



FACULDADE SENAI DE TECNOLOGIA MECATRÔNICA

MONITORAMENTO DE EQUIPAMENTOS COM COMUNICAÇÃO BLUETOOTH E WIFI

BLUETOOTH AND WIFI COMMUNICATION EQUIPMENT MONITORING

Patrick Silva da Conceição ^{1, i}
 Paulo Sebastião Ladivez ^{2, ii}
 Daniel Otavio Tambasco Bruno ^{3, i}
 Vicente Gomes de Oliveira Junior^{4, ii}

RESUMO

Este trabalho tem como objetivo apresentar as facilidades e particularidades da comunicação *Bluetooth* e *WiFi*, seus periféricos, recursos para o desenvolvimento de aplicativos para *smartphone* e aplicativos, que oferecem soluções e recursos de acordo com as necessidades do projeto desenvolvido. Dentre os temas de trabalhos de conclusão de curso (TCC), apresentados por alunos de cursos técnicos por conta das facilidades apresentadas pela *Internet of Things* (IoT), muitos apresentam ideias que abordam monitoramento com *smartphone* e sites na internet, que mostram informações coletadas por algum sistema eletrônico com a capacidade de conexão com a internet como a placa nodeMCU ESP 32.

ABSTRACT

This work aims to present the facilities and particularities of Bluetooth and WiFi communication, its peripherals, resources for the development of applications for smart phone and applications that offer solutions and resources according to the needs of the developed project. Among the topics of Course Completion Work (CCW) presented by students of technical courses because of the facilities presented by Internet of Things (IoT), many present ideas that approach smartphone monitoring and internet sites that show information collected by some electronic system with the ability to connect to the internet such as the board nodeMCU ESP 32.

¹ Pós-graduando em Automação Industrial na Faculdade SENAI de Tecnologia Mecatrônica. E-mail: patrick7432@hotmail.com

² Professor da Faculdade SENAI de Tecnologia Mecatrônica. E-mail: paulo.ladivez@sp.senai.br

³ Mestre em Engenharia da Informação. Professor da Faculdade SENAI de Tecnologia Mecatrônica. E-mail: daniel.bruno@sp.senai.br

⁴ Docente na Faculdade SENAI de Tecnologia Mecatrônica. Mestre em Engenharia Mecânica. E-mail: vgomes@sp.senai.br

1 INTRODUÇÃO

Com o avanço tecnológico cada vez mais rápido, dinâmico e colaborativo, é possível observar facilidades sendo implementadas em ambientes residenciais, comerciais e industriais. Como exemplo, existe a indústria 4.0 e a IoT. É possível encontrar um vasto conteúdo abordando esses avanços e comodidades oferecidas em sites de empresas que criam aplicações específicas para suas necessidades, visando a redução de custo e rapidez no processo de fabricação, validação e distribuição. Como também é possível encontrar recursos e aplicações voltadas para necessidades mais simples, que possam ser adaptadas para uma residência.

Para exemplificar, é possível encontrar diversos projetos sobre automação residencial, onde o usuário controla a iluminação da sua casa e monitora algumas câmeras, através do seu *smartphone* utilizando uma comunicação via *bluetooth* ou WiFi.

A partir desta necessidade de comunicação via *bluetooth* ou WiFi, encontramos alguns recursos como a placa de *bluetooth* HC-05, que deve ser conectada em algum microcontrolador como o PIC ou uma placa de desenvolvimento microcontrolada, como o Arduino. Caso a necessidade seja uma comunicação via WiFi, é possível encontrar algumas placas de desenvolvimento microcontrolada como o ESP 8266 12E ou o ESP32. Estas placas podem ser programas para o acionamento de terminais como um microcontrolador e possui a possibilidade de conexão com o WiFi.

E para a conseguir ler todas as informações geradas em um celular, por essa quantidade de recursos, se faz necessário um aplicativo que pode ser adquirido em loja de apps ou desenvolvido através de algumas ferramentas disponíveis na internet como o appinventor.

Para exemplificar todos os recursos, será desenvolvido neste trabalho o monitoramento de um equipamento através do *smartphone*, utilizando a comunicação *bluetooth* e WiFi abordando a programação da placa de desenvolvimento microcontrolada e da criação do aplicativo responsável pela leitura dessas informações.

2 DESENVOLVIMENTO

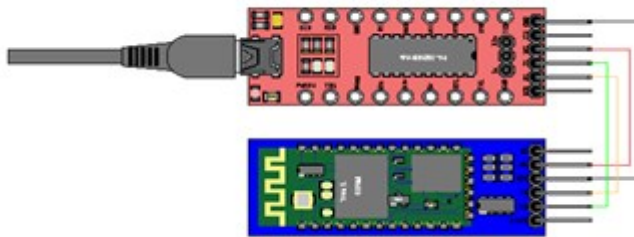
Para o desenvolvimento de um projeto capaz de monitorar e enviar informações para um *smartphone*, é necessário entender o funcionamento dos recursos de transmissão via *bluetooth* e WiFi, assim como os aplicativos de leitura.

2.1 Bluetooth

A transmissão via *bluetooth* ocorrerá através do circuito chamado HC-05. Sua recepção e transmissão de informações acontece de forma serial pelos terminais TX e RX. O HC-05 possui alguns parâmetros de configurações que podem ser modificados através dos comandos AT.

Para se configurar o HC-05 foi necessário utilizar o conversor USB para RS232 TTL, na figura 1. É importante ressaltar que os terminais TX e RX do HC-05 trabalham com uma tensão de 3,3V e o conversor utilizado possui um *jumper*, que permite a atenuação de 5V para 3,3V, compatibilizando os dados da porta USB.

