



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

**DESENVOLVIMENTO DE UM SISTEMA DE COLETA DE DADOS
AMBIENTAIS BASEADO EM PROCESSADORES DA FAMÍLIA ATMEGA
COM TRANSMISSÃO DE DADOS VIA WI-FI.**

Adam Pereira Bidiaki

Relatório de encerramento das atividades do bolsista de Iniciação Científica do programa PIBIC, orientado pelo Dr. Waldeir Amaral Vilela e coorientado pelo Dr. Ricardo Toshiyuki Irita.

INPE

São José dos Campos

2021



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

**DESENVOLVIMENTO DE UM SISTEMA DE COLETA DE DADOS
AMBIENTAIS BASEADO EM PROCESSADORES DA FAMÍLIA ATMEGA
COM TRANSMISSÃO DE DADOS VIA WI-FI.**

Adam Pereira Bidiaki

Relatório de encerramento das atividades do bolsista de Iniciação Científica do programa PIBIC, orientado pelo Dr. Waldeir Amaral Vilela e coorientado pelo Dr. Ricardo Toshiyuki Irita.

INPE

São José dos Campos

2021

RESUMO

Este é um relatório de encerramento das atividades do bolsista do projeto de Iniciação Científica do programa PIBIC desenvolvido no Instituto Nacional de Pesquisas Espaciais (INPE), onde o principal objetivo foi dar continuidade ao desenvolvimento de uma unidade de coleta de dados ambientais de baixo custo que já havia sido iniciado por outro bolsista IC. Essa unidade de coleta de dados está sendo desenvolvida para uso no monitoramento ambiental em um experimento de estudo do potencial fotovoltaico em São José dos Campos. Os dados ambientais coletados são usados para correlacionar os fenômenos ambientais com o desempenho de módulos fotovoltaicos de diversas tecnologias em estudo no laboratório GDF/COPDT/INPE (Grupo de Dispositivos Fotovoltaicos da Coordenação de Pesquisa e Desenvolvimento Tecnológico do INPE). Nas primeiras versões do projeto sessa unidade de coleta de dados, as medições obtidas nos sensores ambientais eram armazenadas na memória do próprio dispositivo de aquisição de dados, cujo acesso só era possível através da transferência dos dados removendo o cartão de memória ou acoplando o dispositivo diretamente ao um computador através de cabo USB. O uso do cabo acoplado a entrada USB de um microcomputador, embora seja funcional, muitas vezes apresenta dificuldade para uso em campo, principalmente quando os experimentos são montados em locais de difícil acesso. Devido a esse inconveniente, viu-se a necessidade de desenvolver uma solução que facilitasse o acesso remoto dos dados por meio da transmissão sem fio. Com esse objetivo, em trabalhos anteriores, foi realizado o estudo de tecnologias de transmissão sem fio, sendo concebida uma solução de bancada com um projeto de um circuito eletrônico e especificação necessários para a sua implementação. Este trabalho teve como objetivo principal dar continuidade ao projeto iniciado por outro bolsista de IC, elaborar e executar uma série de testes no circuito desenvolvido e acoplar este recurso ao dispositivo inicialmente proposto realizando a integração do módulo Wi-Fi com o módulo de aquisição de dados previamente desenvolvido, incluindo a codificação e a instalação do novo sistema em um experimento em campo.

Nesse relatório são apresentadas as atividades e os resultados obtidos no período de dois meses, período esse referente ao início da bolsa de IC até o pedido de cancelamento por parte deste relator por motivos pessoais.

Palavras-Chave: Dados Meteorológicos, Plataforma de Coleta de Dados, Wi-Fi, Atmega, ESP-01.

LISTA DE ILUSTRAÇÕES

Figura 1: Exemplo de PCD.....	4
Figura 2: Exemplo de conexão com Wi-Fi.....	5
Figura 3 - Exemplo de microcontrolador	6
Figura 4 – Arduino UNO.....	7
Figura 5: ESP-01	7
Figura 6: Portas de comunicação do módulo cartão de memória SD.....	8
Figura 7: Arquivo DATA.TXT	10
Figura 8: Mensagem salvas no arquivo DATA.TXT	11
Figura 9: Montagem física do módulo leitor de cartão SD	11
Figura 10: Informações de data e hora geradas pela programação teste do módulo RTC	12
Figura 11: Montagem física do módulo RTC.....	13
Figura 12: Dados do potenciômetro enviados via WiFi pelo ESP-01	14

SUMÁRIO

1 INTRODUÇÃO.....	1
1.1. Objetivos Gerais Inicialmente Proposto	2
1.2. Objetivos Específicos Inicialmente Propostos.....	2
1.3. Objetivos atingidos no período 2 meses	2
2. INTRODUÇÃO TEÓRICA	3
2.1. Plataforma de Coleta de Dados.....	3
2.2. Transmissão Sem Fio.....	4
2.3. Microcontroladores.....	5
2.3.1. Arduino	6
2.3.2. ESP-01	7
3. ATIVIDADES DESENVOLVIDAS.....	8
3.1. Módulo leitor de cartão SD.....	10
3.2. Módulo RTC	12
3.3. ESP-01	13
4. CONCLUSÃO.....	15
REFERÊNCIAS	16
ANEXO I.....	17
ANEXO II	19
ANEXO III	20
ANEXO IV	22

1 INTRODUÇÃO

O grupo de pesquisa em dispositivos fotovoltaicos no INPE em São José dos Campos. (GDF/COPDT/INPE) possui experimentos em campo com objetivo de estudar o comportamento de diferentes tecnologias solares fotovoltaicas. Os dados obtidos nestes experimentos, bem como dados ambientais utilizados nas análises desses experimentos, são coletados automaticamente através de dispositivos comerciais de custo elevado. Para reduzir os custos destes experimentos visando a instalação de novos experimentos em diferentes localidades, foi proposto, como um projeto de iniciação científica, o desenvolvimento de uma estação de coleta de dados ambientais de baixo custo baseado em processadores da família Atmega.

O sistema de aquisição de dados previamente desenvolvido foi baseado na plataforma de prototipagem eletrônica com código aberto denominada Arduino. Porém o sistema desenvolvido dispõe somente de uma memória do tipo SD onde são armazenados os dados obtidos através dos sensores ambientais. O acesso aos dados contidos no cartão SD só podem ser acessados através de duas maneiras: remoção do cartão de memória do dispositivo e posterior leitura em um microcomputador com entrada adequada a esse tipo de cartão ou pela conexão de um cabo USB diretamente entre o dispositivo de aquisição de dados e o microcomputador. Ambos os procedimentos não são soluções práticas para o usuário que precisa de acessar os dados com grande frequência.

