UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I – CAMPINA GRANDE
CENTRO DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIAS DA COMPUTAÇÃO

MAIANARA SALES DO Ó

APLICAÇÃO INICIAL DE ARDUINO UNO COM MÓDULO GSM SIM800L PARA TRANSMISSÃO DE DADOS DE ESTAÇÃO METEOROLÓGICA POR SMS

CAMPINA GRANDE
2023

MAIANARA SALES DO Ó

APLICAÇÃO INICIAL DE ARDUINO UNO COM MÓDULO GSM SIM800L PARA TRANSMISSÃO DE DADOS DE ESTAÇÃO METEOROLÓGICA POR SMS

Trabalho de Conclusão de Curso de Graduação em Ciência da Computação da Universidade Estadual da Paraíba, como requisito à obtenção do título de Bacharel em Ciências da Computação.

Orientador: Prof. Me. Edson Holanda Cavalcante Júnior

CAMPINA GRANDE
2023

O10a Ó, Maianara Sales do.
Aplicação inicial de Arduino Uno com módulo GSM SIM800L para transmissão de dados de estação meteorológica por SMS [manuscrito] / Maianara Sales do Ó. - 2023.
34 p.

Digitado.
Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2024.
"Orientação : Prof. Me. Edson Holanda Cavalcante Júnior, Coordenação do Curso de Computação - CCT. "

1. Arduino. 2. Transmissão de dados. 3. Estação metereológica. I. Título

21. ed. CDD 005.1

Elaborada por Geovani S. de Oliveira - CRB - 15/1009

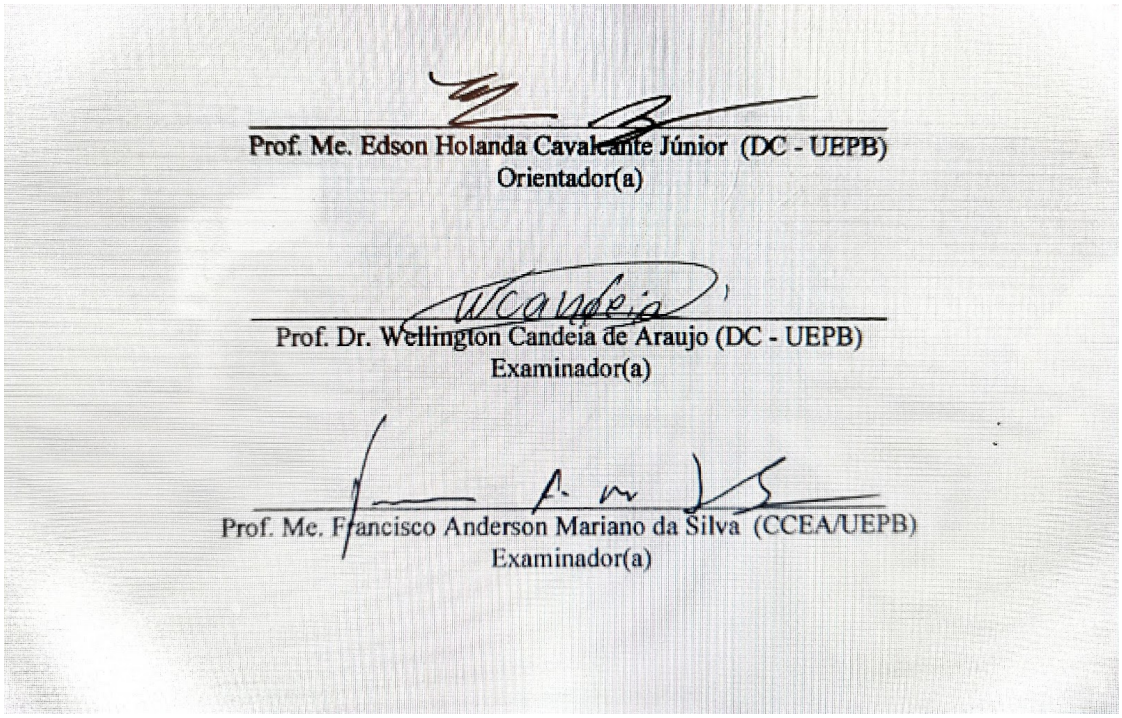
Biblioteca Central BC/UEPB

MAIANARA SALES DO Ó

APLICAÇÃO INICIAL DE ARDUINO UNO COM MÓDULO GSM SIM800L PARA TRANSMISSÃO DE DADOS DE ESTAÇÃO METEOROLÓGICA POR SMS

Trabalho de Conclusão de Curso de Graduação em Ciência da Computação da Universidade Estadual da Paraíba, como requisito à obtenção do título de Bacharel em Ciências da Computação.

Aprovada em 03 de março de 2023.



“Não faz bem viver sonhando e se esquecer de viver, lembre-se” (ROWLING, 2000, p. 185).

RESUMO

Este relatório técnico expõe a elaboração de uma aplicação de um projeto de ARDUINO UNO com módulo GSM SIM800L, que tem como finalidade a transmissão de dados de estação meteorológica por meio de SMS. Com o crescimento da utilização de estações meteorológicas no meio da agricultura, sobretudo por causa das diversas análises superficiais ao longo do dia com diversas informações da atmosfera, buscou-se consequentemente esquematizar um formato executável do processo de coleta de dados dos elementos climáticos, que são grandezas obtidas por determinados instrumentos em um local pertinente para tal coleta. A aplicação manifesta um artifício para a utilização dos dados atmosféricos de uma estação meteorológica, onde implementou a utilização de uma placa de Arduino UNO com um módulo SIM800L como ferramenta de envio e recebimento de dados via SMS. Seu desenvolvimento desde o código até o material de apoio foi desenvolvido como parte das atividades do estágio supervisionado do Curso de Ciência da Computação no Núcleo de Tecnologias Estratégicas em Saúde (NUTES). Diante do exposto no qual foi desenvolvida a aplicação, foi possível realizar testes que estão expostos e compõem este relatório. Nesse hiato, julga-se que este projeto pode ser melhorado através do empreendimento de novas tecnologias e apuração de serventia do mesmo.

Palavras-chave: estação meteorológica; arduino; transmissão de dados.

ABSTRACT

This technical report exposes the elaboration of an application of an ARDUINO UNO project with GSM module SIM800L, which has the purpose of transmitting data from a meteorological station through SMS. With the growth in the use of meteorological stations in the field of agriculture, mainly due to the various superficial analyzes throughout the day with various information from the atmosphere, an attempt was therefore made to outline an executable format for the process of collecting data on climatic elements, which are quantities obtained by certain instruments in a location relevant to such collection. The application manifests a device for using atmospheric data from a meteorological station, where it implemented the use of an Arduino UNO board with a SIM800L module as a tool for sending and receiving data via SMS. Its development, from the code to the support material, was developed as part of the activities of the supervised internship of the Computer Science Course at the Nucleus of Strategic Technologies in Health (NUTES). Given the above in which the application was developed, it was possible to carry out tests that are exposed and make up this report. In this gap, it is believed that this project can be improved through the development of new technologies and verification of its usefulness

Keywords: meteorological station; arduino; data transmission.

