



Medidor de Pressão Arterial Digital

Edmilson dos Santos, Joni Ramos, Wilian França Costa

¹Universidade Presbiteriana Mackenzie (UPM)
Rua da Consolação, 930 Consolação, São Paulo - SP, 01302-907 – Brazil

10919010291@mackenzista.com.br, 10919005636@mackenzista.com.br,
156768@mackenzie

Abstract. Blood pressure measurement is very important to assess the health conditions and risks that many people suffer from hypertension. The relevance of arterial hypertension (AH) as an important cardiovascular risk factor (CVRF), its high worldwide prevalence and the increased probability of fatal or non-fatal circulatory outcomes when other risk factors are associated with it, make knowledge of its national and regional occurrence, as well as the correlation with other possible factors potentially triggering cardiocirculatory events, as presented in the article “Arterial hypertension and some risk factors in a Brazilian capital” [1]. Hypertension patients require blood pressure to be measured continuously and regularly. One of the ways to measure blood pressure and check the health conditions of patients, would be to develop a portable device and that the data read would be sent, in real time to a repository, where the doctor could have access to the averages and statistics of these daily readings.

Resumo. A medição da pressão arterial é muito importante para avaliar as condições de saúde e os riscos que muitas pessoas sofrem com a hipertensão. A relevância da hipertensão arterial (HA) como importante fator de risco cardiovascular (FRCV), sua alta prevalência mundial e o aumento da probabilidade de desfechos circulatórios fatais ou não-fatais quando a ela estão associados outros fatores de risco tornam muito importante o conhecimento de sua ocorrência nacional e regional, assim como a correlação com outros possíveis fatores potencialmente desencadeantes de eventos cardiocirculatórios, conforme apresentado no artigo “**Hipertensão arterial e alguns fatores de risco em uma capital brasileira**” [1]. Pacientes com hipertensão requerem a medição da pressão sanguínea de forma contínua e regular. Uma das formas de medirmos a pressão arterial e verificar as condições de saúde dos pacientes, seria o desenvolvimento de um aparelho portátil e que os dados lidos fossem encaminhados, em tempo real a um repositório, onde o médico poderia terá acesso as médias e estatísticas destas leituras diárias.

1. Introdução

A hipertensão é uma causa de morte ou doença grave [2]. A hipertensão é uma doença assassina silenciosa ou chamada de "assassina silenciosa" porque a causa original é desconhecida ou não apresenta sintomas. A hipertensão pode causar várias complicações que podem levar a várias outras doenças, como doenças cardíacas, cegueira, rins e acidente vascular cerebral até a morte. De acordo com RISKESDAS (Pesquisa Básica de Saúde) em 2010 a hipertensão pode atacar qualquer pessoa, várias faixas etárias também nível socioeconômico. A doença hipertensiva na população brasileira com 18 anos ou mais é de

23,9% da população total, segundo o IBGE em sua pesquisa de 2020. Pessoas que sofrem de hipertensão devem fazer medições regulares da pressão arterial, porque a pressão arterial é um sinal vital dos humanos que pode mudar com o tempo. A medição da pressão arterial pode ser usada como referência para os pacientes que procuram ajuda médica para ajudar em sua condição. Existem alguns microcontroladores que podem ser usados em sistemas que são [3] ATMega, [4] Arduino, ARM, etc. Nesta pesquisa usamos o Arduino. A medição da pressão arterial pode ser feita em qualquer lugar. Por ter um medidor de pressão arterial próprio, os pacientes hipertensos podem medir sua pressão arterial e saber quando encontrar profissionais de saúde. Nesta pesquisa, tentamos projetar e fazer um sistema de monitoramento digital de pressão arterial que possa ser usado de forma fácil e eficiente. [5] Nesta pesquisa utilizamos o protocolo MQTT para transferir dados da pressão arterial automática para um repositório na nuvem, para salvar a medição de dados.

2. Materiais e Métodos

Um dos problemas mais comuns de saúde no Brasil correlaciona-se, de alguma forma, com a Pressão Arterial. Os valores normais da PA de uma pessoa adulta são de 120 mmHg para a Pressão Arterial Sistólica (PAS) e 80 mmHg para a Pressão Arterial Diastólica (PAD). Valores superiores são associados à hipertensão e valores inferiores à hipotensão. O maior perigo, e que atinge cerca de 24% da população adulta no Brasil é a hipertensão, Pressão Arterial Elevada, que é um fator de risco para doença cardíaca coronária, isquêmica, acidente vascular cerebral (AVC) e outras patologias.