Com o objetivo de facilitar o trabalho dos pesquisadores e usuários do sistema de aquisição de dados, foi criada a proposta inicial deste projeto de iniciação científica que consistiu em desenvolver um sistema para realizar a transmissão dos dados coletados no sistema sem a necessidade de remover o cartão de dados ou a utilização de cabos. Foi adotado como solução preliminar no projeto a tecnologia de comunicação que não faz uso de cabos e que é normalmente utilizada por computadores para acesso a redes de comunicação de dados como a Internet. Essa tecnologia denominada WiFi (*Wireless Fidelity*) utiliza rádio frequência ou infravermelho e geralmente é transmitida através de frequências de rádio na ordem de GHz, em faixas que não necessitam de licença para operação.

Em um trabalho de Iniciação Científica anterior (CAMPOS; IRITA; VILELA, 2021), foi realizado o estudo de tecnologias de transmissão sem fio utilizando o

microcontrolador Atmega. Quando foi apresentada uma solução de bancada com layout de circuito eletrônico e codificação necessários para a sua implementação. Tendo como um dos principais requisitos o fato de ser de baixo custo e que utilize componentes disponíveis no mercado nacional. A proposta inicializada por este aluno não foi devidamente concluída devida a falta de tempo hábil e a problemas relacionados a pandemia de Covid-19. Necessitando, portanto, ainda ser finalizada para ser colocada em operação de teste em campo.

Assim, este trabalho teve como objetivo principal dar continuidade ao desenvolvimento do sistema. Realizar uma série de testes em laboratório e em campo com o dispositivo com o módulo Wi-Fi acoplado ao sistema de aquisição de dados previamente desenvolvido, incluindo a codificação e a instalação dele em um experimento em campo para coleta de dados ambientais. Porém, o projeto foi interrompido após 2 meses de seu início. Portanto, neste relatório são apresentadas apenas as atividades iniciais do trabalho, que consistiu basicamente no estudo da literatura e análise e testes de funcionalidade dos módulos que já se encontravam desenvolvidos.

1.1. Objetivos Gerais Inicialmente Proposto

Este trabalho teve como objetivo principal dar continuidade ao trabalho anterior realizando testes nos dispositivos desenvolvidos e concluir a integração do módulo Wi-Fi com o módulo de aquisição de dados existente.

1.2. Objetivos Específicos Inicialmente Propostos

- Desenvolver a placa de circuito impresso para integração dos sistemas;
- Estabelecer a codificação necessário para a integração entre os módulos;
- Aplicar uma série de testes nos dispositivos visando a sua validação.
- Instalação do sistema de aquisição de dados em campo para coleta de dados ambientais em fase experimental.

1.3. Objetivos atingidos no período 2 meses

Estudo dos conceitos e dos módulos já desenvolvidos, e testes funcionais dos módulos que compõe o dispositivo proposto.

2. INTRODUÇÃO TEÓRICA

Neste item do relatório é apresentado de forma resumida o conhecimento teórico adquirido através da literatura indicada pelos orientadores. Foram estudados artigos científicos, manuais, livros, dissertações e relatórios de alunos de IC sobre trabalhos anteriores. O conhecimento teórico sobre o assunto foi de fundamental importância para a compreensão do projeto. São apresentados a seguir alguns conceitos sobre plataforma de coleta de dados ambientais; o conceito da transmissão sem fio; princípios de funcionamento dos microcontroladores; plataforma de desenvolvimento Arduino e o módulo de transmissão dados sem fio ESP -01 escolhido para esse projeto.

2.1. Plataforma de Coleta de Dados

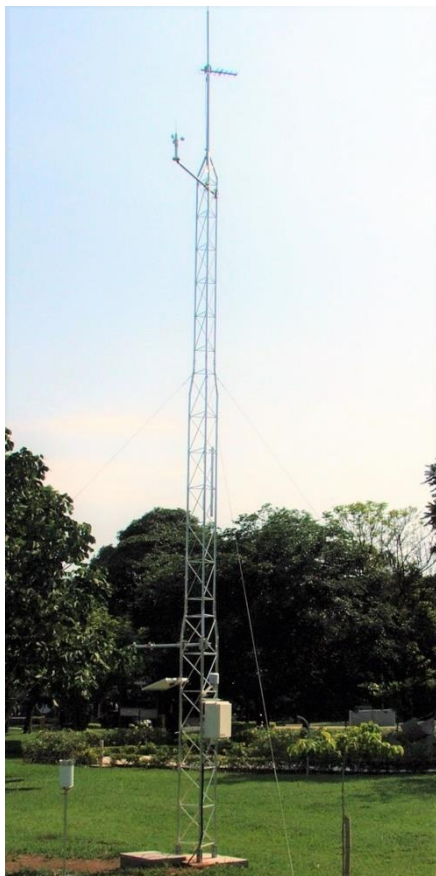
As Plataformas de Coleta de Dados (PCD) são sistemas automáticos de medição, armazenamento e transmissão de dados ambientais. Normalmente são compostas por diversos tipos de sensores que medem as grandezas ambientais como: velocidade do vento, umidade do ar, pressão atmosférica, temperatura, irradiação solar, chuva etc. O INPE utiliza um grande número de PCDs para gerar dados que são utilizados em modelos matemáticos de previsão do tempo e mudanças climáticas.

As PCDs podem ser de diversos tipos e para diversas finalidades. As mais comuns são as plataformas automáticas terrestres que são utilizadas na coleta de dados meteorológicos utilizados na previsão do tempo. Existem também PCDs mais simples que geram dados ambientais para aplicação na agricultura. Um tipo que vem se tornando muito comum são as estações solarimétricas, que geram dados para aplicação em modelos para previsão de potencial solar fotovoltaico ou potencial solar térmico. Elas também podem ser utilizadas para determinar o potencial eólico em determinadas regiões.

As PCDs comuns geralmente utilizam torres metálicas de 10 metros, mas em aplicações onde se deseja medir a velocidade do vento em determinadas alturas ou dentro de florestas as torres podem ter mais de cem metros. Nas torres são instalados os sensores ambientais, antena para transmissão de dados, para raios, painel solar para o caso de PCDs autônomas e uma caixa ambiental onde é alocado o sistema de aquisição de dados, bateria e outros dispositivos eletrônicos. Uma unidade de coleta de dados geralmente tem custo elevado, porque utiliza sensores e sistema de aquisição de dados de alta precisão e qualidade que normalmente são importados. Na Figura 1 é mostrado duas PCDs que

foram instaladas no câmpus do INPE em São José dos Campos para coleta de dados ambientais. Em (A) pode se ver uma torre de 10 metros de altura com sensores ambientais, uma caixa ambiental onde foi instalado o sistema de aquisição automática de dados e a antena do rádio utilizado para a transmissão de dados, esta PCD foi instalada próxima a portaria principal do INPE. Em (B) pode-se observar uma pequena torre de três metros da altura utilizada em experimentos de testes de sensores ambientais, essa PCD foi instalada próxima ao Laboratório de Energia Solar.