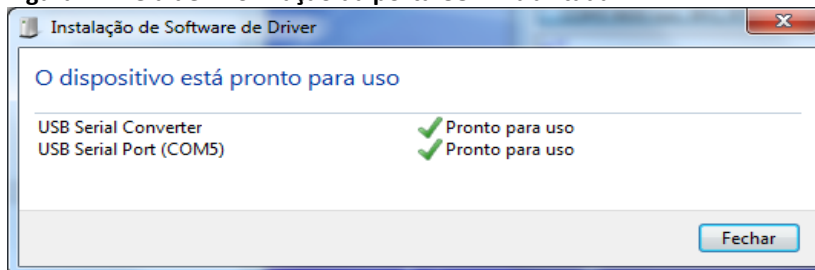
Figura 1 – Ligação do HC-05 ao FTDI



Fonte: Elaborado pelo autor

Após conectar o conversor na porta USB do *personal computer* (PC), o Windows fará a instalação do *driver* como apresenta a figura 2, informando a porta COM, porta serial de comunicação utilizada.

Figura 2 – Tela de informação da porta COM habilitada



Fonte: Elaborado pelo autor

Após a ligação como apresentado na figura 1, é necessário configurar o programa onde serão inseridos os comandos de configuração. O programa usado é o TERMITE 3.1, figura 3.

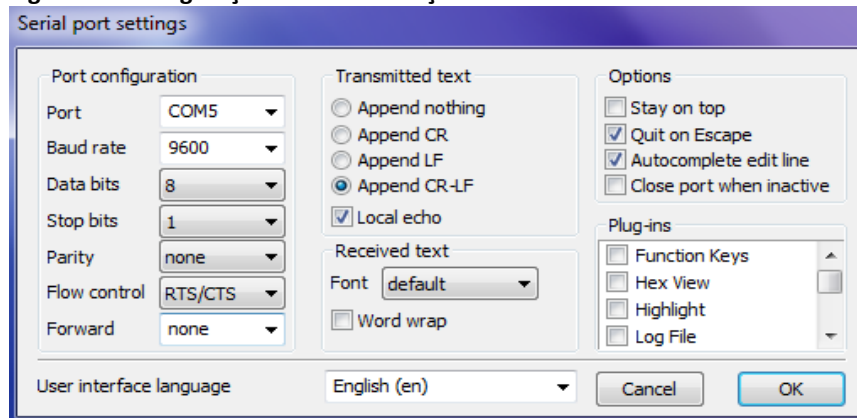
Figura 3 – Tela de comando do TERMITE 3.1



Fonte: Elaborado pelo autor

O programa possui um botão chamado *Settings*, que apresenta uma tela de configurações editáveis, que são necessárias para estabelecer comunicação com o HC-05. A figura 4 apresenta essa tela com as configurações alteradas.

Figura 4 – Configurações de comunicação do TERMITE 3.1



Fonte: Elaborado pelo autor

Ao finalizar e confirmar as configurações, é possível iniciar a conexão e transmitir os comandos AT.

Os comandos AT (atenção) ou comandos Hayes foram concebidos na década de 80 por Dennis Hayes para estabelecer comunicação entre modems.

A lista de parâmetros básicos utilizados para o desenvolvimento do sistema de monitoramento são:

AT (ENTER) = comando para testar a comunicação entre pc e a placa. Caso ocorra a comunicação, o HC – 05 retornará a mensagem Ok. Caso contrário ele retornará a msn ERRO ou não retornará nenhuma mensagem.

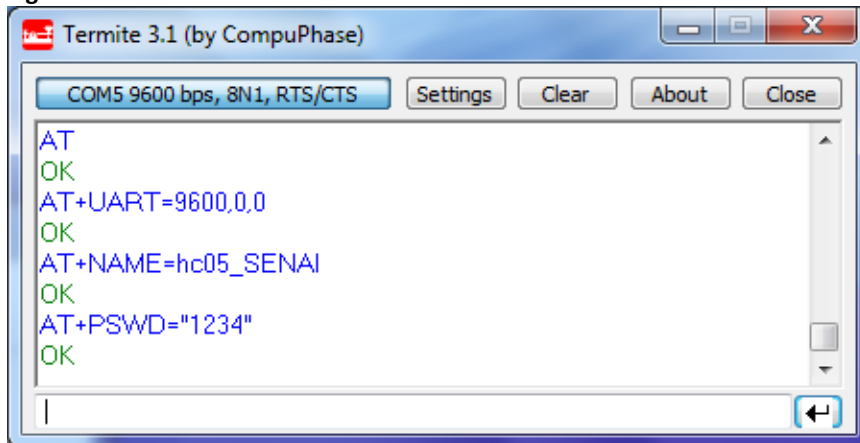
AT +UART=xxxx (ENTER) = comando que configura a taxa de Baud Rate para a comunicação serial. O valor de baud rate parametrizado foi de 9600, portanto, o comando foi AT+UART=9600 e como confirmação o pc recebe a mensagem ok.

AT + NAME=nomebluetooth (ENTER) = comando que configura o nome da rede bluetooth. O nome utilizado foi hc05_SENAI e para inserir essa informação, o comando inserido foi AT + NAME=hc05_SENAI. Como confirmação do comando, a mensagem de retorno será OK.

AT + PSWD="xxxx"(ENTER) = comando que configura uma senha de autorização para que o smartphone se conecte ao HC – 05. A senha usada é 1234 e o comando usado para inserir essa configuração foi AT + PSWD="1234" (ENTER) e como retorno de confirmação a mensagem é OK.

Na figura 5 temos o programa TERMITE 3.1 se comunicando com o HC-05 através dos comandos AT.