LISTA DE ILUSTRAÇÕES

Figura 1 - SIM800L.....14
Figura 2 - Arduino UNO.....15
Figura 3 - Regulador de tensão HW-DY02.....16
Figura 4 - Esquematização (Desenho feito com o Fritzing).....17
Figura 5 - Parte do código de testes. 19
Figura 6 - Resposta do modem ao “AT” com “OK”.....20
Figura 7 - Configuração inicial do modem (resposta).....22
Figura 8 - Recebendo ligação. 23
Figura 9 - Efetuando ligação (O número de telefone discado está encoberto). 24
Figura 10 - Parte inicial do código de ligações e SMS. 28
Figura 11 - Parte dois do código 29
Figura 12 - Parte três do código.....30
Figura 13 - Parte quatro do código.....31
Figura 14 - Parte cinco do código.....32
Figura 15 – A - 1 Parte do Manual de Comandos AT.....33
Figura 16 – A - 2 Parte do Manual de Comandos AT (2) 34

SUMÁRIO

1 INTRODUÇÃO 8
2 REFERÊNCIAL TEÓRICO 10
2.1 Meteorologia e clima 10
2.1.1 Visão Geral 10
2.1.2 Variáveis climáticas e sensores 10
3 ATIVIDADES DESENVOLVIDAS 12
3.1 Visão geral dos softwares utilizados 13
3.2 Hardwares utilizados 14
3.3 Um pouco sobre os comandos AT 17
3.4 Testes realizados com o módulo SIM800L 18
4 CONSIDERÇÕES FINAIS 25
REFERÊNCIAS 26
APÊNDICE A – CÓDIGO LIGAÇÕES E SMS PARA O MÓDULO SIM800L 28
ANEXO A – PARTE DO MANUAL DE COMANDOS AT 33

1 INTRODUÇÃO

Analisar e monitorar o clima vai muito além da vontade de possuir um auxílio para a previsão do tempo. Ter uma monitoria meteorológica torna possível a verificação de pequenas mudanças no comportamento do clima e traz inúmeras vantagens para a agricultura, transportes, controle de voo, indústria, prevenção de desastres, monitoramento dos recursos naturais, uma vez que impactam diretamente na qualidade de vida das pessoas. Para estas monitorias e análises foram criadas as estações meteorológicas (INCAPER, 2022).

São realizadas através das estações meteorológicas coleta e processamentos de dados de forma superficiais e periódica relacionados às diversas variáveis atmosféricas. Estas variáveis caracterizam o tempo, ou seja, o estado da atmosfera. Usando como exemplo a crescente popularidade dessas estações para monitoramento na agricultura vemos sua importância. Estações são uma ferramenta primordial para monitorar as condições meteorológicas na lavoura e assim ajudar as pessoas envolvidas na agricultura a tomarem decisões, por exemplo. As variáveis atmosféricas que são coletadas nestas estações influenciam diretamente a produtividade nas culturas, pois é possível medir e analisar diversos fatores que influenciam no gerenciamento das atividades nos campos, como a direção dos ventos, a precipitação acumulada no período entre as colheitas, a temperatura média em um período de tempo, entre muitos outros (INCAPER, 2022).

Para que esses dados sejam coletados com a máxima precisão é necessária a utilização de alguns equipamentos específicos e padronizações que precisam ser seguidas, tais como horários das coletas de dados e calibragem dos instrumentos utilizados.

Estações meteorológicas são equipadas com instrumentos que aferem a temperatura (termômetro), o vento (anemômetro), a chuva (pluviômetro), a pressão atmosférica (barômetro), a radiação solar (piranômetro) dentre outras variáveis atmosféricas, e podem ser de dois tipos: automáticas ou manuais (INCAPER, 2022).

As estações automáticas comunicam os dados coletados diretamente aos servidores encarregados de processar essas informações. Já para estações meteorológicas manuais, uma pessoa deve se deslocar até lá e ler os dados registrados no equipamento (INCAPER, 2022).

Visando o aumento da rapidez nessa coleta de dados, esta automação se faz necessária. Entretanto, um dos motivos para que nem todas as estações sejam automáticas está no custo elevado de implantação e manutenção destes equipamentos.

Apesar da coleta de dados meteorológicos por meio de estações automatizadas ser muito importante e provavelmente viável, o número de estações em operação no país ainda é pequeno (MELO, 2019).

Até 2019, o estado da Paraíba possuía cerca de 21 estações meteorológicas automáticas sob o controle da AESA, INMET e outras agências, o que é um número bastante pequeno para refletir com precisão o tempo e o clima de um estado com uma área de 56.585 km² (MELO, 2019).

Uma vez que o alto custo se torna um impedimento para implantações dessas estações, surge então a ideia de utilização de um microcontrolador ESP32, tornando o custo mais acessível, uma vez que consegue processar dados com baixo consumo de energia, além de capacidade de transmissão de longa distância. Um equipamento viável, completo e funcional, capaz de ser implantado facilmente a nível governamental e por agências responsáveis (MELO, 2019).

Mesmo que a tecnologia utilizando um microcontrolador ESP32 explorada por MELO et al. (2019), em seu trabalho exponha a implantação muito satisfatória de uma estação meteorológica automática, este artigo apresentará uma breve análise da utilização de um módulo Arduino UNO (semelhante ao ESP32) juntamente ao módulo modem GSM SIM800L como mais uma alternativa de envio e recebimento de dados coletados, através de mensagens SMS que utilizam radiofrequência. Uma vez que a transmissão de dados dessas estações por ventura não possa ser feita por sinal de *WiFi* LoRa, por exemplo, a transmissão por SMS pode ser uma alternativa interessante.

2 REFERÊNCIAL TEÓRICO

2.1 Meteorologia e clima

2.1.1 Visão Geral

A meteorologia se caracteriza como o estudo dos fenômenos físicos da atmosfera, que se manifestam numa localização precisa e num período relativamente curto, convencionalmente chamada de “tempo”. O método desenvolvido pela meteorologia permite atualmente a realização de previsões do estado atmosférico a curto prazo (horas, dias e semanas), fundamentais para a organização de inúmeras atividades humanas (FREITAS, 2022).

O “tempo” muitas vezes se confunde com o “clima”, no entanto o clima caracteriza-se pela análise a longo prazo de fenômenos e características atmosféricas de uma região predeterminada (média do comportamento meteorológico em anos) (FREITAS, 2022).