Figura 1: Exemplos de PCDs instaladas no INPE



(A)
PCD no câmpus do INPE
São José dos Campos - SP



(B)
PCD no câmpus do INPE
São José dos Campos – SP

Fonte: Arquivo de imagens do GDF

2.2. Transmissão sem fio

A transmissão sem fio consiste no envio de dados digitais por meio de ondas eletromagnéticas. O conceito de transmissão sem fio foi primeiramente introduzido pelo

físico italiano Guglielmo Marconi ao demonstrar o funcionamento do telégrafo sem fio (TANENBAUM, 2003). Desde então este sistema vem sendo aprimorado e utilizado nas mais diversas formas, desde transmissão de programação de rádio, televisão, telefonia celular até a transmissão de dados através de rede de computadores.

A estrutura de uma rede sem fio para comunicação de dados entre computadores pode variar muito dependendo da necessidade. A comunicação do tipo Wi-Fi, utilizada no desenvolvimento deste projeto, funciona através de ondas de rádio com modulação digital. Este sistema de comunicação digital utiliza um roteador que é a responsável por captar e emitir os sinais através de antenas. E é assim que os aparelhos conseguem trocar informações. Essa troca de informações pode ocorrer em duas frequências diferentes, 2.4 GHz ou 5 GHz. E quanto mais alta a frequência, maior é a capacidade de transferência de dados (OLIVEIRA, 2020). Pode-se observar um exemplo de conexão Wi-Fi na Figura 2.

Figura 2: Exemplo de conexão com Wi-Fi



Fonte: Adaptado de ALBUQUERQUE (2017).

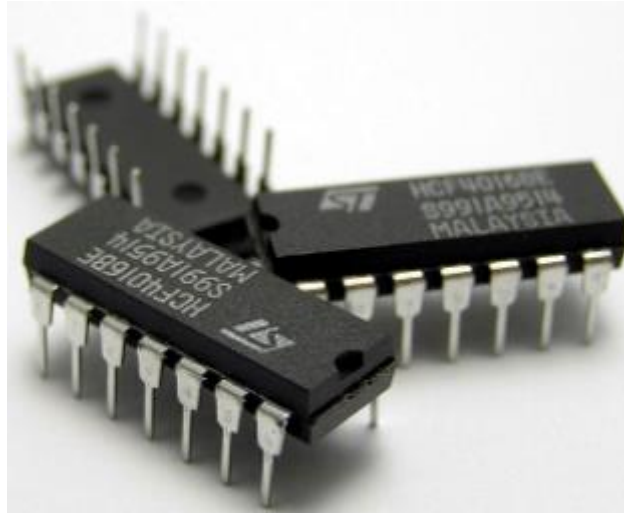
2.3. Microcontroladores

Os microcontroladores podem ser classificados como o cérebro de um dispositivo eletrônico de processamento de dados (SILBERSCHATZ et al., 2015). Eles são componentes programáveis com diversas funcionalidades e aplicações, e foram criados para realizar funções específicas (ELETRÔNICA PROGRESSIVA, 2014).

Os microcontroladores também são utilizados em plataforma de prototipagem. Técnica utilizada para a pesquisa e desenvolvimento de novas tecnologia na fabricação de circuitos eletrônicos, na qual um modelo é criado com a sucessiva colocação de outros circuitos previamente desenvolvidos. Sendo assim, é possível transformar uma

modelagem em algo concreto, de uma forma prática e rápida. Dessa forma podemos compará-los com circuitos integrados (CI), que foram desenvolvidos com funções únicas e específicas. Porém os microcontroladores são uma mistura de diversos circuitos integrados, com diversas funcionalidades sendo realizadas em um mesmo componente. Podemos observar um exemplo de microcontrolador na Figura 3.

Figura 3 - Exemplo de microcontrolador



Fonte: Adaptado de Eletrônica Progressiva (2014)

2.3.1. Arduino

O Arduino UNO, é uma plataforma básica prototipagem com microcontrolador utilizado no desenvolvimento desse projeto. Ele possui quatorze portas digitais que podem ser utilizadas tanto para entrada como saída, sendo seis providas da técnica de modulação PWM (Pulse Width Modulation), seis entradas analógicas e 32 KB de memória flash. Nesse projeto ele é o principal responsável por receber os dados dos sensores e repassá-los para o ESP-01, que por sua vez, enviará os dados para a rede de Internet. Na Figura 4 podemos observar a imagem de um Arduino UNO.

Figura 4: Arduino UNO



Fonte: POLASTRINI (2016).

2.3.2. ESP-01

O ESP-01 se trata de um dispositivo de tamanho reduzido, com poucas entradas para GPIO sendo geralmente utilizado em paralelo com um Arduino ou um ATmega 328 *standalone* através da comunicação serial. O ESP8266 possibilita ao dispositivo ao qual está embarcado se conectar à uma rede Wi-Fi padrão e trocar informações com qualquer outro dispositivo conectado nessa mesma rede já que tem a capacidade de funcionar como um servidor *Web* que executa o protocolo HTTP, protocolo que permite a troca de arquivos de texto entre dispositivos via Internet. Na Figura 5 é possível observação o ESP-01.

Figura 5: ESP-01



Fonte: FilipeFlop (2021).

3. ATIVIDADES DESENVOLVIDAS

Foi realizado o levantamento das características e modos de trabalho e os testes de funcionamento dos seguintes módulos:

- Módulo cartão de memória SD;

O módulo de cartão de memória SD com Arduino permite a gravação dos dados coletados do sistema de aquisição de dados no cartão SD na forma de tabelas.

Basicamente, o módulo possui uma entrada para cartões micro SD com todos os 6 pinos de interface disponibilizados para comunicação com Arduino. Por meio dos pinos SPI (protocolo de comunicação usado para ler e escrever dados na memória *Flash* do cartão), podemos ligar o Arduino e enviar informações para serem armazenadas. Também é possível fazer procedimentos de leitura para apresentar os dados gravados em mostradores como *displays* ou na interface serial.