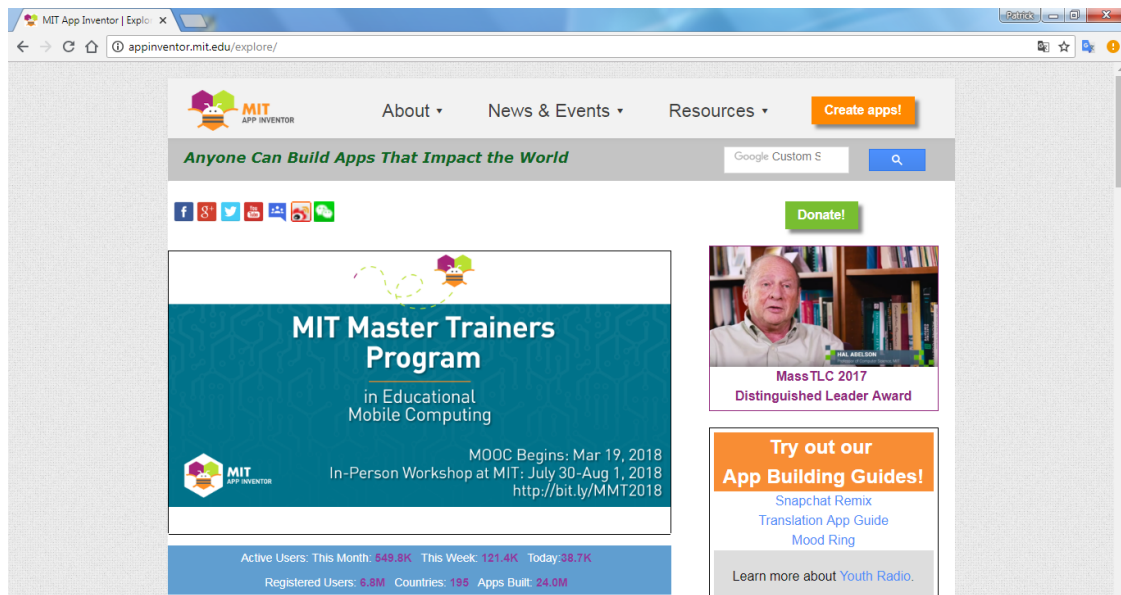
Figura 5 – Inserindo os comandos AT no TERMITE 3.1



Fonte: Elaborado pelo autor

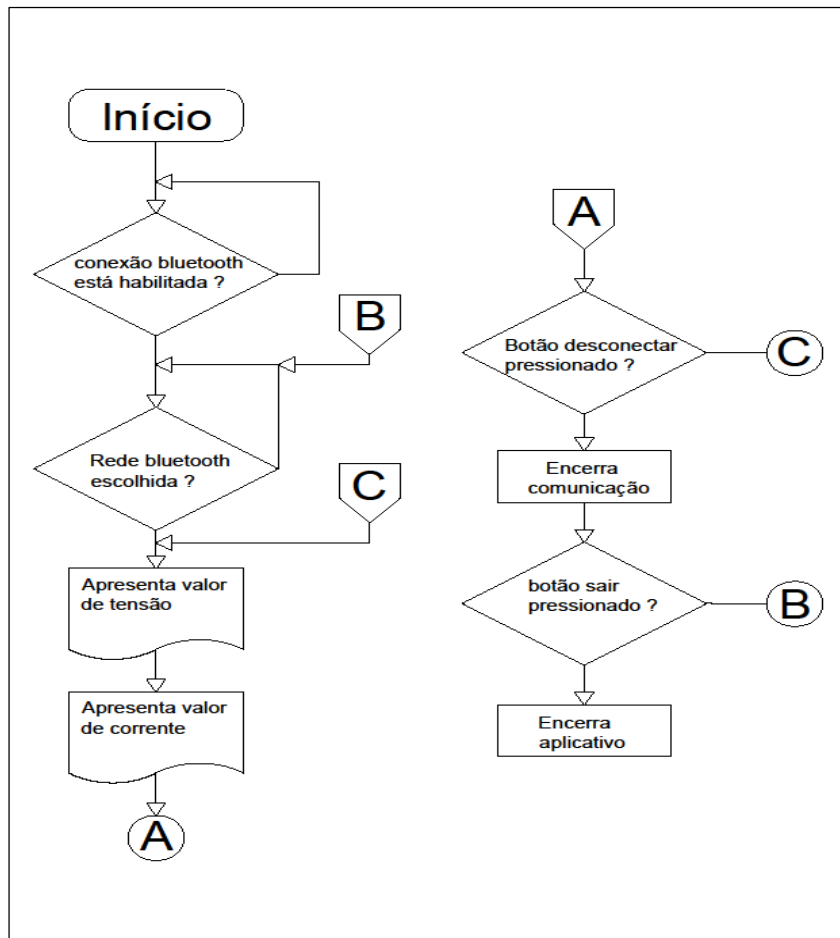
Após as configurações básicas do HC-05, é necessário desenvolver o aplicativo do *smartphone*, que efetuará a leitura das informações transmitidas pelo *bluetooth*. Em lojas de aplicativos é possível encontrar diversos aplicativos, que permitem essa transmissão e recepção de informações via *bluetooth* além de promover pequenas alterações editáveis pelo usuário. O HC-05 irá se comunicar com um aplicativo, que foi desenvolvido através da plataforma online chamada appinventor, que é possível ser acessada pelo site <<http://appinventor.mit.edu/explore/>>, figura 6.

Figura 6 – Página inicial do appinventor



Fonte: Elaborado pelo autor

A lógica de funcionamento do aplicativo é evidenciado pelo fluxograma da figura 7.
Figura 7 – Fluxograma

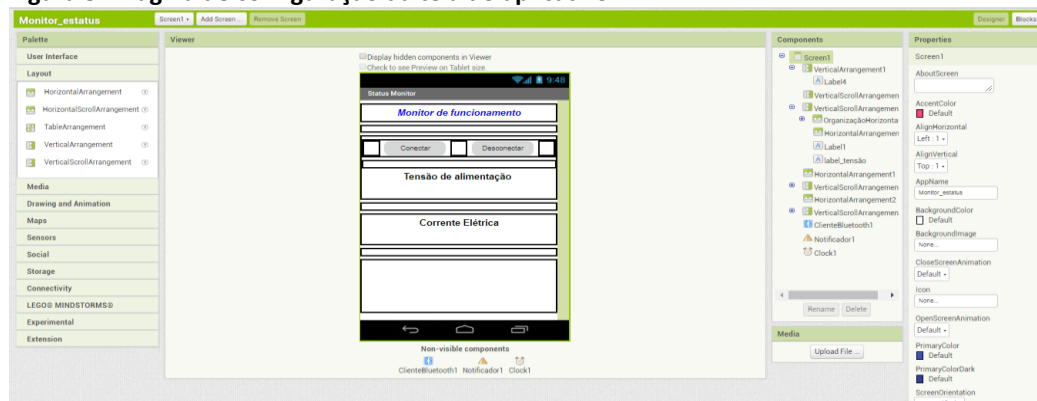


Fonte: Elaborado pelo autor

Para iniciar o processo de desenvolvimento no appinventor, é necessário ter uma conta de email do gmail. Logo após é necessário criar um projeto novo para iniciar a criação do aplicativo.