A Pressão Arterial pode ser medida por procedimentos invasivos e não-invasivos. O procedimento invasivo é mais preciso, porém, apresenta-se de forma inconveniente, aumenta o risco de infecções e proporciona maior custo com produtos descartáveis. No procedimento não-invasivo, por outro lado, encontramos os equipamentos que se baseiam em métodos de controle do fluxo sanguíneo por intermédio de um manguito. O mais comum deles é conhecido como Esgomomanômetro e é composto de um manguito, uma pêra valvulada e um manômetro. O Esgomomanômetro é ilustrado na Figura 1.



Figura 1 – Esgomomanômetro
Fonte: Imagem Internet

Os Esgomomanômetros Digitais são regidos pela norma do Inmetro/Dimel nº 96/2008 ao qual especifica as exigências mínimas necessárias para sua utilização [6]. Apesar da

desconfiança de alguns profissionais sobre o modelo digital, este apresenta recursos adicionais que não são encontrados em modelos analógicos.

Neste contexto é desejável a construção de um Medidor de Pressão Arterial Digital de baixo custo, baseado na tecnologia da Internet das Coisas (IoT), com recurso de monitoramento em tempo real via internet, para auxiliar os profissionais de saúde a atender no Monitoramento Remoto de Pacientes (MRP) [7], em especial aqueles que possuem mobilidade reduzida ou dificuldade de deslocamento até a unidade de saúde.

O nosso projeto consiste em realizar leituras em um transdutor como sensor de pressão, detectar seus pulsos fazendo a transdução do sinal de pressão para uma tensão analógica, e posteriormente digitalizá-los por meio de um conversor analógico-digital (ADC).

Também alteraremos o medidor de pressão original (Esfigmomanômetro), substituindo a Pêra Valvulada por uma bomba de pressão de ar e uma válvula eletrônica.

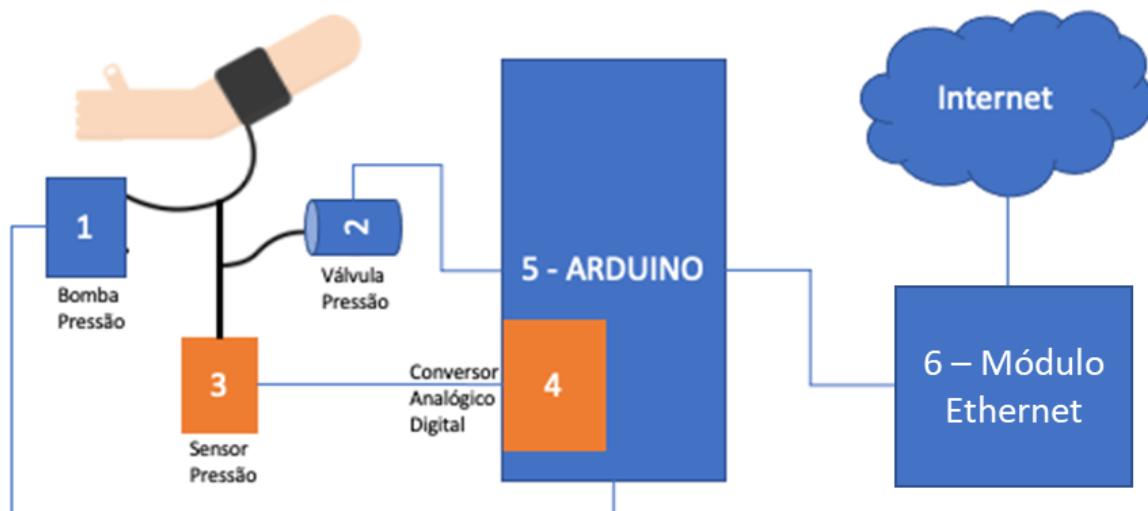


Figura 2 – Diagrama de Blocos para o medidor de Pressão Arterial Digital
Fonte: Autoria Própria

Neste projeto estaremos utilizando os seguintes materiais:

- [5] Blackboard UNO R3, contendo:
 - [4] Microcontrolador: ATmega328P SMD
 - 8 entradas analógicas
 - 6 saídas PWM
- [3] Sensor de Pressão modelo 2050 da linha MPX
- [1] Bomba de Pressão de Ar KPM14A
- [2] Válvula Eletrônica KSV05A
- [6] Módulo Ethernet ENC28J60 para conexão com a Internet

Método para Amostragem e Cálculo da Pressão Arterial

Para que seja possível detectar corretamente as métricas PAS e PAD, é necessário preservar as características dos pulsos oscilométricos durante a digitalização. Os métodos clássicos para calcular estas métricas baseiam-se na amplitude de cada pico, que se comportam como componente de alta frequência.