Usando essencialmente os mesmos elementos (pressão atmosférica, temperatura, umidade, precipitação, radiação, altitude, etc.), o clima e o tempo atmosférico são duas formas complementares de descrever a atmosfera, fazendo referência a várias escalas de tempo. (FREITAS, 2022).

Verificamos que sendo assim a meteorologia desenvolve um papel importantíssimo para a sociedade uma vez que através das informações presentes geradas é capaz de prever desastres naturais pelas variações atmosféricas como: chuvas intensas, inundações, tempestades, granizo. Essas variações interferem diretamente na vida das pessoas e na qualidade de vida, tais como: recursos hídricos, moradia, construção civil, agricultura, transportes, monitoramento de voo, dentre tantas outras.

2.1.2 Variáveis climáticas e sensores

Por ser um equipamento responsável por coletar, medir e registrar eventos climáticos, o objetivo de uma estação meteorológica é possibilitar o acompanhamento das variações climáticas e não de prever o futuro. As informações por eles geradas são realizadas por meio de diversos instrumentos, como

termômetros, barômetros, medidores de umidade, anemômetros, catapultas e birutas, cada um responsável por captar determinado fator climático (MELO, 2019).

As variáveis climáticas, que também são chamadas de elementos climáticos, são grandezas que podem ser medidas em determinado local por instrumentos apropriados, são elas:

-Temperatura: refere-se ao aquecimento do ar, ou seja, ao grau de calor presente na atmosfera. O instrumento utilizado para medir esta grandeza é o termômetro, nas estações é conhecido como sensor de temperatura (INMET, 2023).

-Umidade atmosférica: é a quantidade de água em estado gasoso, em forma de vapor, presente na atmosfera. O instrumento utilizado para medir esta grandeza é o higrômetro, conhecido também como sensor de umidade (BATISTA, 2022).

-Pressão atmosférica: é a força exercida pela atmosfera sobre a superfície terrestre, o peso. O instrumento utilizado para medir esta grandeza é o barômetro (BATISTA, 2022).

-Vento: é a movimentação de frações de ar na direção de uma região de maior pressão para uma de menor pressão, que possui determinada intensidade. O instrumento utilizado para medir esta grandeza é o anemômetro, sensor de velocidade do vento, e a biruta, sensor de direção do vento (INMET, 2023).

-Chuva, ou precipitação pluviométrica: é o fenômeno relacionado à queda de água da atmosfera, na forma líquida (chuva) ou sólida (neve e granizo). O instrumento utilizado para medir essa grandeza é o pluviômetro, sensor de chuva (INMET, 2023).

3 ATIVIDADES DESENVOLVIDAS

Estações meteorológicas automatizadas incorporam sensores eletrônicos para medir condições climáticas infinitamente variáveis. Todos os dados são enviados para um local central conhecido como datalogger (ou registrador de dados), onde programas de computador integram todas as informações e permitem um estudo das condições climáticas. As informações meteorológicas são coletadas minuto a minuto e de hora em hora, esses mesmos dados são integrados e disponibilizados para transmissão a um centro de meteorologia via wireless (alguma rede sem fio), GPRS (Serviços Gerais de Pacote por Rádio) ou satélite (MELO, 2019).

Como explica em seu trabalho, MELO et al. (2019), entre os elementos presentes em todo o projeto da estação meteorológica, o datalogger tem função de receber os dados fornecidos pelos sensores, interpretá-los, armazenar as informações coletadas em uma mídia removível e transmiti-los para a próxima etapa do processo, ou seja, o repetidor (que recebem os sinais e os retransmitem com sua intensidade e definições originais) ou o gateway (que interpreta os dados de uma rede para outra rede de arquitetura diferente), por meio de tecnologia LoRa. O datalogger utilizado por ele na ocasião foi construído utilizando como elemento principal o *WiFi* LoRa 32, uma placa de desenvolvimento fabricada pela empresa Heltec Automation e baseada na construção do ESP32, um produto da empresa Espressif.

Uma vez que o meio mais utilizado para transmissão desses dados é por rede sem fio *WiFi*, foi pensado em como esses dados poderiam se comportar em uma transmissão por rádio ou SMS, caso ocorra uma falha de sinal por este dispositivo. Sendo assim foi imaginado a utilização de uma placa de Arduino UNO com um módulo SIM800L.

3.1 Visão geral dos softwares utilizados

Para auxiliar neste relatório e ter um primeiro contato com este tipo de desenvolvimento foi necessário um pouco mais de estudo sobre sistemas embarcados e placas de Arduino e ESP32.

Para isto foram escolhidos dois frameworks (que são conjuntos de bibliotecas que implementam determinada linguagem de programação) importantíssimos: o Arduino IDE, que é uma aplicação multiplataforma escrita em Java, possuindo a capacidade de programar em C/C++. Inclui um editor de código com recursos de realce de sintaxe, parênteses correspondentes e indentação automática, sendo capaz de compilar e carregar programas para a placa rapidamente; e o Visual Studio Code (VSCode) que é um programa *open-source* (código aberto, pode ser visualizado, modificado e distribuído por qualquer pessoa) desenvolvido pela Microsoft com suporte para *Windows*, *MacOS* e *Linux*. É um software livre e de código aberto, baseado no framework Electron. É altamente customizável, liberado para que os usuários mudem o tema do editor, as teclas de atalho, entre outros aspectos, permitindo assim a adição de várias extensões como o PlatformIO IDE, tendo suporte para diversos tipos de placas, entre elas Arduino e Raspberry Pi (CERQUEIRA, 2018).

3.2 Hardwares utilizados

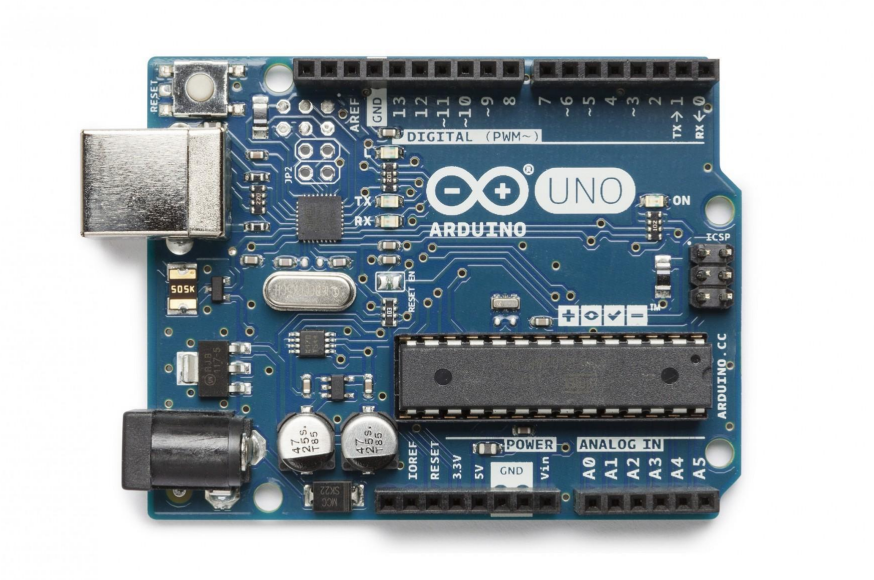
Neste trabalho utilizamos uma placa Arduino UNO (Fig. 2) para conectar o modem. O Arduino é um microcontrolador que possui 14 pinos, sendo destes, 6 entradas analógicas, uma entrada USB, uma conector de energia e um botão de reset. Possuindo tudo para o suporte do microcontrolador, bastando apenas conectá-lo ao computador por meio de um cabo USB. “UNO” significa “um” em italiano e teve esse nome escolhido para o lançamento do Arduino Software IDE (ARDUINO, 2018).