O ambiente de desenvolvimento do aplicativo é dividido em duas partes onde a primeira é destinada a criação da interface gráfica com recursos para inserir elementos que interajam com o usuário. A segunda parte é destinada a programação do aplicativo através de blocos lógicos.

Figura 8 – Página de configuração da tela do aplicativo



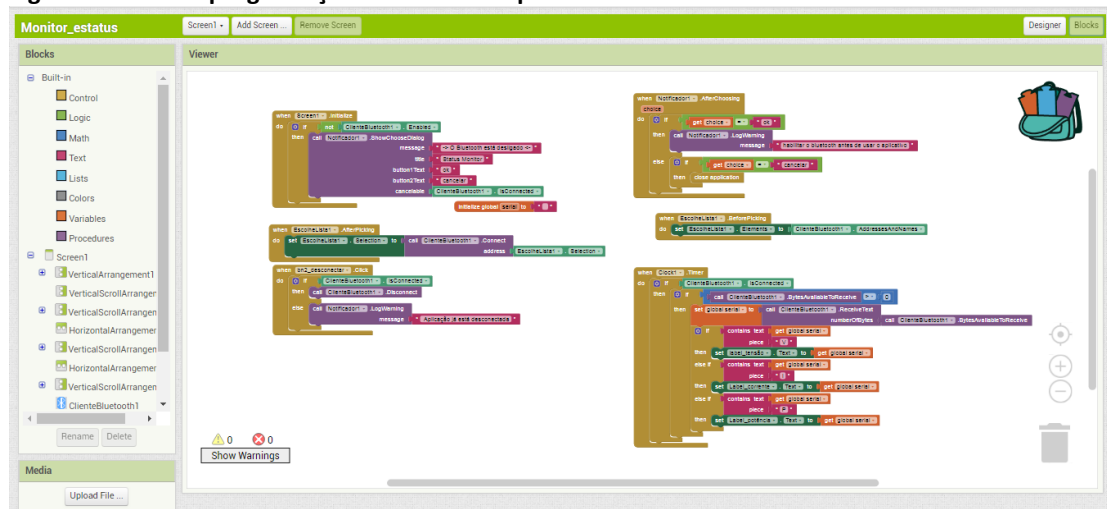
Fonte: Elaborado pelo autor

O aplicativo que irá efetuar a leitura dos dados enviados por *bluetooth* terá uma interface como apresentada na figura 8.

O aplicativo possui dois botões para conectar e desconectar ao HC-05, um espaço destinado a apresentar o valor de tensão e um espaço destinado a apresentar o valor de corrente. Na figura 8 é possível ver retângulos com linhas pretas, que servem para dar espaços entre os itens da tela.

Após o desenvolvimento da interface com o usuário, a próxima etapa é acessar a tela de programação para inserir os blocos responsáveis pelo funcionamento do aplicativo. Na figura 9 é apresentada a tela com os blocos já inseridos.

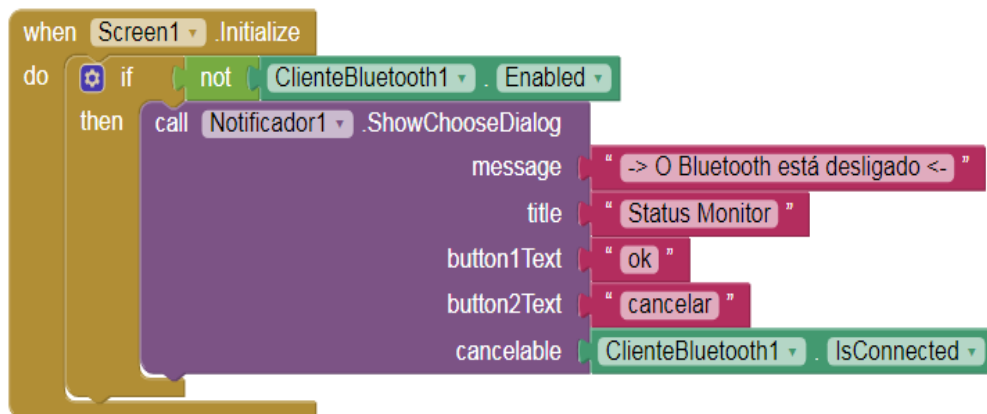
Figura 9 – Tela de programação em bloco do aplicativo



Fonte: Elaborado pelo autor

O primeiro conjunto de blocos inserido é apresentado na figura 10 onde o objetivo é verificar se o *bluetooth* do *smartphone* está habilitado logo que o aplicativo é aberto pelo usuário.

Figura 10 – Bloco de programa inicializa o *bluetooth*

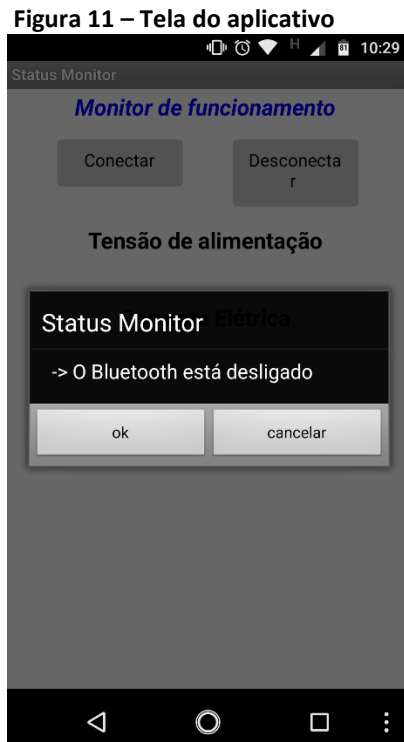


Fonte: Elaborado pelo autor

A lógica aplicada faz a seguinte verificação:

Quando a SCREEN1 for inicializada (aplicativo iniciado), o bloco fará uma condição IF analisando se o ClienteBluetooth1 não está habilitado. Caso essa afirmação seja verdadeira, o bloco CALL Notificador1.ShowChooseDialog será executado fazendo aparecer uma tela de mensagem informando que o *bluetooth* está desabilitado e oferece um ok para que o usuário

habilite e o botão cancelar para que o aplicativo seja encerrado. A figura 11 é a tela que o aplicativo apresenta como resultado do bloco descrito.