Apesar da informação estar concentrada nas faixas compreendidas entre 3 e 40 Hz, costuma-se adotar uma frequência de amostragem muito maior que a frequência de amostragem tradicional para sinais biológicos, usualmente 10 vezes a frequência de amostragem.

No ATmega328p iremos trabalhar com uma frequência de amostragem de 893 amostras por segundo.

Após a aquisição de um pequeno número de amostras a janela será processada extraíndo apenas os parâmetros essenciais para determinar as métricas da PA. As informações de maior relevância extraídas da janela são: diferença pico-a-pico dos pulsos oscilométricos e o valor da pressão do manguito associado a cada pulso. O tamanho da janela determinado experimentalmente será de 180 amostras e a ordem para o processamento da janela a cada 5 novas amostras.

Após a identificação do pulso é realizado uma nova aquisição de amostra da pressão imediata do manguito, associando-a ao pulso encontrado. Em virtude do atraso proveniente do processamento e otimização de memória a localização central do pulso fica comprometida causando uma pequena defasagem em função da real pressão associada ao manguito. O valor médio desta defasagem, em relação ao centro do pulso, foi de 20ms, ao qual não acarreta grandes diferenças no cálculo da Pressão Arterial.

Após termos os dados, enviaremos ao módulo ESP8266, o qual se encarregará de enviar os dados, via WiFi, para um broker de MQTT na internet.

Detalhamento dos Componentes

Arduino Uno R3

O Uno [4] difere de todas as placas antecessoras no sentido de não utilizar o chip FTDI para conversão do sinal serial. Utiliza no seu lugar um Atmega8U2 programado como conversor de USB para serial.

Revisão 3 da placa com as novas características:

- Pinos SDA e SCL adicionados próximos ao AREF.
- Dois outros pinos adicionados próximos ao RESET, o IOREF que permite aos shields se adaptarem à voltagem fornecida pela placa. No futuro os shields serão compatíveis tanto com as placas que utilizam o AVR e operam a 5V, como com o Arduino Due que operará a 3,3V. O segundo pino não está conectado e é reservado para propósitos futuros.
- Circuito de RESET mais robusto.
- Atmega 16U2 em substituição ao 8U2.

Características:

Microcontrolador	ATmega328
Voltagem Operacional	5V
Voltagem de entrada (recomendada)	7-12V
Voltagem de entrada (limites)	6-20V
Pinos E/S digitais	14 (dos quais 6 podem ser saídas PWM)
Pinos de entrada analógica	6

Corrente CC por pino E/S	40 mA
Corrente CC para o pino 3,3V	50 mA
Flash Memory	32 KB (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Velocidade de Clock	16 MHz

Alimentação

O Arduino uno pode ser alimentado pela conexão USB ou com uma fonte de alimentação externa. A alimentação é selecionada automaticamente.

Alimentação externa (não USB) pode ser tanto de um adaptador CA para CC ou bateria. Há um conector para alimentação de 2,1mm com o positivo no centro. Cabos vindos de uma bateria podem ser inseridos diretamente nos pinos Gnd e Vin do conector de alimentação.

Esta placa pode funcionar com uma fonte de alimentação externa de 6 a 20 volts. No entanto se a alimentação for inferior a 7V, o pino 5V pode fornecer menos de cinco volts e a placa pode se mostrar instável. E se a alimentação for maior do que 12V o regulador de voltagem pode superaquecer e danificar a placa. A faixa recomendada é de 7 a 12 volts.

Os pinos de alimentação são os seguintes:

- VIN. A entrada de alimentação para a placa Arduino quando se está utilizando uma fonte de alimentação externa. (em oposição à conexão USB ou outra fonte de alimentação regulada). Você pode fornecer alimentação através deste pino, ou se estiver utilizando o conector de alimentação acessar esta voltagem aqui.
- 5V. A fonte de alimentação regulada usada para o microcontrolador e para outros componentes na placa. Pode vir tanto do VIN através do regulador embarcado ou da conexão USB ou outra fonte regulada em 5V.
- 3V3. Uma fonte de 3,3V gerada pelo regulador embarcado. A corrente máxima suportada é de 50mA.
- GND. Pinos terra.