A interface SPI que permite ler e gravar dados na memória *Flash* é baseada em 6 canais, como apresentado na Figura 6:

Figura 6: Portas de comunicação do módulo cartão de memória SD

Pino	Descrição	Outros nomes
CS	Slave Select	SS, nCS, nSS
SCK	System Clock	SCLK, CLK
MOSI	Master Our Slave In	SDO, DO, SO
MISO	Master In Slave Out	SDI, DI, SI
VCC	Alimentação positiva	
GND	Terra	

Fonte: VIDAL (2017).

As especificações do módulo são as seguintes:

Comunicação	SPI
Leitura e Gravação	FAT32
Tensão de Alimentação	4,5V-5V
Dimensões (CxLxA)	42x24x3,5mm

Fonte: Adaptado de VIDAL (2017).

- Módulo de referência de tempo DS3231

O Real Time Clock, também conhecido como RTC, são pequenos dispositivos eletrônicos geradores de registro de valores de data e hora no decorrer do tempo, ele funciona como um relógio em que o usuário indica os valores iniciais de data e hora, para que então, a partir desses possa incorporar dados de data e hora no funcionamento dos dispositivos eletrônicos. Os RTCs utilizam cristais de quartzo e circuitos osciladores com frequência fixa e alto grau de estabilidade. Eles são integrados em um único chip para gerar base de tempo. Praticamente todos os dispositivos que necessitam de controle preciso do tempo possuem algum tipo de RTC.

O DS3231 é um RTC de alta precisão e baixo consumo de energia. O módulo possui um sensor de temperatura de fábrica e um oscilador para melhorar ainda mais a sua exatidão.

Funciona tanto no formato 12 horas como 24 horas, e as informações de meses com menos de 30/31 dias e anos bissextos são corrigidos automaticamente pelo módulo.

Uma bateria acompanha o módulo para evitar a perda de dados em caso de desenergização do circuito. Um aspecto importante e que diferencia dos outros modelos é o protocolo de comunicação utilizado. No caso do DS3231 o protocolo I2C é usado. Assim, são dois canais(pinos) de comunicação: SCL(*Clock*) e SDA(dados).

As especificações do módulo são as seguintes:

Chip de memória	AT24C32 de 32K bytes
Sensor de temperatura	$\pm 3^{\circ}\text{C}$ de exatidão
Tensão de Alimentação	3,3V-5V
Consumo	<500nA
Dimensões (CxLxA)	38 x 22 x 14mm

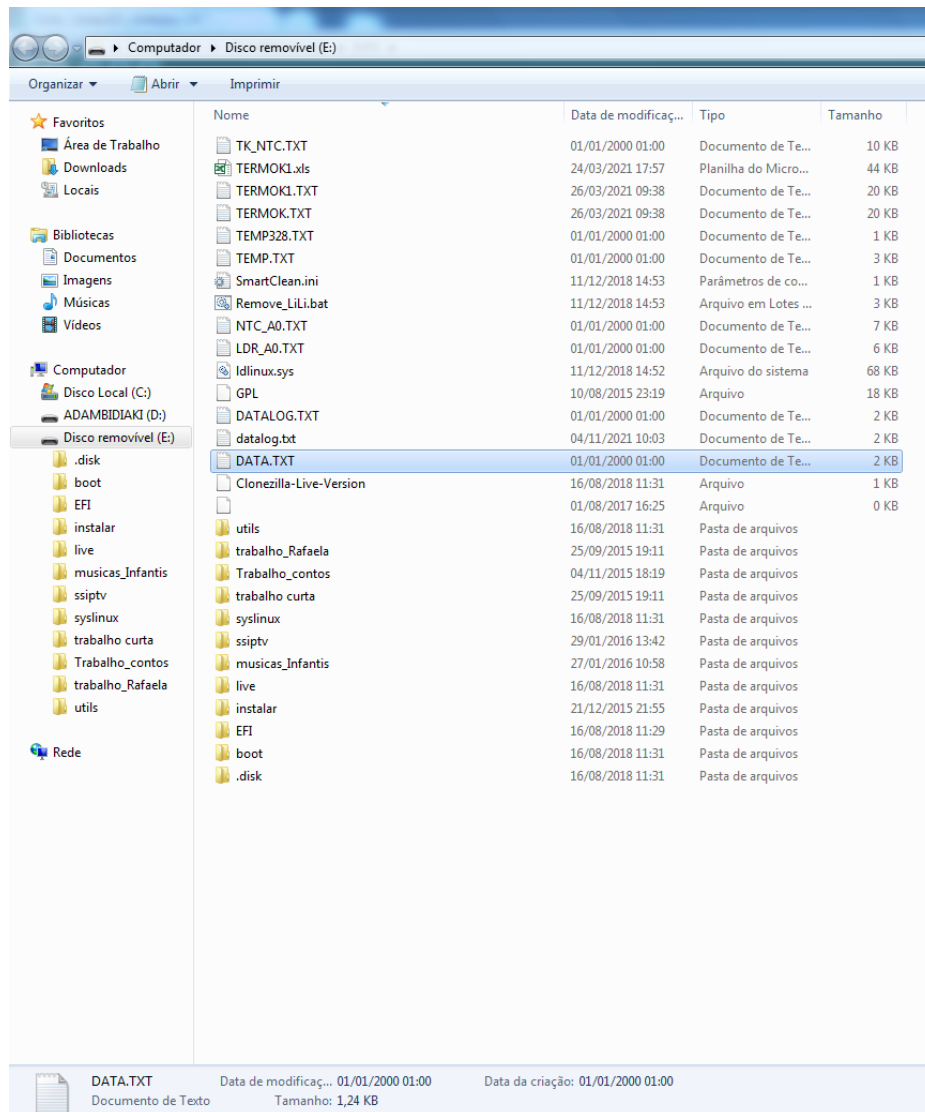
Fonte: Adaptado de VIDAL (2017).

3.1. Módulo leitor de cartão SD

O módulo leitor de cartão SD, foi um componente essencial no desenvolvimento desse projeto, uma vez que ele permite que os dados gerados pelos sensores sejam armazenados do cartão SD. Que é um tipo de memória não volátil, ou seja, mesmo se não tiver energia, manterá as informações salvas nela. Sendo assim, uma programação de teste foi desenvolvida, para que fosse possível compreender seu funcionamento.

No ANEXO I é possível observar a programação teste, onde o objetivo é gravar os segundos que o Arduino está ligado no arquivo “DATA.TXT”. A Figura 7 e a Figura 8 apresentam as mensagens salvas no arquivo.

Figura 7: Arquivo DATA.TXT



Fonte: Autor (2021).