Fonte: Elaborado pelo autor

O segundo conjunto de bloco está vinculado ao botão conectar do aplicativo. Para o bloco foi dado o nome de EscolheLista1.

Este bloco irá apresentar uma tela com a lista dos periféricos com *bluetooth* disponíveis para serem conectados ao aplicativo.

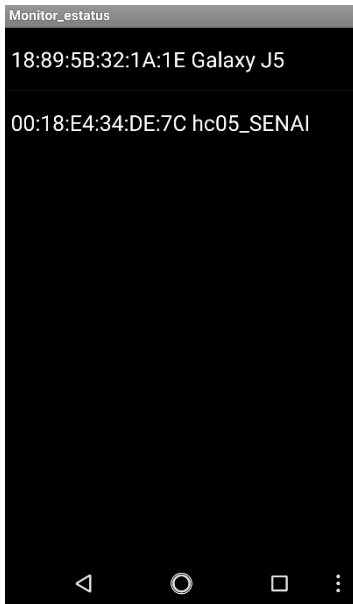
Quando (When) o botão conectar (EscolheLista1) for pressionado, o recurso ClienteBluetooth1.connected será evocado e apresentará os dispositivos bluetooth. Porém para selecionar o periférico desejado é necessário o bloco de programa da figura 12.

Figura 12 – Bloco de programa que habilita lista de conexões



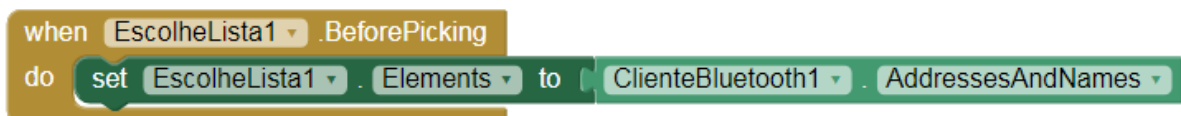
Fonte: Elaborado pelo autor

Figura 13 - Lista de conexões *bluetooths* disponíveis



Fonte: Elaborado pelo autor

Figura 14 – Bloco de programa que permite selecionar e iniciar a conexão

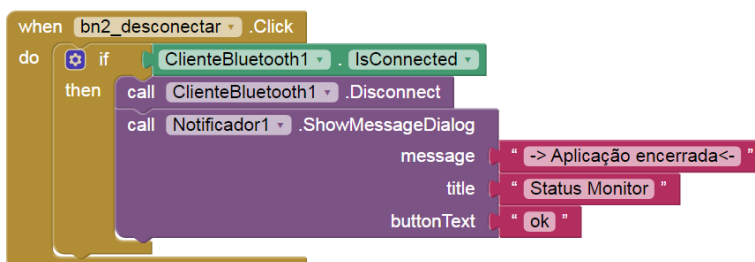


Fonte: Elaborado pelo autor

Através da lista de *bluetooths* na região, a conexão poderá ser estabelecida, conforme figuras 123 e 14

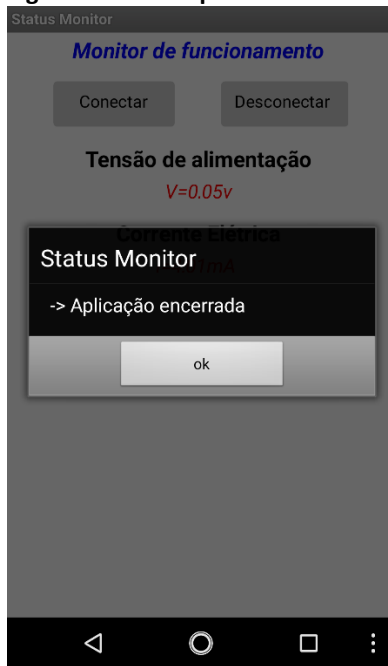
O terceiro bloco é destinado a desconectar *bluetooth*. Quando o botão Desconectar é pressionado, a conexão é interrompida e a mensagem Aplicação encerrada é apresentada junto com um botão de OK, que finaliza a tela. Na figura 15 temos o bloco do programa que executa esta ação e na figura 16 a tela apresentada no aplicativo.

Figura 15 - Bloco de programa que desconecta o bluetooth



Fonte: Elaborado pelo autor

Figura 16 – Tela aplicativo desconectado



Fonte: Elaborado pelo autor

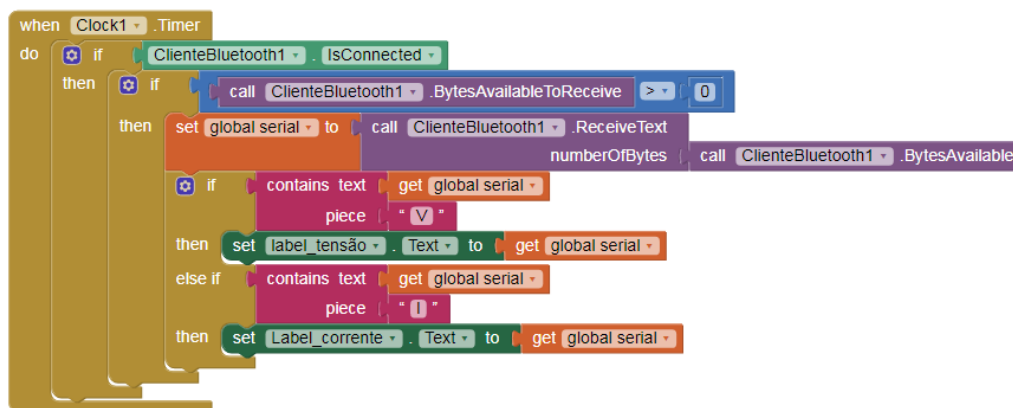
O último bloco de programa é responsável por receber e tratar as informações recebidas pelo hc05_SENAI.