Memória

O ATmega328 tem 32KB (dos quais 0,5 são utilizados pelo bootloader). Também tem 2KB de SRAM e 1KB de EEPROM.

Entrada e Saída

Cada um dos 14 pinos digitais do Uno pode ser utilizado como uma entrada ou uma saída utilizando-se as funções pinMode(), digitalWrite(), e digitalRead(). Eles operam a 5V. Cada pino pode fornecer ou receber um máximo de 40mA e tem um resistor pull-up interno (desconectado por padrão) de 20-50kΩ. Além disso alguns pinos tem funções especializadas:

- Serial: 0 (RX) e 1 (TX). Usados para receber (RX) e transmitir (TX) dados seriais TTL. Estes pinos são conectados aos pinos correspondentes do chip serial USB-para-TL ATmega8U2.
- Interruptores Externos: 2 e 3. Estes pinos podem ser configurados para disparar uma interrupção de acordo com alguma variação sensível pelo circuito.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estes pinos dão suporte à comunicação SPI.
- LED: 13. Há um LED integrado ao pino digital 13. Quando este pino está no valor HIGH este LED está aceso, quando o pino está em LOW o LED está apagado.
- I²C: 4 (SDA) and 5 (SCL). Fornecem suporte a comunicação I²C (TWI).
- AREF. Voltagem de referência para as entradas analógicas.
- Reset. Envie o valor LOW para esta linha para resetar o microcontrolador. Tipicamente usado para adicionar um botão de reset para shileds montados sobre a placa original.

O Uno tem 6 entradas analógicas, etiquetadas de A0 a A5, cada uma tem 10 bits de resolução (i.e. 1024 valores diferentes). Por padrão elas medem de 0 a 5V, embora seja possível alterar o limite superior utilizando o pino AREF e a função analogReference().

Comunicação

O Arduino Uno possui uma série de facilidades para se comunicar com um computador, outro Arduino, ou outros microcontroladores. O ATmega328 fornece comunicação serial UART TTL (5V) que está disponível nos pinos digitais 0 (RX) e 1 (TX). Um ATmega8U2 na placa canaliza esta comunicação para a USB e aparece como uma porta virtual para o software no computador. O firmware do '8U2 utiliza os drivers padrão USB COM e nenhum driver externo é necessário. Entretanto, no Windows, um arquivo .inf é necessário. Ainda faltam as instruções específicas, mas em breve estarão disponíveis. O software do Arduino inclui um monitor serial que permite dados textuais ser enviados e recebidos da placa. LEDs conectados ao RX e TX piscarão enquanto dados estiverem sido transmitidos pelo chip USB-para-serial e pela conexão USB (mas não para comunicação serial nos pinos 0 e 1).

Uma biblioteca de Software Serial permite comunicação serial em qualquer dos pinos digitais do Uno.

O ATmega328 também suporta comunicação I²C (TWI) e SPI. O software do Arduino inclui uma biblioteca Wire para simplificar o uso do bus I²C.

Programação

O Arduino Uno pode ser programado com o software Arduino. Simplesmente selecione "Arduino Uno" no menu Tools > Board.

O ATmega328 no Arduino Uno vem pré-gravado com um bootloader que permite a você enviar código novo para ele sem a utilização de um programador de hardware externo. Ele se comunica utilizando o protocolo original STK500.

Reset automático por software

Ao invés de necessitar do pressionamento físico de um botão antes de um upload, o Arduino Uno é desenvolvido de maneira que permita que esta operação seja feita por meio do software rodando em um computador. Uma das linhas de controle de fluxo do hardware (DTR) do ATmega8U2 é conectado à linha de reset do ATmega328 através de um capacitor de 100nF. Quando esta linha é declarada (rebaixada) a linha de reset cai o suficiente para resetar o chip. O software do Arduino utiliza esta capacidade para permitir o envio de código novo simplesmente pressionando o botão de upload na IDE. Isto significa que o bootloader pode ter um intervalo mais curto, uma vez que o rebaixamento do DTR pode ser mais bem coordenado com o início do upload.

Esta configuração tem outras implicações. Quando o Uno é conectado a um computador rodando Mac OS X ou Linux, ele é resetado cada vez que uma conexão for estabelecida com o software (via USB). Durante o próximo meio segundo o bootloader rodará no Uno. Uma vez que ele está programado para ignorar dados malformados (i.e. qualquer coisa diferente do upload de um novo código), ele irá interceptar os primeiros bytes de informação após a abertura da conexão. Se um programa rodando na placa recebe alguma configuração ou outra informação quando começa a rodar esteja seguro de que o software com o qual ela se comunica espere por um segundo antes de começar a enviar dados.

