

SISTEMA DE MONITORAMENTO REMOTO DO MÓDULO GPRS COM CONFECÇÃO DE PCI EM CNC ROUTER

GPRS MODULE REMOTE MONITORING WITH PCB MANUFACTURING IN CNC ROUTER

Christian Araujo de Souza

christian.araujo96@outlook.com

Ygo Neto Batista

ygo@pesqueira.ifpe.edu.br

RESUMO

Este trabalho apresenta o desenvolvimento de um sistema de configuração e monitoramento de um módulo de comunicação GPRS e sistema de *reset*, inserido no projeto mais abrangente “Interface GPRS para *datalogger*” da equipe Oficinas 4.0 do IFPE *Campus* Pesqueira. Este projeto tem como objetivo aprimorar a eficiência dos sistemas de aquisição de dados em estações solarimétricas. O sistema de configuração e monitoramento do módulo GPRS visa permitir a operação remota do módulo GPRS, enviando comandos AT para o módulo GPRS SIM900 e tratando as informações retornadas pelo módulo. Os comandos podem ser informados pelo usuário via SMS, terminal serial do computador ou via dispositivo móvel conectado em módulo Bluetooth. O sistema de *reset* visa solucionar problemas de operação do módulo GPRS e é composto por uma placa com relé, desenvolvida neste trabalho, conectada ao microcontrolador e ao módulo SIM900. A manufatura da Placa de Circuito Impresso (PCI) foi realizada na CNC router do LabMaker no IFPE *Campus* Pesqueira. Em paralelo a este trabalho, outras equipes estão desenvolvendo o algoritmo para armazenamento e verificação das informações do *datalogger* em servidor FTP remoto, o desenvolvimento dos suportes mecânicos em Impressora 3D, estudo de viabilidade técnico e econômico do projeto, estudos para registro da propriedade intelectual, entre outras atividades para a conclusão da Interface GPRS.

Palavras-chave: Sistema de monitoramento remoto. GPRS. Estações solarimétricas. Fabricação de PCI.

ABSTRACT

This work presents the development of a configuration and monitoring system for a GPRS communication module and reset system, inserted in the more comprehensive project “GPRS Interface for datalogger” of the Oficinas 4.0 team at IFPE *Campus* Pesqueira. This project aims to improve the efficiency of data acquisition systems in solarimetric stations. The GPRS module configuration and monitoring system aims to allow

remote operation of the GPRS module, sending AT commands to the GPRS SIM900 module and processing the information returned by the module. Commands can be entered by the user via SMS, computer serial terminal or via mobile device connected to a Bluetooth module. The reset system aims to solve GPRS module operation problems and is composed of a relay board, developed in this work, connected to the microcontroller and the SIM900 module. The manufacturing of the Printed Circuit Board (PCB) was carried out on the LabMaker CNC router at IFPE Campus Pesqueira. In parallel to this work, other teams are developing the algorithm for storing and verifying datalogger information on a remote FTP server, the development of mechanical supports in 3D Printers, technical and economic feasibility studies for the project, studies to register intellectual property, among other activities to complete the GPRS Interface.

Keywords: Remote monitoring system. GPRS. Solarimetric stations. PCB manufacturing.

1 INTRODUÇÃO

O IFPE *Campus* Pesqueira se destaca como um laboratório de referência em energia solar fotovoltaica, inserido na Rede Federal de Educação Profissional, Científica e Tecnológica. A conquista do *status* de laboratório de referência marca um grande avanço do instituto no campo da energia solar fotovoltaica. Os projetos relacionados às estações solarimétricas e ao laboratório de energias renováveis no *Campus* Pesqueira têm gerado colaborações em pesquisa e desenvolvimento com instituições públicas e privadas (IFPE, 2020).

Além disso, o IFPE *Campus* Pesqueira está envolvido no desenvolvimento de um mapa de irradiação solar para o estado de Pernambuco (REDESOLPE), envolvendo 75 estações solarimétricas em todo o estado e algumas áreas limites (PEDROSA FILHO, 2022). Apesar de investimentos recentes visando a conexão remota, o projeto da REDESOLPE ainda enfrenta desafios no monitoramento da operação e dos parâmetros ambientais em algumas estações associadas. O instituto está empenhado em superar essa limitação, investindo em novos equipamentos e tecnologias para possibilitar a coleta remota dos dados, melhorando assim a eficiência e a confiabilidade da aquisição de dados. Contudo, as opções de conectividade atualmente disponíveis são ou excessivamente dispendiosas ou não atendem aos requisitos do coordenador do laboratório de referência em energia solar do IFPE.