O SIM800L (Fig. 1) é um modem, *shield* para Arduino, fabricado pela SIMCom. Os *shields* são placas que podem ser acopladas em uma placa Arduino, adicionando funcionalidades. Esse *shield* tem uma capacidade enorme para vários tipos de projetos utilizando um chip (como os de celulares) e uma antena (nesse caso foi utilizada a antena mola que acompanha este *shield*). Com ele pode-se enviar e receber mensagens de SMS assim como ligações. Ele habilita automaticamente o suporte GSM e GPRS (SIMCOM, 2018).

Figura 1 - SIM800L

Fonte: Avalectronics.

Figura 2 - Arduino UNO.



Fonte: Coeur grenadine.

A tecnologia GSM (Sistema Global de Comunicações Móveis) surgiu acompanhada de uma proposta de prestação de serviços de baixo custo, como a troca de mensagens de texto, além de baixos custos de infraestrutura para as operadoras. O GSM difere de seus predecessores porque os canais de sinal e voz são digitais ; por isso é chamada de tecnologia 2G , ou tecnologia de segunda geração. Este serviço é suportado no módulo SIM800L para além do GPRS, cujo objetivo é possibilitar a transmissão de dados através de redes de pacotes para que a rede móvel possa ser ligada à internet (SMARTINS e PIRES, 2019).

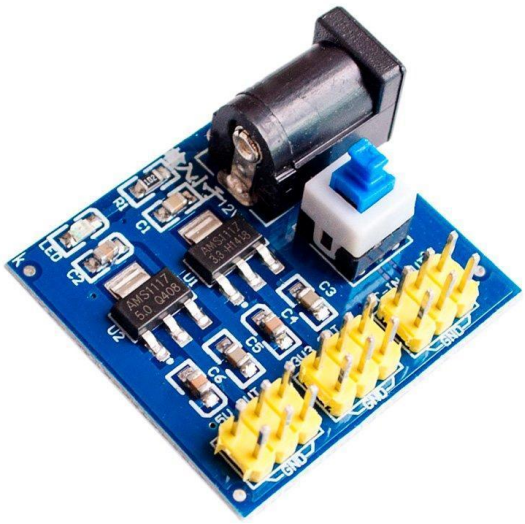
A transmissão de dados era feita por comutação de circuitos pela tecnologia GSM antes do GPRS, isto é, uma conexão entre dois aparelhos era estabelecida, e em seguida a comunicação era feita de forma ininterrupta. Com a implementação do GPRS (Serviços Gerais de Pacotes por Rádio), passou a se utilizar a comunicação de dados por comutação de pacotes, onde a informação é dividida em vários pacotes na origem, transmitida e remontada no destino. Cada pacote leva o endereço do destino bem como a informação para montagem no destino. Um avanço significativo para comunicação de dados móveis, o sistema GSM com GPRS integrado recebeu o apelido geracional 2.5G (SMARTINS e PIRES, 2019).

Com essa tecnologia presente nesse modem simples, buscou-se uma comunicação entre estações de monitoramento meteorológico com servidores, onde

os dados coletados de cada sensor em cada estação serão guardados, e outra comunicação do servidor para rádios em que estes irão receber os dados armazenados nos servidores quando solicitado.

Para o uso correto desses equipamentos e transmissão segura e eficiente foi utilizado um Arduino UNO, um regulador de tensão HW-DY02 (Fig. 3) e uma fonte externa de 12v, pois esse modem funciona melhor na faixa de tensão de 3,7v a 4,5v, mais especificamente.

Figura 3 - Regulador de tensão HW-DY02.

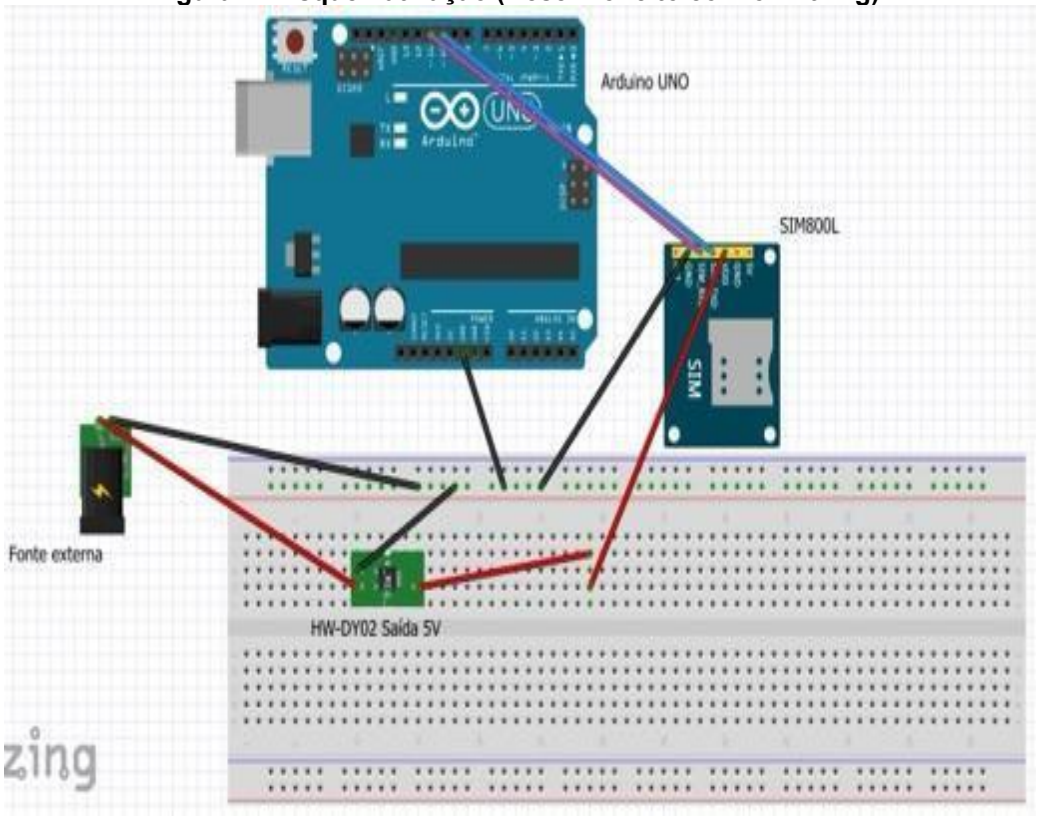


Fonte: Voltiq.