O Uno contém uma trilha que pode ser interrompida (cortada fisicamente) para desabilitar o auto-reset. Os conectores de cada lado da trilha podem ser soldados para reabilitar esta função. Ela está identificada como "RESET-EN". Você também pode desabilitar o auto-reset conectando um resistor de 110Ω do 5V à linha de reset.

Proteção contra sobre-corrente na USB

O Arduino Uno possui um polyfuse resetável que protege a porta USB do seu computador contra sobre-corrente e curtos-circuitos. Embora muitos computadores tenham sua própria proteção interna, o fuse fornece uma camada a mais de proteção. Se mais de 500mA forem aplicados a porta USB ele automaticamente irá interromper a conexão até que o curto ou a sobrecarga seja removido.

Características físicas

A largura e o comprimento máximos do PCB do Uno são 68,58 e 53,34mm respectivamente, com os conectores USB e de alimentação extendendo-se além destas dimensões. Quatro orifícios para parafusos permitem que a placa seja fixada a uma superfície ou encapsulamento. Verifique que a distância entre os pinos digitais 7 e 8 é de 160mil (milésimos de polegada ou 0,16"), não é nem mesmo um múltiplo dos 100mil que separam os outros pinos.

Sensor de Pressão modelo 2050 da linha MPX

Fabricante: NXP [8]

Categoria de produto: Sensores de pressão de suporte de quadro

RoHS: Detalhes

Tipo de pressão: Differential

Pressão operacional: 0 kPa to 50 kPa

Estilo de montagem: Through Hole
Tensão de alimentação operacional: 10 V
Tipo de porta: Dual Radial Barbed
Caixa / Gabinete: Case 344C-01
Temperatura operacional mínima: - 40 C
Temperatura operacional máxima: + 125 C
Série: MPX2050
Embalagem: Tray
Marca: NXP Semiconductors
Tamanho da porta: 4.93 mm
Tipo de Produto: Pressure Sensors
Quantidade do pacote de fábrica: 100
Subcategoria: Sensors
Aliases de núm de peça: 935322701117
Peso unitário: 5.058 g

Bomba de Pressão de Ar KPM14A

Mini bomba de Ar/Vácuo
Tamanho da bomba: 40mm * 90mm * 35mm
Diâmetro da tomada, diâmetro 6mm, diâmetro exterior 9mm
Tensão de funcionamento: DC6-12v (recomendado)
Corrente de trabalho: 0.5-0.7-a
Tráfego: 1.5 - 2 l/min
Maior sucção: 2 m
Elevação: máximo de 3 metros.

Válvula Eletrônica KSV05A

Válvula solenoide 2 vias é indicada para inúmeras aplicações. A função deste tipo de válvula é controlar a passagem de fluidos, abrindo e fechando, por meio de uma bobina elétrica. Disponibilizamos esta válvula na condição normal fechada, ou seja, ao acionar o solenoide, ela é aberta para a passagem do fluido.

Dados Técnicos:

Fluido: ar comprimido, água e óleos leves
Função: 2/2 vias normal fechada (NF)
Vazão: 230 L/min
Orifício de passagem: 2.5 mm
Rosca de entrada e saída: 1/4 BSP
Pressão máxima de entrada: 10 Bar
Pressão de trabalho: 0 ... 8,0 Bar
Temperatura de trabalho: -5 ... 60°C
Material: corpo - alumínio / Bobina e conector elétrico: plástico
Vedaçāo: Buna-N

Módulo Ethernet ENC28J60 para conexão com a Internet

Este é um driver que utiliza o novo chip da Microchip ENC28J60 – Controlador de Internet IC que suporta vários protocolos da internet que são requeridos. Este módulo conecta diretamente com a maioria dos microcontroladores com uma interface SPI cuja transferência ultrapassa 20MHz.