As informações de tensão e corrente devem ser atualizadas constantemente, pois são valores que sofrerão alterações, portanto o aplicativo deve buscar atualizar essas informações de tal forma que seja possível acompanhar em tempo real essas mudanças, figura 17.

Essa atualização se faz possível por conta com recurso chamado CLOCK1.

Esta função irá executar um bloco de forma cíclica a cada 100ms. A primeira verificação desse bloco é uma condição IF (SE), que verifica se existe conexão do aplicativo com o hc05_SENAI. A segunda condição IF verifica se o aplicativo está recebendo por *bluetooth* um valor maior que 0. Após uma verificação positiva, os valores recebidos serão armazenados em uma variável. A terceira condição irá verificar se o valor recebido é uma medição de tensão ou corrente elétrica. Se o valor recebido tiver a letra V, o valor será apresentado na “text_label_tensão” e se o valor recebido tiver a letra I, o valor será apresentado na “text_label_corrente”.

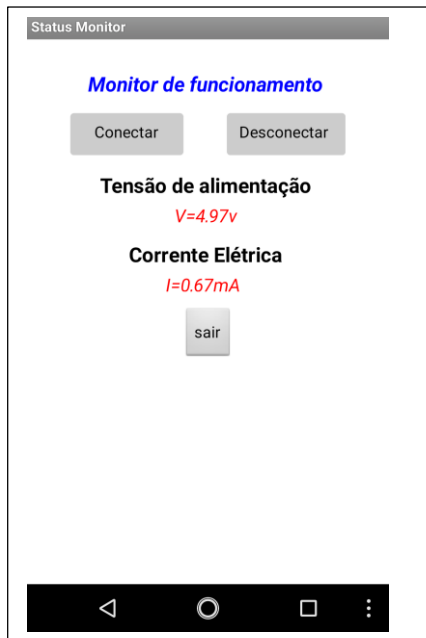
Figura 17 – Bloco de programa que condiciona as informações do HC-05



Fonte: Elaborado pelo autor

A partir das configurações descritas nas figuras anteriores é possível gerar o arquivo .apk e instalar no smartphone como apresenta a figura 18.

Figura 18 -Tela do aplicativo



Fonte: Elaborado pelo autor

A leitura, interpretação e conversão de dados para o formato serial ocorrerá através do modo USART do microcontrolador PIC 18F4550, segundo Miyadaira:

A USART é um circuito eletrônico responsável pela interface entre o dispositivo e a porta serial. Na transmissão, o dispositivo envia os dados para a USART de forma paralela e a USART se encarrega de transmiti-los bit a bit. Pela comunicação serial. Na recepção, a USART realiza a operação contrária, ou seja, recebe a mensagem bit a bit e a entrega de forma paralela para o dispositivo (MIYADAIRA, 2012, p. 209).

O microcontrolador irá receber os dados do sensor de corrente e do sensor de tensão que passarão pelo processo de conversão ADC e pela conversão para caracteres que informem os valores compatíveis com os verificados pelos sensores. logo após é iniciada a transmissão dos valores via UART para o HC-05.

Toda a programação foi elaborada usando o MPLAB 8.5 e com o compilador C18. Como um dos objetivos deste trabalho foi desenvolver a lógica de programação, todo o desenvolvimento lógico no pic foi realizado a partir da configuração dos registradores do ADC e dos registrados do UART. Para a constante atualização dos valores, foi utilizado na programação o recurso de interrupção por TIMER0, segundo Miyadaira:

As interrupções desempenham um papel importante em sistemas microcontrolados. Pois permitem suspender a qualquer momento a execução de uma linha de código e começar a executar uma rotina designada ao tratamento de uma determinada interrupção. Após finalizar a rotina, o processador retorna ao ponto do programa que estava antes de a interrupção ser gerada (MIYADAIRA, 2012, p.195).

A figura 19 apresenta as sub-rotinas executadas no PIC a cada finalização de contagem do TIMER0.

Figura 19 – Subrotinas no pic

```

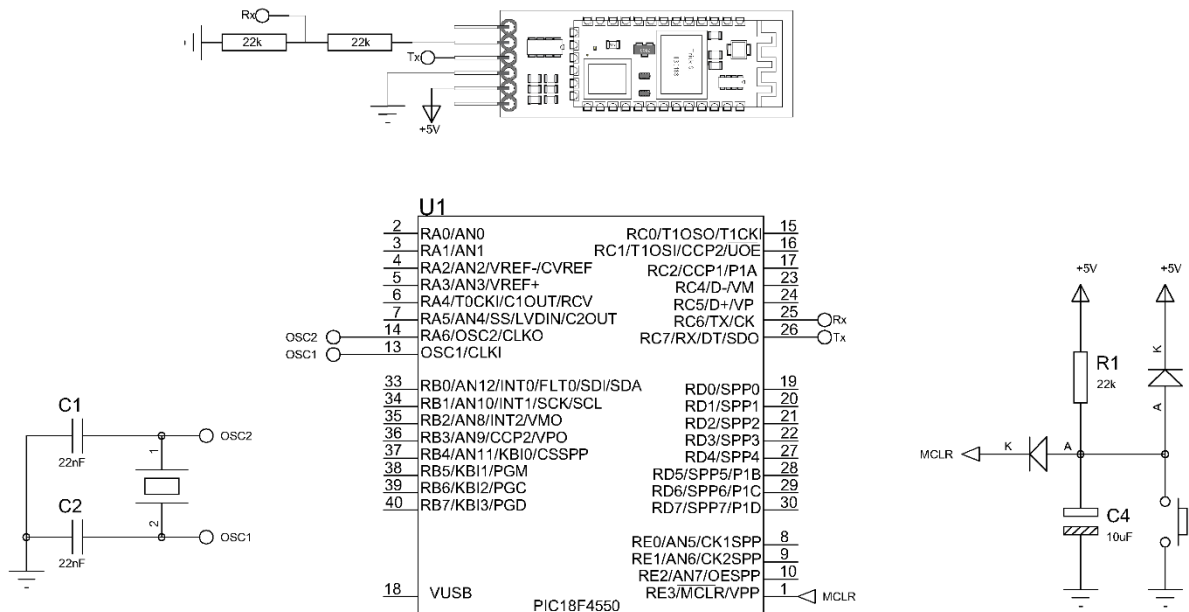
//===Interrupções=====
#pragma code int_alta=0x008
void int_alta(void){
    leitura_TxRx();
}
#pragma code
#pragma interrupt leitura_TxRx

//--- subrotina de interrupção
void leitura_TxRx(void){
    if(INTCONbits.TMR0IF==1) {
        TMR0L = 0xff;
        TMR0H = 0x00;
        ADC_leitura(0); // chama subrotina de leitura do canal 0 do ADC
        volts_AN0(); // faz o tratamento do valor lido no canal 0
        TX_volts(); // transmite o valor lido pelo canal 0 via bluetooth
        temp=0; // atualiza o valor de temp=0
        Delay10KTCYx(50); // atraso de 0.1s
        ADC_leitura(1); // chama subrotina de leitura do canal 1 do ADC
        amper_AN1(); // faz o tratamento do valor lido no canal 1
        TX_amper(); // transmite o valor lido pelo canal 1 via bluetooth
        Delay10KTCYx(100); // atraso de 0.2s
        INTCONbits.TMR0IF=0; // zera a flag de ocorrência de interrupção do TMR0
    }
}
    
```