Das 75 estações solarimétricas, 4 ainda não tem transmissão remota de dados, requerendo a presença física de um profissional no local para coletar os dados. É importante observar que o problema de conectividade não se restringe ao IFPE *Campus* Pesqueira. Estudos anteriores, como os de Priambodo e Nugroho (2021), Patil (2022) e Xiong et al. (2021) abordaram problemas na conexão entre dispositivos meteorológicos e sistemas de irrigação, e no monitoramento de subestações de energia.

Para resolver essa questão de comunicação entre os *dataloggers* e a internet, especialmente considerando que muitos locais onde esses dispositivos estão instalados não têm acesso à internet Wi-Fi (IEEE 802.11), e com baixo custo, o IFPE Campus

Pesqueira tem desenvolvido sua própria interface de conexão. Neste projeto, financiado pelo edital 83/2022 SETEC/MEC, Oficinas 4.0, o IFPE tem desenvolvido uma interface *General Packet Radio Service* (GPRS) com foco na conexão de *dataloggers* aplicados em estações solarimétricas. *Dataloggers* são componentes fundamentais em sistemas de aquisição de dados, podendo se conectar a vários sensores, executar cálculos e tarefas lógicas, converter unidades de medida, armazenar informações na memória e controlar dispositivos externos.

Sabendo que alguns dos empecilhos na transmissão remota se dão por problemas de conexão, é importante realizar o monitoramento do status da comunicação entre o sistema de coleta de dados e a internet. Dessa forma, o operador pode decidir o que fazer diante de eventuais falhas ou interrupções na transmissão, contribuindo para a eficiência e confiabilidade do sistema.

Neste trabalho foi estudada a plataforma de prototipagem a ser utilizada, onde foi escolhido um microcontrolador, no caso o Arduino Mega; foi escolhido um módulo de comunicação GPRS, no caso o SIM900; e realizada uma busca de anterioridades. Foi desenvolvido um sistema de *reset* para o módulo SIM900 para casos de falhas; foi desenvolvido um algoritmo para recepção, segmentação e armazenamento das informações do módulo GPRS; e, foi desenvolvido um procedimento para elaboração e confecção de Placas de Circuito Impressas (PCI).

2 FUNDAMENTAÇÃO TEÓRICA

Dentre as atividades desenvolvidas durante este TCC, destacam-se o estudo aprofundado da plataforma de prototipagem selecionada pela equipe Oficinas 4.0 e a análise da comunicação entre o microcontrolador e o periférico de comunicação. Essas atividades são cruciais para a compreensão dos princípios básicos dos sistemas embarcados e dos modos de comunicação do *datalogger*, bem como para estabelecer uma comunicação eficiente com o módulo de comunicação SIM900, um dispositivo de comunicação para projetos de IoT e monitoramento remoto que suporta tecnologias GSM e GPRS, possibilitando transmissão de dados, SMS e chamadas telefônicas em redes celulares.

A escolha da plataforma de prototipagem adequada é um passo crítico neste processo, levando em consideração não apenas as interfaces de comunicação, mas também outros parâmetros essenciais, como consumo energético, capacidade computacional, espaço necessário na memória e outros requisitos já minuciosamente analisados pela equipe do projeto Oficinas 4.0. O projeto Oficinas 4.0 é um programa de capacitação extracurricular para estudantes do ensino médio técnico, graduação e pós-graduação. O programa é realizado em ambientes de colaboração, utilizando métodos de aprendizado centrados em projetos e têm como foco a criação de soluções para desafios reais provenientes do setor produtivo. O programa capacita os estudantes para empreendedorismo, inovação e a aplicação das tecnologias digitais da Economia 4.0.

2.1 Sistemas Embarcados

Sistemas embarcados são sistemas computacionais especializados projetados para desempenhar funções específicas em dispositivos eletrônicos. Eles são amplamente