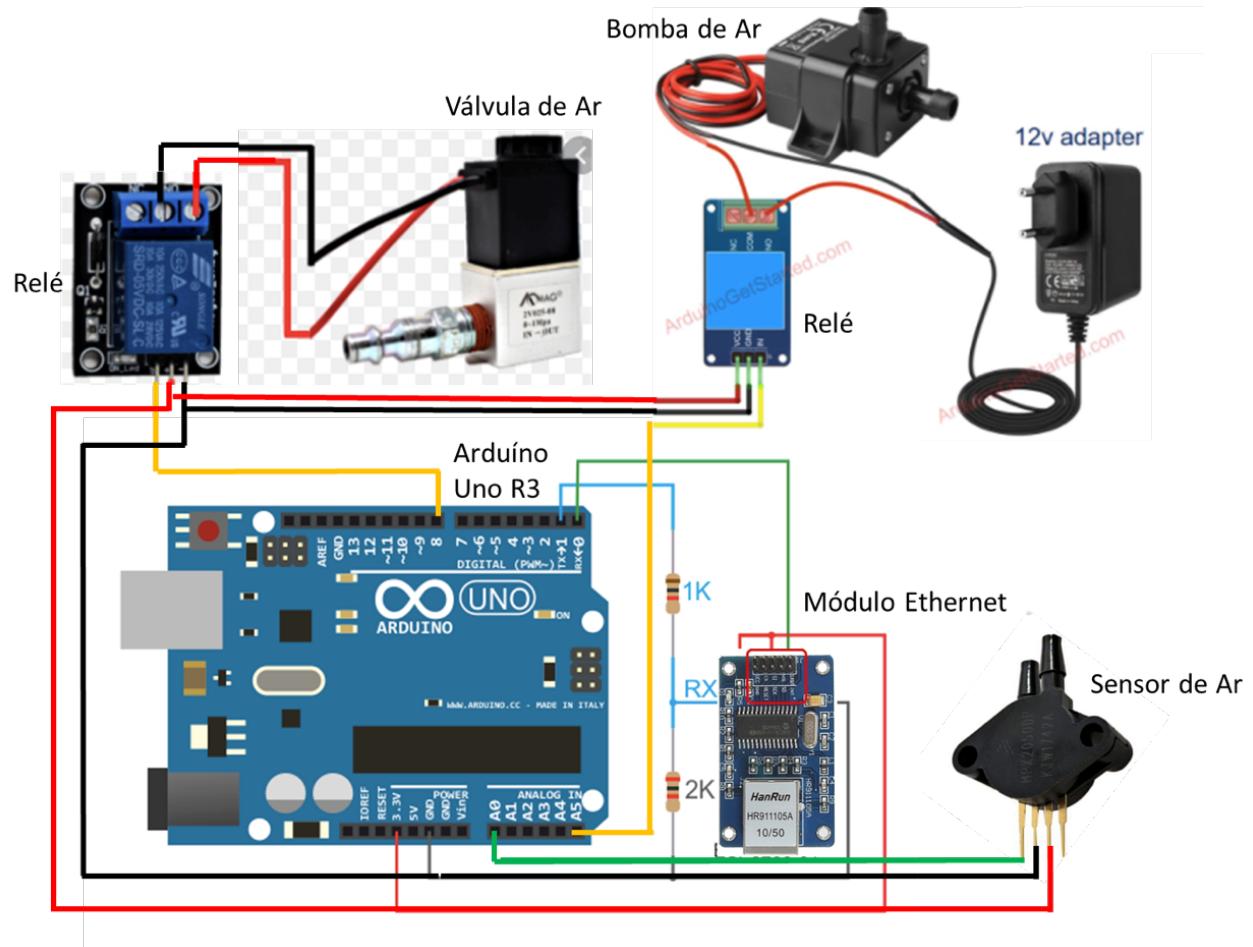


Figura 3 – Diagrama de Conexões para o medidor de Pressão Arterial Digital
Fonte: Autoria Própria

Integração entre o Arduino e o servidor MQTT

O padrão de troca de mensagens no MQTT é o publish/subscriber (publicador/subscritor). Neste padrão, quando um elemento da rede deseja receber uma determinada informação, ele se subscreve, fazendo uma requisição para um outro elemento da rede capaz de gerir as publicações e subscritões. Na rede MQTT este elemento é conhecido como broker, o intermediário no processo de comunicação. Elementos que desejam publicar informações o fazem também através do *broker*, enviando-lhe as informações que possuem. Esse padrão não é novo e existe em outros protocolos. Por exemplo, a troca de informação de controle (links) em redes Foundation Fieldbus segue o paradigma publish/subscriber.

A conexão do cliente ao *broker*, seja ele subscritor ou publicador, é originalmente feita via TCP, com opções de login (usuário e senha) e uso de criptografia (SSL/TLS). É possível encontrar também outros meios físicos, com MQTT rodando em links seriais, por exemplo.