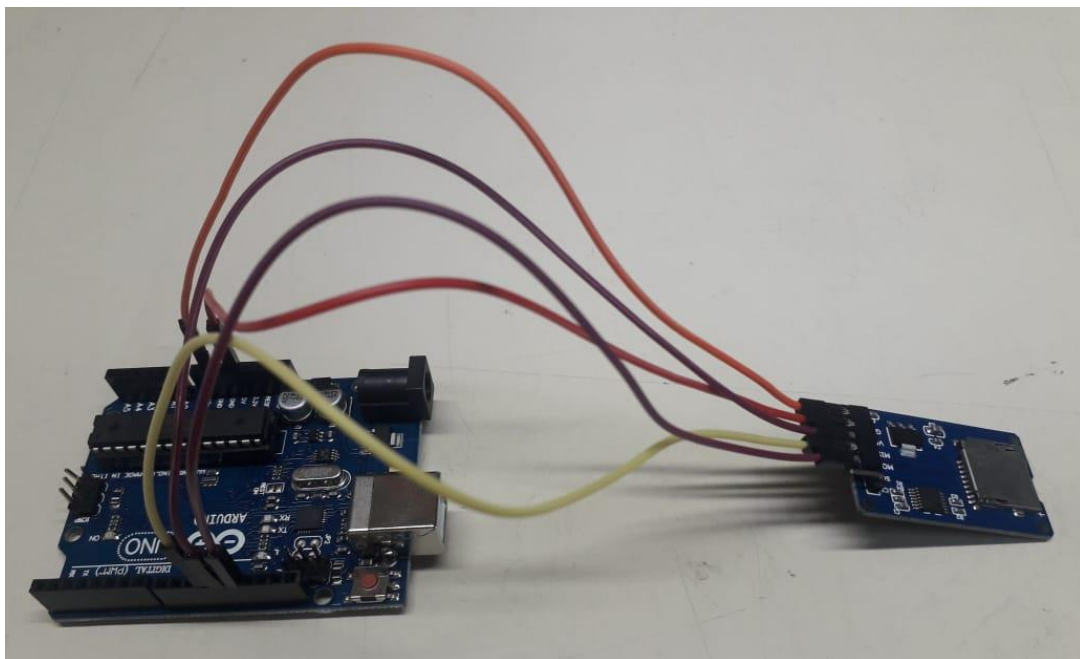
Figura 8: Mensagem salvas no arquivo DATA.TXT

```
DATA.TXT - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda
Este arduino esta ligado a 0 segundos
Este arduino esta ligado a 2 segundos
Este arduino esta ligado a 0 segundos
Este arduino esta ligado a 2 segundos
Este arduino esta ligado a 4 segundos
Este arduino esta ligado a 6 segundos
Este arduino esta ligado a 8 segundos
Este arduino esta ligado a 10 segundos
Este arduino esta ligado a 12 segundos
Este arduino esta ligado a 14 segundos
Este arduino esta ligado a 16 segundos
Este arduino esta ligado a 18 segundos
Este arduino esta ligado a 20 segundos
Este arduino esta ligado a 22 segundos
Este arduino esta ligado a 24 segundos
Este arduino esta ligado a 26 segundos
Este arduino esta ligado a 28 segundos
Este arduino esta ligado a 30 segundos
Este arduino esta ligado a 32 segundos
Este arduino esta ligado a 34 segundos
Este arduino esta ligado a 36 segundos
Este arduino esta ligado a 38 segundos
Este arduino esta ligado a 40 segundos
Este arduino esta ligado a 42 segundos
Este arduino esta ligado a 44 segundos
Este arduino esta ligado a 46 segundos
Este arduino esta ligado a 48 segundos
Este arduino esta ligado a 50 segundos
Este arduino esta ligado a 52 segundos
Este arduino esta ligado a 54 segundos
Este arduino esta ligado a 56 segundos
Este arduino esta ligado a 58 segundos
```

Fonte: Autor (2021).

Já a montagem física desse módulo pode ser observada na Figura 9.

Figura 9: Montagem física do módulo leitor de cartão SD



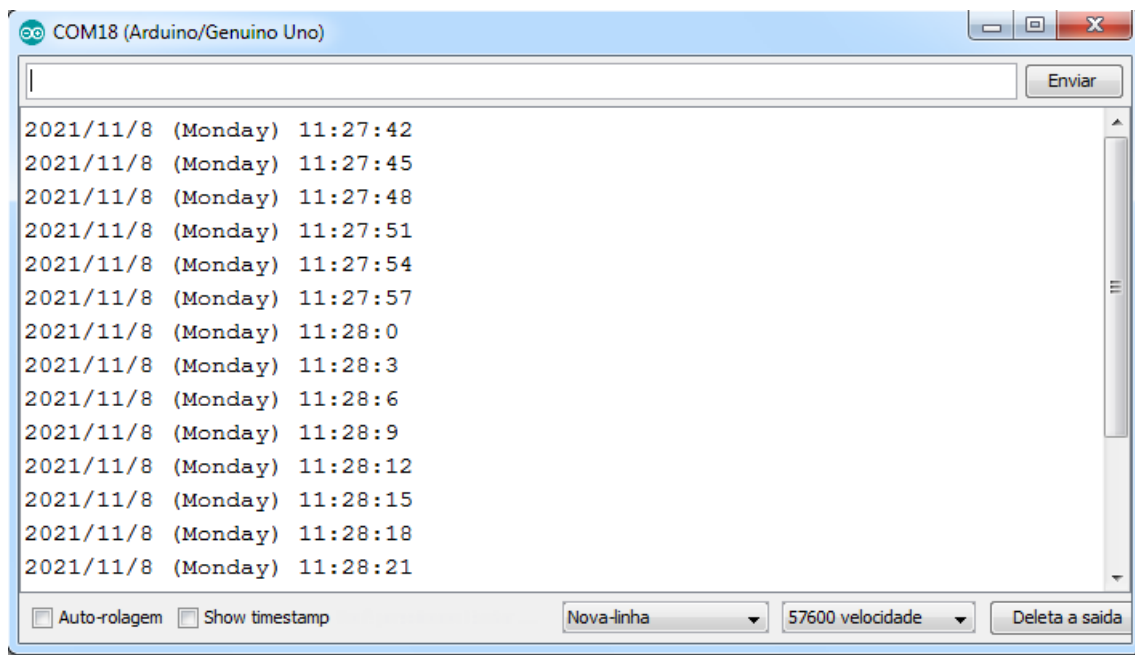
Fonte: Autor (2021).

