficam responsáveis por conter alguns parâmetros do sinal que será reconstituído, como amplitude, taxa de amostragem e tempo, além de matrizes que são utilizadas para armazenar as amostras do sinal reconstruído. Os laços de repetição que se encontram nessas linhas de código fazem a leitura da frequência na porta serial do microcomputador e através dela amostram um sinal senoidal com essa frequência de modo esse se assemelhar com o recebido na porta analógica do primeiro arduino. Essa amostragem acontece até se encontrar o valor unitário de tempo, isso significa que para cada frequência lida na porta serial, o algoritmo irá gerar amostras a equivalentes um segundo de tempo, assim em sessenta segundos temos uma frequência para cada segundo.

Tendo terminado esse processo de amostragem o algoritmo pega as amostras, de amplitude e tempo da onda senoidal estimada, e as converte em um sinal sonoro que tem como saída os alto falantes do microcomputador. Esse sinal sonoro é armazenado em um arquivo de áudio dentro do diretório do algoritmo, também é realizada a plotagem gráfica desse sinal sonora em uma janela de trabalho do software.

III. RESULTADOS E DISCUSSÕES

Utilizando o software Matlab criamos uma senóide com a frequência recebida pelo receptor Rx, e reproduzimos o som obtido no próprio software. O sinal de onda obtido pode ser visualizado na Fig. 5.

E o valor de frequência simulado recebidos no receptor pode ser observado na Fig. 6.

Com base nos resultados obtidos verificamos que as frequências recebidas estão com os valores tendendo a se aproximar da frequência da senóide aplicada na entrada.

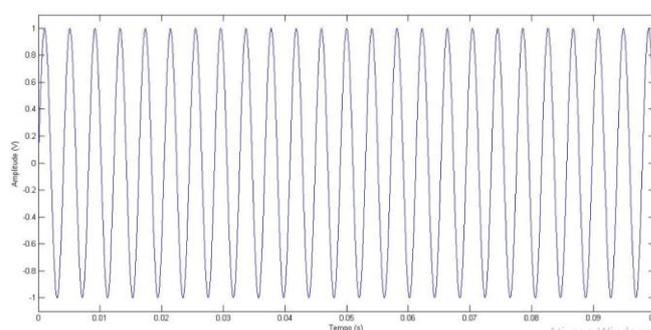


Fig. 5. Sinal senoidal reconstruído no Matlab com a frequência recebida

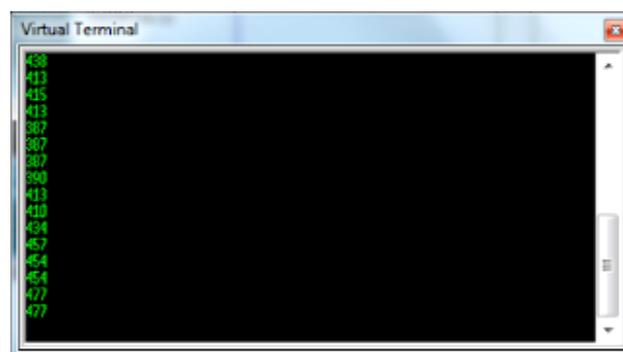


Fig. 6. Frequências recebidas no receptor

No entanto, como os valores recebidos no receptor são valores de amostras ora variam os valores com pequenos erros ora mantem-se fixos. Esses erros na frequência recebida teve influência de vários fatores desde o processo de conversão analógica digital até a transmissão e recepção. Uma das principais fontes de erro foi a metodologia utilizada para a obtenção da frequência onde foi realizada a leitura de vários pontos aplicada e utilizados da onda aplicada e utilizando a definição de que o pico da onda é o ponto de encontro da parte crescente e decrescente de onda senoidal.

Tendo como base esta definição podemos encontrar o valor aproximado do pico da onda e para cada leitura de pico guardamos na memória o tempo deste ponto e pico. Entretanto, uma das desvantagens desse método é o erro das aproximações que ocasionam erros na frequência obtida onde observamos flutuações nos valores que em alguns instantes são erros consideráveis.

Uma alternativa para diminuir esse erro seria usar mais amostras no processador de leitura da tensão de entrada o que diminuiria a distância entre os valores lidos e aumentaria a precisão dos valores de frequência recebidos.

Outra observação é sobre a taxa de amostragem na rotina no MATLAB onde de acordo com as frequências do sinal recebido esse valor pode variar de acordo com o Teorema de Nyquist eu afirma que a frequência de amostragem deve ser maior do que a frequência do sinal no mínimo duas vezes maior, pois isso influenciará no sinal reconstruído e na qualidade do mesmo. Considerando o acima exposto realizamos uma mudança na frequência da onda de entrada e na taxa de amostragem do sinal no MATLAB o que melhorou a qualidade sonora do sinal reproduzido.

CONSIDERAÇÕES FINAIS

Este artigo teve como objetivo apresentar os resultados obtidos a partir da análise de aplicação dos módulos RF 433 MHz, os sinais transmitidos e capturados de uma comunicação via rádio frequência. Para a realização das análises foram necessários os estudos do sinal de rádio frequência, como a sua forma de modulação e alcance, onde foi encontrado como resultado através de equipamentos de medição o comportamento adequado da forma de onda transmitida.

As aferições dos valores obtidos com os equipamentos e softwares estão de acordo com a modulação simples ASK/OOK. A utilização dos módulos para a transmissão e recepção no dispositivo mostrou um funcionamento adequado e condizente com a teoria de modulação ASK, e essa afirmação pode ser comprovada uma vez que todo sinal transmitido é recebido, apesar que as métricas utilizadas através do visualizador da IDE do arduino e do MATLAB mostrarem uma pequena variação nos valores transmitidos e recebidos no entanto a forma de onda não houve variação.

Considerando a análise dos resultados conclui-se que o projeto ocorreu de forma satisfatória contribuindo como exercício real de aplicação da comunicação e modulação digital.

REFERÊNCIAS

- [1] R. Arthur, "Sistema de Aquisição, Armazenamento, Tratamento e Visualização de Sinais Neurológicos de Cobaias em Experimentos de "Spreading Depression", unpublished.
- [2] Yoshinori Kuno, Saturo Nakanishi, Teruhisa Murashima, Nobutaka Shimada, and Y. Shirai, "'Robotic Wheelchair with Three Control Modes.'" proceedings of the 1999 IEEE International Conference on Robotics & Automation., pp. 2590-2595, May 1999.
- [3] F. Ahmed, S. Alim, S. Islam, K. Bhusan and S. Islam "433 MHz (Wireless RF) Communication between Two Arduino UNO", American Journal of Engineering Research (AJER), Vol-5, Issue-10, pp-358-362.
- [4] J. Boxball "Arduino Workshop: A Hands-On Introduction with 65 Projects" 1st edition, 2013 (AJER), Vol-5, Issue-10, pp-358-362.