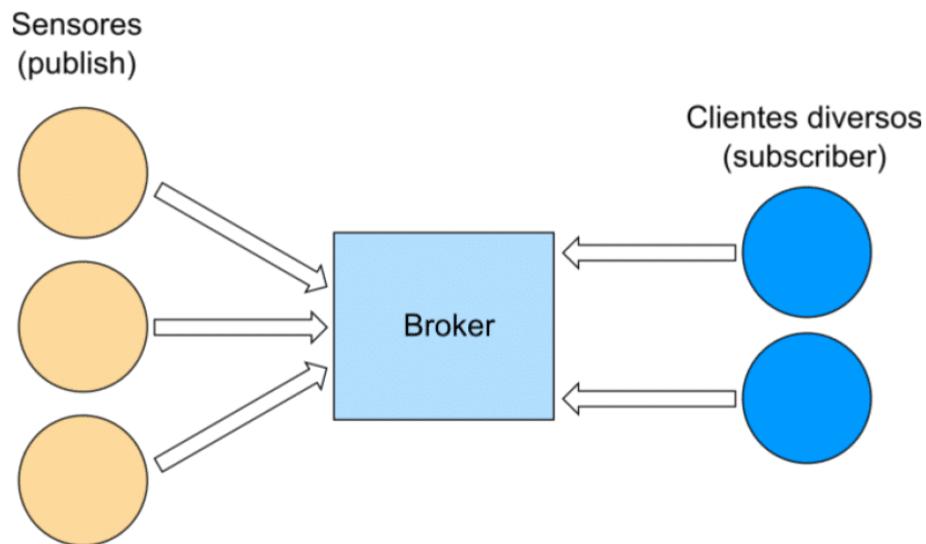


Figura 4 – Diagrama de Conexões MQTT – Protocolos para IoT

Fonte: Autoria Própria

3. } Resultados

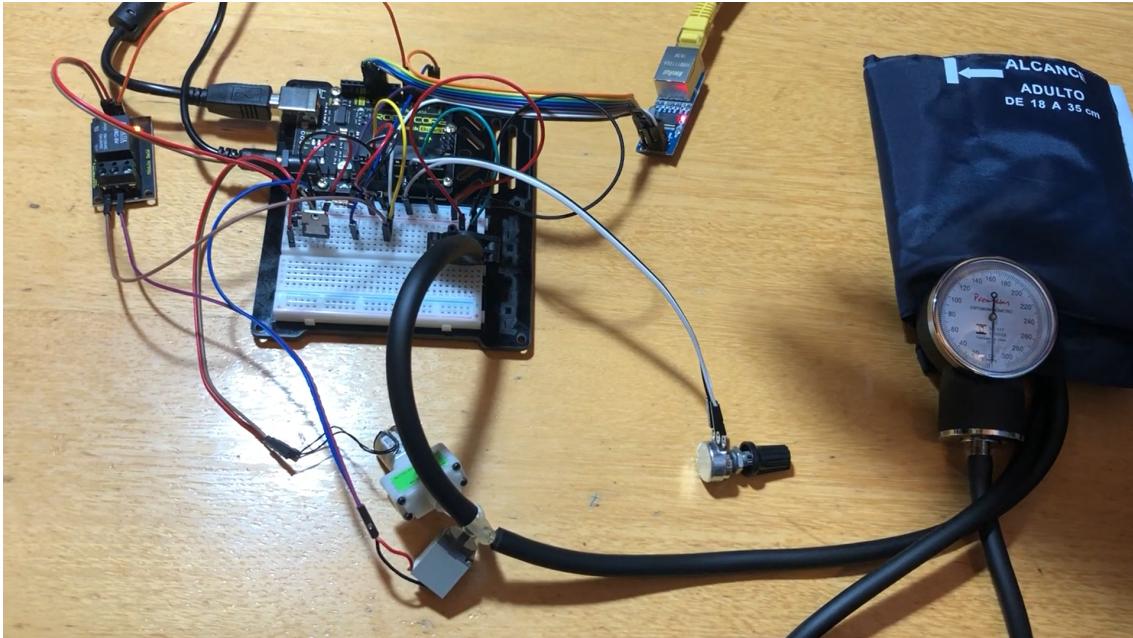


Figura 5 – Imagem de nossa montagem em funcionamento

Fonte: Autoria Própria

```
Publish message success
20:42:49.644 -> Topic: Joni_ED_Topic
20:42:49.644 -> Message: Hello 1234
20:42:49.691 -> Publish message success
20:42:53.674 -> Publish message success
20:42:57.705 -> Topic: Joni_ED_Topic
20:42:57.705 -> Message: Hello 1234
20:42:57.705 -> Publish message success
20:43:01.694 -> Publish message success
20:43:05.725 -> Topic: Joni_ED_Topic
20:43:05.771 -> Message: Hello 1234
20:43:05.771 -> Publish message success
20:43:09.742 -> Publish message success
20:43:13.819 -> Topic: Joni_ED_Topic
20:43:13.819 -> Message: Hello 1234
20:43:13.866 -> Publish message success
20:43:17.849 -> Publish message success
20:43:21.880 -> Topic: Joni_ED_Topic
20:43:21.880 -> Message: Hello 1234
20:43:21.880 -> Publish message success
20:43:25.863 -> Publish message success
20:43:29.893 -> Topic: Joni_ED_Topic
20:43:29.940 -> Message: Hello 1234
20:43:29.940 -> Publish message success
20:43:33.910 -> Publish message success
20:43:37.974 -> Topic: Joni_ED_Topic
20:43:37.974 -> Message: Hello 1234
20:43:38.021 -> Publish message success
```

Figura 6 – Imagem do fluxo de mensagens do protocolo MQTT

Fonte: Autoria Própria

Link para a demonstração em vídeo no YouTube:

<https://www.youtube.com/watch?v=d5JsXOMCsaw>

Repositório do Projeto:

<https://github.com/edpaes07/MedidorPressaoArduino>

Quais foram os achados encontrados em função da proposta do projeto?

Configuramos o Arduino Uno com o código Firmata, para estabelecer a comunicação MQTT, objetivo do projeto. Adicionamos a bomba de ar, conectado a um circuito regulador de corrente e tensão, bem como uma válvula acionada por um relé, para ir liberando o ar armazenado, para que o sensor de pressão possa monitorar os batimentos cardíacos na corrente sanguínea e determinar o nível da pressão sistólica e diastólica.