Fonte: Elaborado pelo autor.

A figura 20 apresenta a ligação do PIC 18f4550 com o HC – 05. Como já informado neste artigo, o terminal RX do HC – 05 tem uma tensão de trabalho de aproximadamente 3,3V e se faz necessário um divisor de tensão no terminal TX do PIC.

Figura 20 – Ligação do pic ao HC 05



Fonte: Elaborado pelo autor

2.2 WiFi

Outra possibilidade de monitoramento de valores pode ocorrer ao acessar a internet com o *smartphone*, o que permite ao usuário ter acesso às informações de qualquer lugar, que apresente um sinal 3G ou 4G.

A transmissão de dados através de internet ocorrerá por meio da placa ESP 32. Para facilitar a montagem deste protótipo, foi usado o ESP 32 nodeMCU, que consiste em uma placa já confeccionada para ser inserido em uma matriz de contatos por pinos de inserção além de possuir uma montagem, que permite a direta conexão com o pc via USB sem a necessidade de atenuar a tensão para 3,3V, já que está placa possui um regulador de tensão.

A figura 22 apresenta o ESP32 e a Figura 23 apresenta o ESP 32 nodeMCU.

Figura 22 - NodeMCU



Fonte: Elaborado pelo autor

Figura 23 – Esp 32



Fonte: espressif

O esp32 pode ser programado utilizando o IDE do ARDUINO o que se mostra como um grande benefício pela facilidade de programação. Em conjunto com a programação desenvolvida no IDE do Arduino, se faz necessário citar o aplicativo utilizado no smartphone, uma vez que ele interfere de forma direta no desenvolvimento e funcionamento do projeto.

O aplicativo utilizado chama-se BLYNK e tem como principal característica apresentar ferramentas gráficas editáveis destinadas a variáveis de entrada e saída tanto para níveis analógicos como digitais.

Ao acessar o site do desenvolvedor do BLYNK e apresentado uma infinidade de recursos e exemplos aplicáveis aos diversos tipos de Arduino e placas de comunicação com WIFI. A figura 24 apresenta a tela onde é possível escolher o tipo de placa, o tipo de conexão, auto token e exemplos de programas associados ao aplicativo BLYNK.

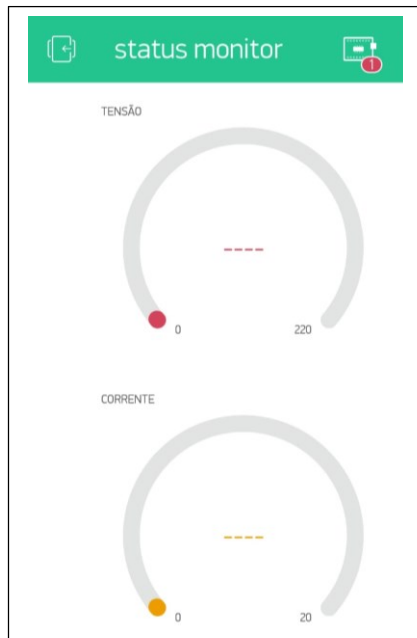
Figura 24 – Página de exemplos



Fonte: examples.blynk.cc

Para configurar a o aplicativo de acordo com a necessidade do projeto, foi elaborado o *layout* apresentado na figura 25.

Figura 25 – Layout do aplicativo pelo BLYNK

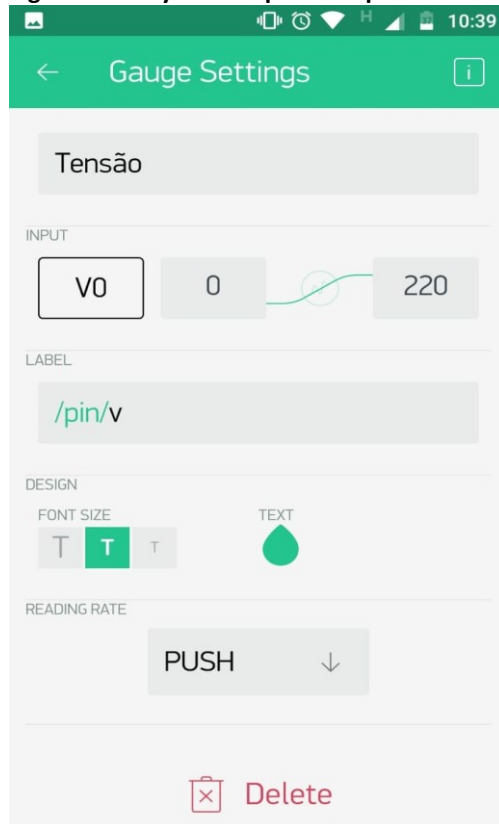


Fonte: Elaborado pelo autor

Os dois recursos destinados para apresentar os valores de tensão e corrente estão atrelados a uma variável de entrada virtual chamada Vx (V1, V2). Essa conexão é indicada

ao lado superior direito de cada mostrador. Esses mostradores são chamados de *gauge*, devem ser configurados no aplicativo como indicado na figura 26.