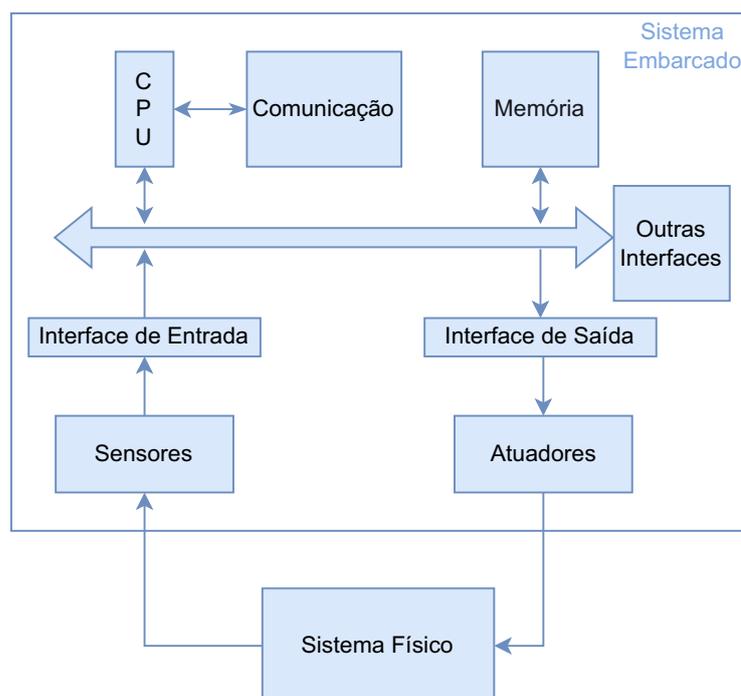
empregados em uma variedade de aplicações, desde eletrodomésticos e veículos até equipamentos médicos e sistemas de automação industrial (GARCIA, 2018).

O hardware de um sistema embarcado consiste e conecta componentes como o processador, responsável pelo processamento de dados, a memória, que armazena programas e informações, sensores, atuadores, periféricos e fontes de alimentação.

Os sensores coletam dados do ambiente, fornecendo informações para o sistema tomar decisões, enquanto os atuadores realizam ações com base nas decisões do sistema. Os periféricos podem ser necessários para realizar algumas ações específicas, como a comunicação entre sistemas embarcados ou registro de data e hora em tempo real.

Enquanto o software embarcado abrange o *firmware*, armazenado na memória do sistema e responsável pelo controle básico do dispositivo, o sistema operacional embarcado, que gerencia recursos e tarefas do sistema, e aplicativos embarcados, que executam funções específicas, como controle de dispositivos e processamento de dados. Na Figura 1 pode ser visto um diagrama esquemático de um sistema embarcado.

Figura 1 – Diagrama esquemático de um sistema embarcado



Fonte: Adaptado de Oliveira e Shin-Ting (2019)

2.2 Comunicação com o Datalogger

O *datalogger* utilizado é o CR1000, fabricado pela Campbell *Scientific*, devido à sua presença nas estações solarimétricas da REDESOLPE, onde demonstrou alta confiabilidade e resistência, tornando-o uma opção ideal para utilização em ambientes

adversos, como estações meteorológicas e solarimétricas. No entanto, é importante observar que o modelo CR1000 não possui capacidade de conexão com a internet.

A Campbell *Scientific* oferece uma opção de interface para conexão à internet (IEEE 802.11), mas o custo dessa opção é proibitivo para nossa instituição, considerando que precisaríamos de mais que uma unidade. Um exemplo desse tipo de equipamento é a interface *Ethernet* NL201 (sem Wi-Fi) que tem um custo aproximado de USD 700 nos Estados Unidos, além dos impostos. A substituição dos *dataloggers* por modelos mais recentes, que já incluem conectividade *Ethernet*, como o CR1000X (com custo aproximado de USD 4000), é ainda mais onerosa. Além disso, vale destacar que o CR1000X não possui suporte para conexão GPRS, o que implica na necessidade de instalação de um modem GPRS separadamente.

Para a comunicação com o *datalogger*, foram testadas as bibliotecas *ArduinoModbus*, *ModbusRTUMaster* e *ModbusMaster*. Não foi possível utilizar a primeira biblioteca pois ela dependia do módulo RS485. Para as demais bibliotecas, foram variados os parâmetros de *delay* entre as solicitações, endereço dos registradores e tipo dos registradores (*Holding* e *Input*). Nenhuma das bibliotecas nos atendeu, pois seu retorno era automático. Ou seja, conectado ou não, era retornado números aleatórios ou zeros.