3.2. Módulo RTC

O módulo RTC, é outro componente essencial para o desenvolvimento deste projeto, já que para analisar com eficiência os dados coletados pela PCD, informações como data e hora são essenciais. Dessa forma, uma programação de teste também foi desenvolvida, para compreender o funcionamento do módulo RTC.

No ANEXO II é possível observar a programação teste, onde o objetivo é apresentar no monitor serial da IDE do Arduino as informações de data e hora como apresentado na Figura 10.

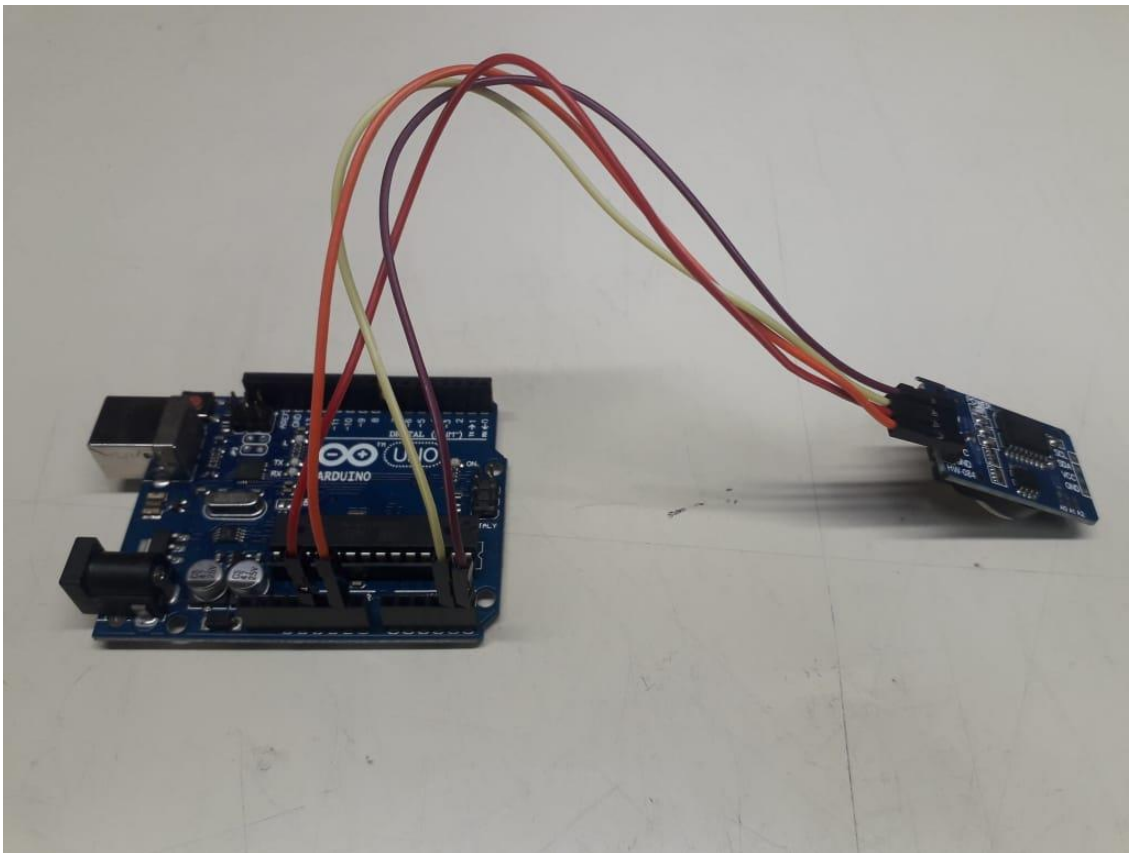
Figura 10: Informações de data e hora geradas pela programação teste do módulo RTC



Fonte: Autor (2021).

Já a montagem física desse módulo pode ser observada na Figura 11Figura 9.

Figura 11: Montagem física do módulo RTC



Fonte: Autor (2021).

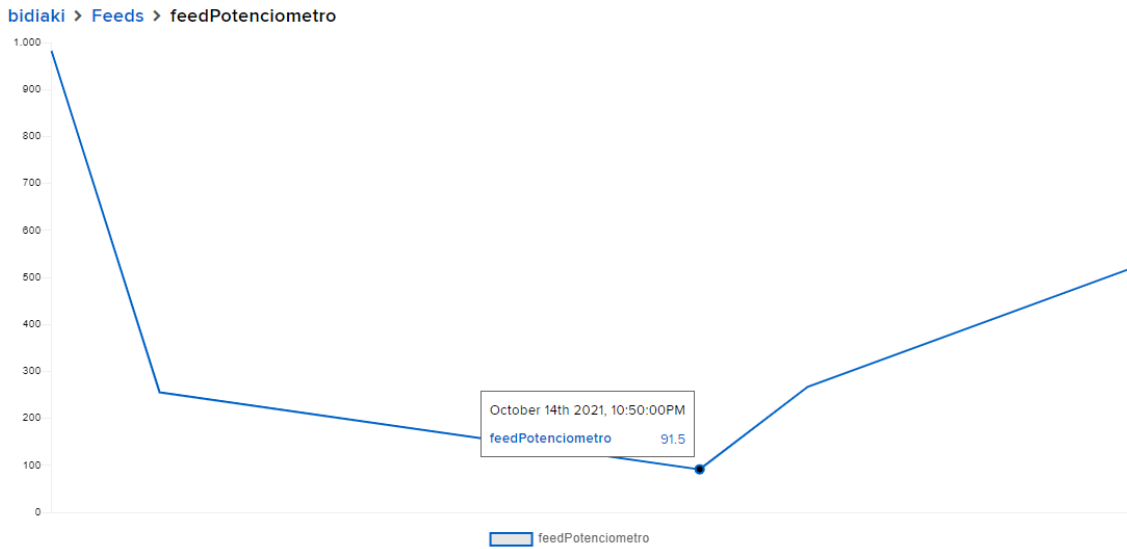
3.3. ESP-01

Provavelmente o ESP-01 junto com o Arduino UNO são os componentes mais importantes para o desenvolvimento do projeto, uma vez que o ESP-01 é o responsável por enviar os dados para a internet, via WiFi, e o Arduino é o responsável pela coleta de dados dos sensores e demais componentes. Sendo assim, uma programação teste também foi desenvolvida para a comunicação entre o ESP-01 e o Arduino. O objetivo desse algoritmo era enviar os dados coletados de um potenciômetro para que um site exibisse em tempo real as informações do componente. O ANEXO III apresenta a programação desenvolvida para a comunicação do ESP-01 com o WiFi e o Arduino.