Os pinos TX e RX do modem foram conectados nas portas 10 e 11 do Arduino, respectivamente, para simular uma conexão serial, usada pelo modem, já que essas portas do Arduino são digitais. Assim pode-se usar a biblioteca *Software Serial* na programação do módulo. Esse esquema é mostrado na imagem seguinte utilizando o Fritzing (Fig. 4).

O Fritzing é uma ferramenta de código aberto criada como iniciativa para criar protótipos eletrônicos e circuitos (com suporte para vários tipo de placas, incluindo Arduino), ensinar eletrônica em sala de aula e fazer layouts. Tanto em software quanto no site, permitindo o compartilhamento de projetos entre os usuários (FRITZING, 2019).

Figura 4 - Esquematização (Desenho feito com o Fritzing).



Fonte: Elaborado pela autora, 2019.

3.3 Um pouco sobre os comandos AT

Depois de alimentado corretamente o modem foi reconhecido pela placa de Arduino e começou a receber os códigos de testes enviados para ele. Nessa etapa, juntamente com a Linguagem C, foram aplicados os comandos AT para configuração e testes.

Os comandos AT (cujo prefixo AT significa atenção) são descendentes diretos do Padrão Hayes, criado na década de 80 para comunicação direta dos computadores pessoais com conexões telefônicas controlando um modem. O conjunto de comandos AT é uma linguagem de comandos com uma série de cadeias de texto curtas, que se combinam para emitir comandos completos para diferentes operações, como desligar o telefone e alterar os parâmetros de conexão para os modems. A maioria dos modems de computadores pessoais seguem as especificações do conjunto de comandos AT (SIMCOM, 2018).

O prefixo AT deve ser definido em cada linha antes de cada comando mais específico e um parâmetro (opcional), ou apenas uma vez, quando os comandos

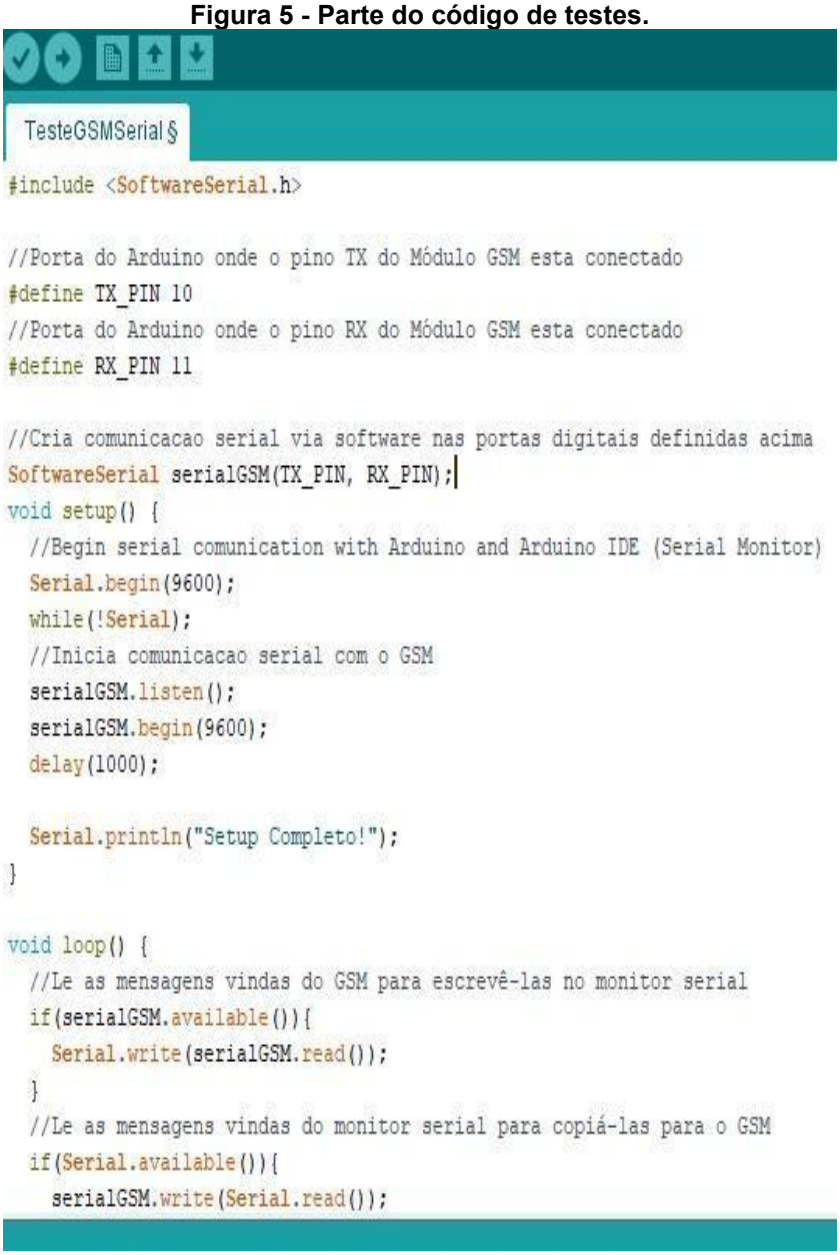
estão na mesma linha. Um exemplo de linha de comando é “AT+CMGF=1”, onde “AT” é o prefixo do comando, “+CMGF” é o comando específico, indicando o envio de uma mensagem de SMS e “=1” é o parâmetro, que indica o formato dessa mensagem, no caso “1” significando que a mensagem será enviada em forma de texto (SIMCOM, 2018).

Uma parte do manual de comandos AT está nas figuras do Anexo A deste trabalho.

3.4 Testes realizados com o módulo SIM800L

Como citado anteriormente o objetivo com o SIM800L e o Arduino foi principalmente realizar e receber chamadas, e enviar e receber SMS de forma a verificar e testar o funcionamento de ambos, para que na implantação final das estações meteorológicas elas sejam capazes de se conectar com o módulo e enviar informações através de SMS, por exemplo, dos dados coletados, de forma segura e eficaz. Nesse sentido foram feitos os testes de implementação, utilizando códigos bem simples com linguagem C e os comandos AT.

Figura 5 - Parte do código de testes.