Figura 26 – Layout do aplicativo pelo BLYNK



Fonte: Elaborado pelo autor

5 CONSIDERAÇÕES FINAIS

O projeto desenvolvido apresenta as facilidades e possibilidades de conexões e transmissão de informações entre dispositivos através do *bluetooth* ou WiFi.

É possível observar a quantidade de informações e recursos para a criação de aplicativos ou soluções já elaboradas, permitindo assim a integração de informações e monitoramento em tempo real de equipamentos de acordo com as configurações estabelecidas pelo programador.

Portanto é possível implementar em ambiente residencial soluções relativamente baratas e confiáveis para verificar o comportamento de equipamentos elétricos. Para ambientes industriais a confecção do projeto deve levar em consideração eventos que possam interferir no funcionamento do projeto.

REFERÊNCIAS

BLYNK. **Example blynk ESP32**. Disponível em: <https://examples.blynk.cc/?board=ESP32&shield=ESP32%20WiFi&example=GettingStarted%2FBlynkBlink>. Acesso em: 03 mar. 2019.

ESPRESSIF SYSTEMS. **ESP32 Series**: Datasheet. Disponível em:
https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.
 Acesso em: 02 fev. 2019.

ESPRESSIF SYSTEMS. **ESP32**: Technical Reference Manual. Disponível em:
https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf . Acesso em: 03 mar. 2019.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY. **MIT App Inventor**. Disponível em:
<http://appinventor.mit.edu/explore/>. Acesso em: 10 jan. 2019.

MICROCHIP TECHNOLOGY. **PIC18F2455/2550/4455/4550 data sheet**. USA, 2006. 430 p.
 Disponível em: <https://ww1.microchip.com/downloads/en/devicedoc/39632c.pdf>. Acesso em: 01 fev. 2019.

MIYADAIRA, Alberto Noboru. **Microcontroladores PIC**: Aprenda e programe em linguagem C. São Paulo: Erica, 2013. 400p. ISBN 978-85-365-0244-1.

AGRADECIMENTOS

A finalização deste artigo só foi possível primeiramente por Deus ter dado a capacidade e paciência investida ao longo do desenvolvimento deste artigo, por conta da ajuda e dedicação do professor orientador Paulo S. Ladivez da Escola SENAI Armando de Arruda Pereira e de vários amigos de trabalho que contribuíram com suas experiências adquiridas ao longo da vida profissional.

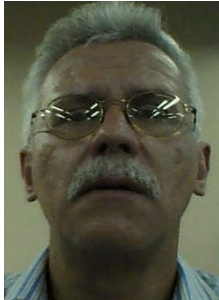
Sobre os autores:

ⁱ PATRICK SILVA DA CONCEIÇÃO



Possui graduação em Tecnologia em Eletrônica Industrial pela Faculdade FAENAC (2010), cursando atualmente a Pós-Graduação em Automação Industrial pela Faculdade SENAI de Tecnologia Mecatrônica (2016). Tem experiência na área de Eletrônica.

ii PAULO SEBASTIÃO LADIVEZ (Orientador)



Possui graduação em Engenharia Elétrica pela Universidade Mogi das Cruzes (1984) com especialização em Tecnologias e Sistemas de Informação pela Universidade Federal do ABC (2013). Atualmente é professor da Faculdade SENAI de Tecnologia Mecatrônica, lecionando as disciplinas Eletrônica, Projetos, Microcontroladores, Controle no Curso Tecnológico em Mecatrônica Industrial e na Pós-Graduação em Automação Industrial. Tem experiência na área de Engenharia Eletrônica, com ênfase em Automação Industrial e Mecatrônica, atuando principalmente nos seguintes temas: Mecatrônica, Manufatura Digital, Redes Industriais, Automação Industrial, Microcontroladores e Controle.

iii DANIEL OTÁVIO TAMBASCO BRUNO (Banca)



Doutorando em Engenharia da Informação na Universidade Federal do ABC (2016), Mestre em Engenharia da Informação pela Universidade Federal do ABC (2013), Especialista em Banco de Dados pela Universidade de Ribeirão Preto (2007) e Especialista em Educação à Distância pela Universidade Paulista (2012). Possui graduação em Análise de Sistemas pela Universidade Paulista (2003). Revisor do Periódico IEEE Transactions on Medical Imaging. Co-editor, editor de área e revisor da Revista Brasileira de Mecatrônica. É Revisor do periódico ForScience do Instituto Federal de Minas Gerais. Atualmente é Técnico em Manufatura Digital e professor da Escola Senai "Eng. Octávio Marcondes Ferraz" e Professor de ensino superior na Universidade Paulista. Tem experiência na área de Inteligência Artificial, Desenvolvimento de Sistemas de Informação, Processamento de imagens, Implantação e Administração de sistemas Product Lifecycle Management (PLM), Segurança de redes e Internet das Coisas.

Endereço para acessar este CV: <http://lattes.cnpq.br/3491851270517427>

iv VICENTE GOMES DE OLIVEIRA JUNIOR (Banca)



Possui graduação em Tecnologia Elétrica pela Universidade Presbiteriana Mackenzie (1982). Complemento em pedagogia na Universidade Metodista de Piracicaba (1999), Mestrado em Engenharia Mecânica pela Universidade Estadual de Campinas (2006). Atualmente é professor na área de automação industrial da Faculdade Senai de Tecnologia Mecatrônica nos cursos de graduação e pós-graduação. Tem experiência na área de Automação Industrial, atuando principalmente nos seguintes temas: pneumática, eletropneumática, hidráulica, eletro hidráulica, controlador programável, robótica básica, sistema supervisorio, algumas redes industriais.