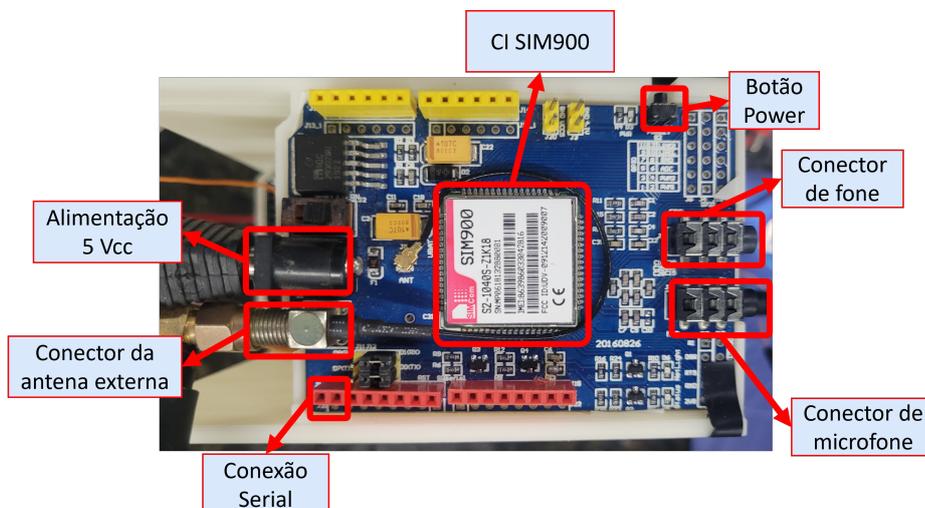
2.3 Módulo de comunicação SIM900

Fabricado pela SIMCom *Wireless Solutions*, o Circuito Integrado (CI) SIM900 é um dispositivo de comunicação que permite a transmissão de dados pela rede de telefonia móvel GSM/GPRS. Pode ser utilizado em transmissão de voz, SMS (*Short Message Service*), aplicações *Machine to Machine* (M2M) ou projetos que requerem conectividade de dados sem fio, como rastreamento de veículos, sistemas de monitoramento remoto, automação residencial e outros aplicativos de Internet das Coisas (IoT) (SILVA, 2016).

Esse dispositivo é projetado de maneira a ser facilmente integrado a circuitos eletrônicos, estabelecendo comunicação com o microcontrolador por meio da interface UART (*Universal Asynchronous Receiver/Transmitter*).

Como podemos ver na Figura 2 que não há um botão de “reset”. Neste trabalho, veremos na seção de metodologia, iremos modificar a placa na conexão do botão “Power” para executar o “reset” da placa. Observamos também a conexão serial através do conector pwb indicado na figura.

Figura 2 – Vista superior do módulo SIM900



Fonte: Autoria Própria (2023)

2.3.1 Comandos AT

A comunicação com o módulo SIM900 é realizada por meio de comandos do tipo AT, que podem ser enviados via comunicação serial. Esses comandos AT permitem configurar o módulo, fazer chamadas, enviar mensagens de texto e configurar várias opções de comunicação (OLIVEIRA, 2016).

O SIM900 AT *Command Manual* (SIMCOM, 2010) foi examinado com o propósito de compreender tanto os estados do módulo quanto os comandos AT que ele é capaz de suportar. Adicionalmente, foram conduzidas pesquisas online para reunir informações complementares. Os principais comandos AT, utilizados neste trabalho, podem ser vistos no Anexo 1

2.4 Placas de Circuito Impresso

A elaboração de Placas de Circuito Impresso (PCI) fornece suporte para componentes eletrônicos e estabelece conexões elétricas confiáveis. A fresagem CNC (Controle Numérico Computadorizado) é um dos métodos utilizados para confecção de PCIs, especialmente em prototipagem e produção de pequena escala, onde a remoção controlada de cobre de uma placa revestida é realizada para criar trilhas. Neste trabalho, utilizaremos a CNC Router para confecção de PCI, o qual é detalhado a seguir.

2.4.1 Operação da CNC Router

A necessidade de seguir os procedimentos corretos de operação da CNC Router é fundamental para garantir a segurança dos operadores, a precisão do trabalho e a durabilidade da máquina. Uma CNC Router é uma ferramenta poderosa que utiliza um sistema computadorizado para modelar, cortar e gravar materiais com extrema preci-

são. O não cumprimento dos procedimentos adequados pode resultar em acidentes graves, danos à máquina e peças de trabalho defeituosas. Além disso, a configuração e operação corretas garantem a consistência na produção e a eficiência no uso dos recursos, tornando-a uma prática indispensável em ambientes de fabricação e usinagem. Portanto, seguir rigorosamente os procedimentos estabelecidos é essencial para obter resultados satisfatórios e manter um ambiente de trabalho seguro.

A equipe do projeto Oficinas 4.0 2023 do *Campus* elaborou documentos do tipo Procedimento Operacional Padrão (POP) para a execução de tarefas e procedimentos na CNC Router do *LabMaker*. Esse tipo de documento serve para informar aos profissionais envolvidos nos processos sobre as diretrizes de atuação, mantendo assim a padronização das atividades.