TesteGSMSerial §

```
#include <SoftwareSerial.h>

//Porta do Arduino onde o pino TX do Módulo GSM esta conectado
#define TX_PIN 10
//Porta do Arduino onde o pino RX do Módulo GSM esta conectado
#define RX_PIN 11

//Cria comunicacao serial via software nas portas digitais definidas acima
SoftwareSerial serialGSM(TX_PIN, RX_PIN);
void setup() {
  //Begin serial comunication with Arduino and Arduino IDE (Serial Monitor)
  Serial.begin(9600);
  while(!Serial);
  //Inicia comunicacao serial com o GSM
  serialGSM.listen();
  serialGSM.begin(9600);
  delay(1000);

  Serial.println("Setup Completo!");
}

void loop() {
  //Le as mensagens vindas do GSM para escrevê-las no monitor serial
  if(serialGSM.available()){
    Serial.write(serialGSM.read());
  }
  //Le as mensagens vindas do monitor serial para copiá-las para o GSM
  if(Serial.available()){
    serialGSM.write(Serial.read());
```

Fonte: Elaborado pela autora, 2019.

Figura 6 - Resposta do modem ao “AT” com “OK”.

Fonte: Elaborada pela autora, 2019.

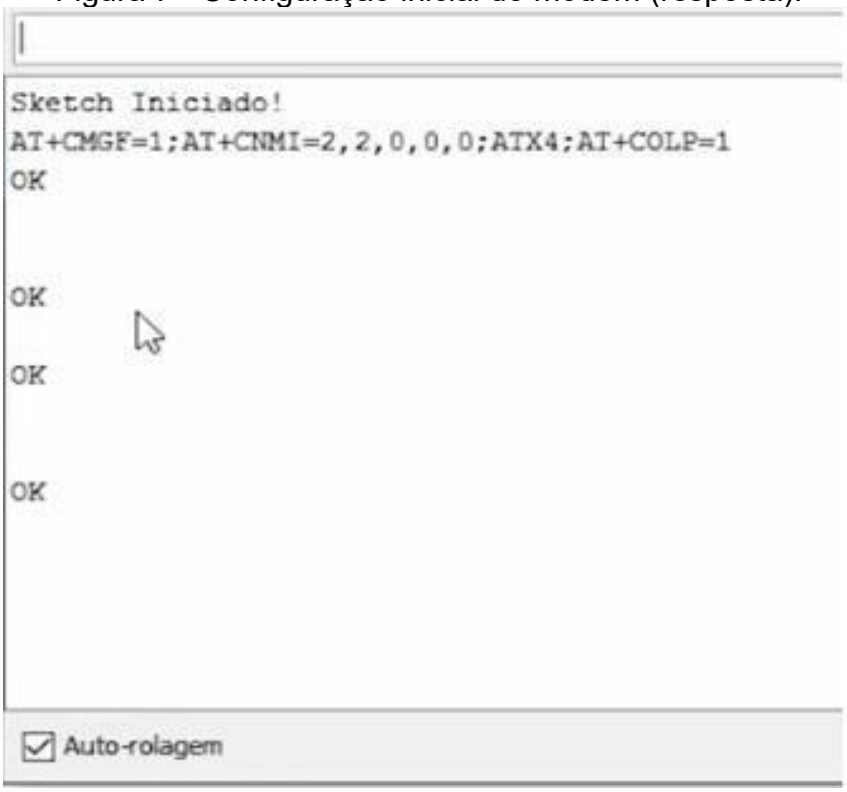
Nas imagens acima (Fig. 7 e 8) é observável uma parte do código de testes e na seguinte a imagem do Monitor Serial do Arduino IDE. Na imagem podemos ver o comando “AT” enviado para o modem, seguido de uma resposta “OK”, significando que o modem está funcionando e recebendo a informação de forma correta.

Os trechos de código abaixo, indicam uma ligação sendo feita, pode-se observar o que cada comando AT especifica na configuração do modem através das linhas de comentário do código, através das linhas comentadas e detalhadas de código. Em seguida vemos as respostas do modem no Monitor Serial.

Para a visualização do código inteiro basta consultar o Apêndice A deste trabalho.

```
#include <SoftwareSerial.h>

SoftwareSerial serialGSM(10, 11);
// TX e RX do SIM800L nos pinos 10 e 11 do Arduino (UNO), respectivamente
bool temSMS = false;
String telefoneSMS;
String dataHoraSMS;
String mensagemSMS;
String comandoGSM = "";
String ultimoGSM = "";

#define senhaGsm "1234" //Para receber e enviar SMS, confirmar número
//#define numeroCall "9xxxxxxxx" //Número que irá receber chamadas do
módulo//////////////////////
//Funções devem ser declaradas antes do setup, coisas do C para Arduino

void leGSM();
void enviaSMS(String telefone, String mensagem);
void configuraGSM();

void setup() {

 Serial.begin(9600); //Velociade padrão de leitura e resposta da serial
 serialGSM.begin(9600);

 Serial.println("Sketch Iniciado!"); //Verificação do módulo
 configuraGSM();
}
.....
void enviaSMS(String telefone, String mensagem) {   //a mensagem será a string de dados de cada
sensor da estação
 serialGSM.print("AT+CMGS=\"" + telefone + "\"\n"); //o telefone é o número de telefone do chip do
módulo que receberá os dados
 serialGSM.print(mensagem + "\n");
 serialGSM.print((char)26);
}

void configuraGSM() {
  serialGSM.print("AT+CMGF=1\n;AT+CNMI=2,2,0,0,0\n;ATX4\n;AT+COLP=1\n"); //Algumas
configurações específicas
  //AT+CMGF=1, Formato de SMS (1 = texto, 0 = dados(PDU)),
  //AT+CNMI=2,2,0,0,0, Indicadores de nova mensagem (Mode 2 = mensagem solicitada
diretamente,
  //                    mt 2 = verifica tipografia das mensagens, itálico, por exemplo,
  //                    bm 0 = roteamento em modem,
  //                    ds 0 = Sem SMS-STATUS-REPORT,
  //                    bfr 0 = Sem mensagem no buffer),
  //ATX4, Monitor de monitoramento de chamada (4 = habilita dial tone e detecção de linha
ocupada),
  //AT+COLP=1, Linha conectada, habilitar ou náo notificação (1 = habilitar)
}
```

Figura 7 - Configuração inicial do modem (resposta).



Fonte: Elaborado pela autora, 2019.

Figura 8 - Recebendo ligação.

Fonte: Autoria própria, 2019.

Figura 9 - Efetuando ligação (O número de telefone discado está encoberto).

Fonte: Autoria própria, 2019.

4 CONSIDERÇÕES FINAIS

O projeto no qual este relatório se baseia, principalmente em relação à fundamentação teórica, foi feito com o intuito de criar uma Estação Meteorológica com Tecnologia LoRa, encontra-se implementado e dentro do esperado pelo corpo técnico do Núcleo de Tecnologias Estratégicas em Saúde (NUTES) no campus Campina Grande da UEPB e é atualmente utilizado como membro do portfólio de projetos construídos ao longo de vários anos de atuação do núcleo. As tecnologias utilizadas para realizar o projeto em si ocupam um espaço de grande aceitação na área de engenharia e possuem tecnologia de ponta em um ponto de vista inovador.