Infelizmente em nosso projeto, as leituras do sensor de pressão demonstraram um problema, pois o sensor envia variações de 40 mV, que não podemos detectar diretamente pelo Arduíno Uno. Assim teremos que adicionar ao nosso circuito um amplificador operacional e realizar melhorias e novos testes para finalização do projeto.

4. Conclusões

Como conclusão preliminar, atingimos parcialmente os objetivos, pois todos os sensores e acionadores foram utilizados e colocados em funcionamento no projeto, mas temos que elencar que enfrentamos um problema na leitura do sensor, e precisamos adicionar um amplificador operacional para obtermos as leituras precisas do sensor.

5. Referências

[1] Paulo César B. Veiga Jardim; Maria do Rosário Peixoto Gondim; Estelamaris Tronco Monego; Humberto Graner Moreira; Priscila Valverde de Oliveira Vitorino; Weimar Kunz Sebba Barroso Souza; Luiz César Nazário Scala, Arq. Bras. Cardiol. vol.88 no.4 São Paulo Apr. 2007

https://www.scielo.br/scielo.php?pid=S0066-782X2007000400015&script=sci_arttext

- [2] Hypertension Canada, “Hypertension Canada 2017 Guidelines for management of Hypertension,” *Hypertens. Canada*, 2017.
- [3] I. B. A. I. Iswara and I. G. M. N. Desnanjaya, “Trainer ATMEGA32 Sebagai Media Pelatihan Mikrokontroler dan Arduino,” *Resistor*, vol. 1, no. 1, pp. 55–64, 2018.
- [4] M. Margolis, *Arduino Cookbook*, First Edit. O'Reilly, 2011.
- [5] <https://mqtt.org/>
- [6] Serviço Público Federal, Portaria Inmetro n.º 096, de 20 de março de 2008. INSTITUTO NACIONAL DE METROLOGIA, NORMALIZAÇÃO E QUALIDADE INDUSTRIAL, Prédio da Rua Santa Alexandrina, Rua: Santa Alexandrina, 416, Rio Comprido - Rio de Janeiro - RJ. CEP: 20261-232, 96 ed., mar 2008. Disponível em <http://www.inmetro.gov.br/legislacao/rtac/pdf/RTAC001287.pdf>
- [7] E. Prado, “Monitoramento remoto de pacientes reduz custos,” may 2015. Disponível em <https://www.saudebusiness.com/hospital/monitoramento-remoto-de-pacientes-reduz-custos>
- [8] NXP Freescale Semiconductor <https://www.nxp.com/>