Agora o ANEXO IV apresenta a programação desenvolvida para a comunicação do Arduino com o WiFi.

E por fim, na Figura 12 é possível observar os dados do potenciômetro que foram coletados pelo Arduino e enviado via WiFi pelo ESP-01 para o site de exibição em tempo real.

Figura 12: Dados do potenciômetro enviados via WiFi pelo ESP-01



Fonte: Autor (2021).

4. CONCLUSÃO

O desenvolvimento uma unidade de coleta de dados ambientais de baixo custo que permita o acesso remoto através de uma conexão sem fio para o envio de dados à um computador remoto é um projeto de grande interesse para os pesquisadores do GDF, visto que este tipo de dispositivo viabiliza diversas pesquisas sem a necessidade de depender de grandes recursos financeiros. Os sistemas de aquisição de dados ambientais comerciais são importados e possuem custos elevados que muitas vezes não se justifica a compra em pesquisas de pequeno porte.

Embora as atividades presentes nesse relatório não evidenciam uma evolução significativa ao projeto, as atividades mostraram que é possível coletar dados ambientais e enviá-los para um site externo via WiFi, o que é um dos principais objetivos desta pesquisa.

A pesquisa bibliográfica e análises das programações desenvolvidas permitiram acumular informações importantes que irão nortear a continuação do trabalho.

Por fim, como proposta de trabalho futuro, é preciso unir em um único programa todas as programações que foram desenvolvidas para módulos específicos, também é preciso finalizar o desenvolvimento da placa de circuito impresso, realizar a montagem dos componentes eletrônicos e iniciar os testes em laboratório e em campo.

REFERÊNCIAS

ALBUQUERQUE, Igor. Estação meteorológica com Arduino. Embarcados, jun. 2016. Disponível em: . Acesso em: 15 de novembro de 2021.

CAMPOS, A. M.; IRITA, R. T.; VILELA, W. A. Aprimoramento de uma plataforma de coleta de dados ambientais com dispositivos de transmissão sem fio. In: SEMINÁRIO DE INICIAÇÃO CIENTÍFICA E INICIAÇÃO EM DESENVOLVIMENTO TECNOLÓGICO E INOVAÇÃO (SICINPE), online. Resumos... São José dos Campos: INPE, 2021. p. 1. On-line. Bolsa PIBIC/PIBITI/INPE/CNPq. IBI: <8JMKD3MGP3W34T/45LA2TS>. Disponível em: <<http://urlib.net/ibi/8JMKD3MGP3W34T/45LA2TS>>.

Eletrônica Progressiva. **Microcontroladores - O que são, Para Que Servem e Onde São Usados.** Disponível em: <<https://www.eletronicaprogressiva.net/2014/08/Microcontroladores-O-que-sao-Para-que-servem-Onde-sao-usados.html>>. Acesso em: 17 de novembro de 2021.

OLIVEIRA, C. **Como funciona o WiFi.** Disponível em: <<https://www.embarcados.com.br/como-funciona-o-wifi/>>. Acesso em: 16 de novembro de 2021.

POLASTRINI, Fernando. **Desenvolvimento de uma máquina CNC de baixo custo com software e hardware abertos.** Disponível em: <https://www.formiga.ifmg.edu.br/documents/2017/PublicacoesTCCsBiblioteca/EE/TC_C_FINAL_FERNANDOPOLASTRINI_2016_EE-.pdf>. Acesso em: 13 de novembro de 2021.

SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE, G. **Fundamentos de sistemas operacionais.** 9 ed. [S.l.]: LTC, 2015. ISBN 978-85-216-3000-5.

TANENBAUM, A. S. **Redes de Computadores.** 4 ed. Rio de Janeiro: Editora Campus, 2003. p. 23.

ANEXO I

Teste_CartaoSD §

```
1 //Programa: Teste modulo cartao micro SD Arduino
2
3 #include <SPI.h>
4 #include <SD.h>
5
6 //Pino de conexão do pino CS do modulo
7 const int chipSelect = 4;
8
9 void setup()
10 {
11   Serial.begin(9600);
12
13   Serial.println("Teste de Modulo cartao micro SD");
14   Serial.println("Inicializando cartao SD...");
15
16   //Inicia a comunicacao com o modulo SD
17   if (!SD.begin(chipSelect))
18   {
19     Serial.println("Falha ao acessar o cartao !");
20     Serial.println("Verifique o cartao/conexoes e reinicie o Arduino...");
21     return;
22   }
23   Serial.println("Cartao iniciado corretamente !");
24   Serial.println();
25 }
26
27 void loop()
28 {
29   unsigned long currentMillis = millis();
```

```
30
31 //Mostra os dados no Serial Monitor
32 Serial.print("Este arduino esta ligado a ");
33 Serial.print(currentMillis / 1000);
34 Serial.println(" segundos");
35
36 //Abre o arquivo datalog.txt
37 File dataFile = SD.open("datalog.txt", FILE_WRITE);
38
39 //Grava as informacoes no arquivo
40 if (dataFile)
41 {
42     dataFile.print("Este arduino esta ligado a ");
43     dataFile.print(currentMillis / 1000);
44     dataFile.println(" segundos");
45
46     //Fecha o arquivo
47     dataFile.close();
48 }
49
50 //Exibe um erro se nao conseguir abrir o arquivo
51 else
52 {
53     Serial.println("Erro ao abrir o arquivo datalog.txt");
54 }
55 delay(2000);
56 }
```


ANEXO II

```
Teste_RTC $
1 // Date and time functions using a DS3231 RTC connected via I2C and Wire lib
2 #include "RTClib.h"
3
4 RTC_DS3231 rtc;
5
6 char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};
7
8 void setup () {
9     Serial.begin(57600);
10
11 #ifndef ESP8266
12     while (!Serial); // wait for serial port to connect. Needed for native USB
13 #endif
14
15     if (!rtc.begin()) {
16         Serial.println("Couldn't find RTC");
17         Serial.flush();
18         abort();
19     }
20
21     if (rtc.lostPower()) {
22         Serial.println("RTC lost power, let's set the time!");
23         // When time needs to be set on a new device, or after a power loss, the
24         // following line sets the RTC to the date & time this sketch was compiled
25         rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
26         // This line sets the RTC with an explicit date & time, for example to set
27         // January 21, 2014 at 3am you would call:
28         // rtc.adjust(DateTime(2021, 11, 05, 15, 20, 00));
29     }
30
31     // When time needs to be re-set on a previously configured device, the
32     // following line sets the RTC to the date & time this sketch was compiled
33     // rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
34     // This line sets the RTC with an explicit date & time, for example to set
35     // January 21, 2014 at 3am you would call:
36     // rtc.adjust(DateTime(2021, 11, 05, 15, 58, 0));
37 }
38
39 void loop () {
40     DateTime now = rtc.now();
41
42     Serial.print(now.year(), DEC);
43     Serial.print('/');
44     Serial.print(now.month(), DEC);
45     Serial.print('/');
46     Serial.print(now.day(), DEC);
47     Serial.print(" ");
48     Serial.print(daysOfTheWeek[now.dayOfTheWeek()]);
49     Serial.print(" ");
50     Serial.print(now.hour(), DEC);
51     Serial.print(':');
52     Serial.print(now.minute(), DEC);
53     Serial.print(':');
54     Serial.print(now.second(), DEC);
55     Serial.println();
56
57     delay(3000);
58 }
```