Um ponto de destaque no desenvolvimento desta pesquisa utilizando o shield SIM800L com o Arduino foi a dificuldade de encontrar mais auxílio, materiais e outros trabalhos utilizando comandos AT para programar e enviar os comandos para o módulo. Desse modo, foi muito difícil programar utilizando apenas o manual de comandos AT, em inglês, com os comandos simples, porém sem exemplos reais, sendo utilizado com linguagem de programação, como a linguagem C, por exemplo, como foi utilizada neste trabalho.

Também foi observado que a linguagem de programação C, além de ser uma linguagem estruturada, também possui outras especificações típicas para placas como o Arduino, como por exemplo as funções inseridas no código devem ser declaradas antes do “void setup() {}”.

Para finalizar pode-se ressaltar que, tanto para ampliação de área de interesse quanto sugestão de trabalhos futuros, o estudo em proteção de dados é necessário, para garantir mais segurança e utilização de criptografia, assim como um aprofundamento de tecnologia via rádio, uma vez que foi provado também a capacidade de uma placa Arduino em funcionamento com o shield SIM800L. Também como uma sugestão de trabalhos futuros estão testes para transmissão de dados utilizando os dados da Estação Meteorológica hoje terminada e presente no NUTES, demonstrando a aplicação e integração destas tecnologias. A utilização deste shield para apoio de uma estação pode ser interessante e totalmente aplicável.

REFERÊNCIAS

ARDUINO. *In*: Arduino. SP. Disponível em: https://www.arduino.cc/. Acesso em: 6 dez. 2018.

BRINCANDO COM IDEIAS. **Módulos para Arduino - Vídeo 16 - GSM SIM800L.** [S.: l. s. n]. 16 jul. 2017. 1 vídeo (37 min 17 s). Disponível em: https://www.youtube.com/watch?v=GbVXixOUUPM&t=1980s. Acesso em: 6 dez. 2018.

BATISTA, Carolina. **Fatores que Influenciam o Clima**. *In*: TodaMateria. SP. Disponível em: https://www.todamateria.com.br/fatores-que-influenciam-o-clima. Acesso em: 31 jan. 2023.

CAMPOS, Augusto. **ESP8266: Comandos AT.** *In*: Arduino. SP**.** Disponível em: https://br-arduino.org/2015/11/esp8266-comandos-at.html. Acesso em: 29 nov. 2018.

CERQUEIRA, Giovanni. **Como programar o Arduino com o Visual Studio Code e PlatformIO IDE.** Disponível em: https://www.embarcados.com.br/arduino-vscode-platformio/. Acesso em: 27 nov. 2018.

FREITAS, Michele Martinenghi Sidronio de. **Meteorologia**. *In*: InfoEscola. SP. Disponível em: https://www.infoescola.com/geografia/meteorologia. Acesso em: 31 jan. 2023.

FRITZING. Eletrônica Facilitada. Disponível em: https://fritzing.org/ . Acesso em: 18 jun. 2019.

INSTITUTO CAPIXABA DE PESQUISA, ASSISTÊNCIA TÉCNICA E EXTENSÃO RURAL - INCAPER (INCAPER). **Estações Meteorológicas**. Disponível em: https://meteorologia.incaper.es.gov.br/estacoes-meteorologicas. Acesso em: 31 jan. 2023.

INMET. Estação meteorológica de observação de superfície automática. 2015. Instituto Nacional de Meteorologia (INMET). 2012. *In*: Inmet. BL. Disponível em https://portal.inmet.gov.br/sobre-meteorologia. Acesso em: 16 fev. de 2023.

MELO, Widson Gomes de. **Estação Meteorológica Automática com Tecnologia LoRa**. 2019. Disponível em: http://dspace.bc.uepb.edu.br/jspui/bitstream/123456789/22051/1/PDF%20-%20Widson%20Gomes%20de%20Melo.pdf. Acesso em: 31 jan. 2023.

ROWLING, J. K. **Harry Potter e a Pedra Filosofal**. Rio de Janeiro: Rocco, 2000. 185p.

SMARTINS & PIRES LTDA. **Sabe como funciona o sistema GSM e GPRS?**. *In*: SM. Santa Barbosa, SP. Disponível em: https://www.smartinstec.com.br/sabe-como-funciona-o-sistema-gsm-e-gprs. Acesso em: 18 jun. 2019.

SIMCOM, A Company of SIM Tech. SIM800 Series AT Command Manual. V1.09 Release. 3 de agosto de 2015. Disponível em: https://simcom.ee/documents/SIM808/SIM800%20Series_AT%20Command%20Manual_V1.09.pdf. Acesso em: 27 nov. 2018.

SIMCOM, A Company of SIM Tech. SIM800L Hardware Design. V1.09 Release. 30 de junho de 2016. Disponível em: https://simcom.ee/documents/SIM800/SIM800_Hardware%20Design_V1.09.pdf. Acesso em: 27 nov. 2018.

APÊNDICE A – CÓDIGO LIGAÇÕES E SMS PARA O MÓDULO SIM800L

Figura 10 - Parte inicial do código de ligações e SMS.

```
#include <SoftwareSerial.h>

SoftwareSerial serialGSM(10, 11);
// TX e RX do SIM800L nos pinos 10 e 11 do Arduino (UNO), respectivamente

bool temSMS = false;
String telefoneSMS;
String dataHoraSMS;
String mensagemSMS;
String comandoGSM = "";
String ultimoGSM = "";

#define senhaGsm "1234" //Para receber e enviar SMS, confirmar número
//#define numeroCall "9xxxxxxxx" //Número que irá receber chamadas do módulo//////////////////////



//Funções devem ser declaradas antes do setup, coisas do C para Arduino
void leGSM();
void enviaSMS(String telefone, String mensagem);
void configuraGSM();

void setup() {

  Serial.begin(9600); //Velociade padrão de leitura e resposta da serial
  serialGSM.begin(9600);

  Serial.println("Sketch Iniciado!"); //Verificação do módulo
  configuraGSM();
}
```

Fonte: Elaborado pela autora, 2019.

Figura 11 - Parte dois do código

```
31
32  void loop() {
33    leGSM();
34
35    if (comandoGSM != "") { //Verifica o que foi feito e o comando utilizado anteriormente
36        Serial.println(comandoGSM);
37        ultimoGSM = comandoGSM;
38        comandoGSM = "";
39    }
40
41    if (temSMS) { //Verifica recebimento de SMS e envio
42
43       Serial.println("Chegou Mensagem!!");
44       Serial.println();
45
46       Serial.print("Remetente: ");
47       Serial.println(telefoneSMS);
48       Serial.println();
49
50       Serial.print("Data/Hora: ");
51       Serial.println(dataHoraSMS);
52       Serial.println();
53
54       Serial.println("Mensagem:");
55       Serial.println(mensagemSMS);
56       Serial.println();
57
58       mensagemSMS.trim();
59       /*if ( mensagemSMS == senhaGsm ) {
60          Serial.println("Enviando SMS de Resposta.");
61           enviaSMS(telefoneSMS, "SMS Recebido e Senha OK!"); //Envia SMS de confirmação de recebimento
62       }*/
```

Fonte: Elaborado pela autora, 2019.

Figura 12 - Parte três do código

```
63        temSMS = false;
64    }
65
66  }
67
68  void leGSM()
69  {
70    static String textoRec = "";
71    static unsigned long delay1 = 0;
72    static int count=0;
73    static unsigned char buffer[64];
74
75    if (serialGSM.available()) {
76
77       while(serialGSM.available()) {
78
79          buffer[count++] = serialGSM.read();
80          if(count == 64)break;
81       }
82
83       textoRec += (char*)buffer;
84       delay1   = millis();
85
86       for (int i=0; i<count; i++) {
87           buffer[i]=NULL;
88       }
89       count = 0;
90    }
91
```

Fonte: Elaborado pela autora, 2019.

Figura 13 - Parte quatro do código

```
93    if ( ((millis() - delay1) > 100) && textoRec != "" ) {
94
95       if ( textoRec.substring(2,7) == "+CMT:" ) {
96          temSMS = true;
97       }
98
99       if (temSMS) {
100
101         telefoneSMS = "";
102         dataHoraSMS = "";
103         mensagemSMS = "";
104
105         byte linha = 0;
106         byte aspas = 0;
107         for (int nL=1; nL < textoRec.length(); nL++) {
108
109            if (textoRec.charAt(nL) == '"') {
110               aspas++;
111               continue;
112            }
113
114            if ( (linha == 1) && (aspas == 1) ) {
115               telefoneSMS += textoRec.charAt(nL);
116            }
117
118            if ( (linha == 1) && (aspas == 5) ) {
119               dataHoraSMS += textoRec.charAt(nL);
120            }
121
122            if ( linha == 2 ) {
123               mensagemSMS += textoRec.charAt(nL);
124            }
```

Fonte: Elaborado pela autora, 2019.

Figura 14 - Parte cinco do código

```
125
126             if (textoRec.substring(nL - 1, nL + 1) == "\r\n") {
127                linha++;
128             }
129          }
130       } else {
131         comandoGSM = textoRec;
132       }
133
134       textoRec = "";
135    }
136 }
137
138
139 void enviaSMS(String telefone, String mensagem) {    //a mensagem será a string de dados de cada sensor da estação
140    serialGSM.print("AT+CMGS=\"" + telefone + "\"\n"); //o telefone é o número de telefone do chip do módulo que receberá os dados
141    serialGSM.print(mensagem + "\n");
142    serialGSM.print((char)26);
143 }
144
145
146 void configuraGSM() {
147     serialGSM.print("AT+CMGF=1\n;AT+CNMI=2,2,0,0,0\n;ATX4\n;AT+COLP=1\n"); //Algumas configurações específicas
148     //AT+CMGF=1, Formato de SMS (1 = texto, 0 = dados(PDU)),
149     //AT+CNMI=2,2,0,0,0, Indicadores de nova mensagem (Mode 2 = mensagem solicitada diretamente,
150     //                          mt 2 = verifica tipografia das mensagens, itálico, por exemplo,
151     //                          bm 0 = roteamento em modem,
152     //                          ds 0 = Sem SMS-STATUS-REPORT,
153     //                          bfr 0 = Sem mensagem no buffer),
154     //ATX4, Monitor de monitoramento de chamada (4 = habilita dial tone e detecção de linha ocupada),
155     //AT+COLP=1, Linha conectada, habilitar ou não notificação (1 = habilitar)
156 }
```

Fonte: Elaborado pela autora, 2019.

ANEXO A – PARTE DO MANUAL DE COMANDOS AT

Figura 15 – A - 1 Parte do Manual de Comandos AT

1.4.1 Basic syntax

These AT commands have the format of "**AT**<*x*><*n*>", or "**AT&**<*x*><**n**>", where "<*x*>"is the Command, and "<*n*>"is/are the argument(s) for that Command. An example of this is "**ATE**<*n*>", which tells the DCE whether received characters should be echoed back to the DTE according to the value of "<*n*>". "<*n*>" is optional and a default will be used if missing.

1.4.2 S Parameter syntax

These AT commands have the format of "**ATS**<*n*>=<*m*>", where "<*n*>" is the index of the S register to set, and "<*m*>"is the value to assign to it. "<*m*>" is optional; if it is missing, then a default value is assigned.

1.4.3 Extended Syntax

These commands can operate in several modes, as in the following table:

Table 1: Types of AT commands and responses

Test Command	AT+<x>=?	The mobile equipment returns the list of parameters and value ranges set with the corresponding Write Command or by internal processes.
Read Command	AT+<x>?	This command returns the currently set value of the parameter or parameters.
Write Command	AT+<x>=<…>	This command sets the user-definable parameter values.
Execution Command	AT+<x>	The execution command reads non-variable parameters affected by internal processes in the GSM engine.

1.4.4 Combining AT commands on the same Command line

You can enter several AT commands on the same line. In this case, you do not need to type the "**AT**" or "**at**" prefix before every command. Instead, you only need type "**AT**" or "**at**" the

Fonte: Simcom.ee.

Figura 16 – A - 2 Parte do Manual de Comandos AT (2)

2 AT Commands According to V.25TER

These AT Commands are designed according to the ITU-T (International Telecommunication Union, Telecommunication sector) V.25ter document.

2.1 Overview of AT Commands According to V.25TER

Command	Description
A/	Re-issues the last command given
ATA	Answer an incoming call
ATD	Mobile originated call to dial a number
ATD><N>	Originate call to phone number in current memory
ATD><STR>	Originate call to phone number in memory which corresponds to field <str>
ATDL	Redial last telephone number used
ATE	Set command echo mode
ATH	Disconnect existing connection
ATI	Display product identification information
ATL	Set monitor speaker loudness
ATM	Set monitor speaker mode
+++	Switch from data mode or ppp online mode to command mode
ATO	Switch from command mode to data mode
ATP	Select pulse dialling
ATQ	Set result code presentation mode
ATS0	Set number of rings before automatically answering the call
ATS3	Set command line termination character
ATS4	Set response formatting character
ATS5	Set command line editing character
ATS6	Pause before blind dialling
ATS7	Set number of seconds to wait for connection completion
ATS8	Set number of seconds to wait for comma dial modifier encountered in dial string of D command
ATS10	Set disconnect delay after indicating the absence of data carrier
ATT	Select tone dialing
ATV	TA response format

Fonte: Simcom.ee.