ANEXO III

```
Teste_ESP-01 § config.h
1 /*
2     CÓDIGO: Q0612-Master
3     AUTOR: BrincandoComIdeias
4     LINK: https://www.youtube.com/brincandocomideias ; https://cursodearduino.net/ ; https://cursoderobotica.net
5     COMPRE: https://www.arducore.com.br/
6     SKETCH: AdafruitIO com ESP e Uno - Master
7     DATA: 07/02/2020
8 */
9
10 // INCLUSÃO DE BIBLIOTECAS
11 #include <A2a.h>
12 #include "config.h"
13
14 // DEFINIÇÃO DO ENDEREÇO DO SLAVE
15 #define endereco 0x08 // ENDEREÇO HEXADECIMAL
16
17 // DEFINIÇÕES
18 #define tempoAtualizacao 15000
19 #define LIGA LOW
20 #define DESLIGA HIGH
21
22 // INSTANCIANDO OBJETOS
23 A2a arduinoSlave;
24
25 AdafruitIO_Feed *feedPotenciometro = io.feed("feedPotenciometro");
26
27 // DECLARAÇÃO DE FUNÇÕES
28 void configuraMQTT();
29 bool monitoraPotenciometro();
30
31 // DECLARAÇÃO DE VARIÁVEIS
32 unsigned long controleTempo = 0;
33 bool comandoRecebido = false;
34
35 float _valorPotenciometro;
36
37 void setup() {
38     // INICIA A COMUNICAÇÃO SERIAL
39     Serial.begin(9600);
40     while (! Serial);
41
42     // INICIA A COMUNICAÇÃO ENTRE ARDUINOS COMO SLAVE NO ENDEREÇO DEFINIDO
43     arduinoSlave.begin(0, 2);
44
45     configuraMQTT();
46
47     io.run();
48
49     Serial.println("Fim setup()");
50 }
51
52 void loop() {
53     io.run();
54
55     if (millis() > controleTempo + tempoAtualizacao) {
56
57         if ( monitoraPotenciometro() ) {
```

```

58     controleTempo = millis();
59     feedPotenciometro->save(_valorPotenciometro);
60 }
61 }
62 }
63
64 // IMPLEMENTO DE FUNÇÕES
65 void configuraMQTT() {
66     Serial.print("Conectando ao Adafruit IO");
67     io.connect();
68
69     while (io.status() < AIO_CONNECTED) {
70         Serial.print(".");
71         delay(500);
72     }
73
74     Serial.println();
75     Serial.println(io.statusText());
76 }
77
78 bool monitoraPotenciometro() {
79     static int leituraAnt;
80
81     byte byte1 = arduinoSlave.varWireRead(endereco, 0);
82     byte byte2 = arduinoSlave.varWireRead(endereco, 1);
83
84     unsigned int leitura = byte1 << 8 | byte2;
85
86     if (leitura != leituraAnt) {
87
88         _valorPotenciometro = ( float(leitura) );
89
90         Serial.print("Pot: ");
91         Serial.println(_valorPotenciometro);
92
93         leituraAnt = leitura;
94         return true;
95     } else {
96         return false;
97     }
98 }

```

Teste_ESP-01 §

config.h

```

1  |/***** Adafruit IO Config *****/
2
3  // visit io.adafruit.com if you need to create an account,
4  // or if you need your Adafruit IO key.
5  #define IO_USERNAME    "bidiaki"
6  #define IO_KEY         "aio_veus641Dg1H1CwvUMnqW0ZAz0T9F"
7
8  /***** WIFI *****/
9
10 #define WIFI_SSID     "Bik 2G"
11 #define WIFI_PASS     "yourgul23"
12
13 #include "AdafruitIO_WiFi.h"
14
15 AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS);
16
17 /*****

```

ANEXO IV

```
Teste_Arduino
1  /*
2   CÓDIGO: Q0612-Slave
3   AUTOR: BrincandoComIdeias
4   LINK: https://www.youtube.com/brincandocomideias ; https://cursodearduino.net/ ; https://cursoderobotica.net
5   COMPRE: https://www.arduocore.com.br/
6   SKETCH: AdafruitIO com ESP e Uno - Slave
7   DATA: 07/02/2020
8  */
9
10 // INCLUSÃO DE BIBLIOTECAS
11 #include <A2a.h>
12
13 // DEFINIÇÃO DO ENDEREÇO DO SLAVE
14 #define endereco 0x08 // ENDEREÇO HEXADECIMAL
15 #define potenciometro A0
16
17 // INSTANCIANDO OBJETOS
18 A2a arduinoMaster;
19
20 // DECLARAÇÃO DE VARIÁVEIS
21 int valorPotenciometro;
22
23 void setup() {
24   // INICIA A COMUNICAÇÃO SERIAL
25   Serial.begin(9600);
26
27   // INICIA A COMUNICAÇÃO ENTRE ARDUINOS COMO SLAVE NO ENDEREÇO DEFINIDO
28   arduinoMaster.begin(endereco);
29
30   // FUNÇÕES PARA COMUNICAÇÃO
31   arduinoMaster.onReceive(receberDados);
32   arduinoMaster.onRequest(enviarDados);
33
34   pinMode (potenciometro, INPUT);
35
36   Serial.println("Fim setup()");
37 }
38
39 void loop() {
40   valorPotenciometro = analogRead (potenciometro);
41   //umidade = int( 10*(sensorDHT.readHumidity()) );
42
43   arduinoMaster.wireWrite(0,highByte(valorPotenciometro));
44   arduinoMaster.wireWrite(1,lowByte(valorPotenciometro));
45
46   Serial.println(valorPotenciometro);
47 }
48
49 void receberDados() {
50   arduinoMaster.receiveData();
51 }
52
53 void enviarDados() {
54   arduinoMaster.sendData();
55 }
```