Os POPs descrevem as etapas necessárias para realizar operações de usinagem CNC de forma segura e precisa. Eles incluem medidas de segurança, instruções para exportar modelos para máquinas CNC, procedimentos para gerar percursos de corte, etapas para configurar fresas e orientações para preparar peças para usinagem.

O POP 001 delinea os passos requeridos para assegurar a limpeza e lubrificação adequadas de uma máquina CNC. Essas ações são vitais para estender a durabilidade do equipamento e garantir o seu funcionamento adequado. Além disso, o documento destaca a relevância de executar tais procedimentos a cada 7 dias em caso de uso intensivo da CNC ou a cada 14 dias em cenários de uso mais moderado.

O POP 002 descreve as etapas necessárias para exportar um modelo do Fusion 360, *software* de design e modelagem 3D que integra funcionalidades de CAD, CAM e CAE, para uma máquina CNC. Essas etapas incluem preparar o modelo para garantir que ele seja processado corretamente, selecionar o formato de exportação desejado e configurar as opções de exportação.

O POP 003 aborda os fundamentos da modelagem tridimensional, incluindo tópicos como a escolha do plano de trabalho, a elaboração e a extrusão de esboços para a criação de modelos 3D.

O POP 004 detalha as etapas necessárias para gerar percursos de corte para uma máquina CNC. Essas etapas incluem preparar o modelo para garantir que ele seja processado corretamente, selecionar o tipo de percurso desejado e configurar as opções de percurso.

O POP 005 fornece instruções sobre as etapas requeridas para configurar ferramentas de corte no Fusion 360 para uma máquina CNC. Estas etapas englobam a seleção da ferramenta de corte adequada e a configuração dos parâmetros para garantir a instalação correta da ferramenta na máquina CNC.

O POP 006 retrata o processo de um auto nivelamento utilizando o bCNC, para fins de trabalho em uma placa metálica, como por exemplo uma PCI. O bCNC é um *software* de código aberto que oferece recursos como programação, simulação e operação de máquinas CNC.

O POP 007 delinea as etapas essenciais para a preparação de uma máquina CNC antes do processo de usinagem. Estas etapas abrangem a organização da mesa, garantindo que ela esteja limpa, desobstruída, nivelada e com o material devidamente fixado, além de preparar o *spindle* com a fresa e seu suporte apropriado.

Além de seguir os POPs, é importante seguir algumas dicas adicionais para garantir operações de usinagem CNC seguras e eficientes. Dentre essas dicas podemos citar o uso das ferramentas e equipamentos adequados para cada tarefa, usar equipamentos de segurança e verificar as peças e a máquina CNC após a preparação para garantir que estejam em boas condições. Ao seguir as etapas descritas nos POPs e as dicas adicionais, podemos garantir a segurança e a qualidade das operações de usinagem CNC. Os POPs podem ser acessados pelo link <https://1drv.ms/f/s!AuNzKMjUo5uVhL41wEKdaR045Le96Q>.

2.5 Busca de anterioridades e textos técnicos-científicos para interfaces de comunicação

Esse estudo fornece uma visão abrangente do estado da técnica na área, evitando o desperdício de recursos em pesquisas que já foram realizadas. Além disso, a análise das classificações de patentes existentes ajuda a direcionar o desenvolvimento de novas inovações, permitindo que os pesquisadores identifiquem em quais categorias uma possível patente poderá ser classificada. Isso é fundamental para o planejamento e a proteção eficaz de novas ideias e tecnologias.

Em uma busca preliminar por trabalhos científicos, foram identificados mais de cinco mil trabalhos relacionados ao monitoramento usando GPRS desde 2021, incluindo trabalhos mencionados anteriormente. Priambodo e Nugroho (2021) descrevem sobre o projeto de uma estação solarimétrica automática, onde os usuários podem acessar os dados através de um *website* e, assim, fazer uma análise sobre a necessidade de irrigar uma certa área. Nesse projeto, é instalado um temporizador *watchdog* para que reinicie o sistema em caso de falha de *software*. Xiong et al. (2021) relatam o projeto de um sistema de monitoramento remoto de uma subestação baseado em GPRS, onde esse sistema fornece informações sobre mau funcionamento em tempo real, assim como a solução. Já Patil (2022) apresenta um sistema de irrigação onde a comunicação GSM é utilizada para comunicar o usuário sobre os parâmetros ambientais do local monitorado.

Por outro lado, a busca por patentes concentra-se em identificar inovações tecnológicas legalmente protegidas. As patentes concedem direitos exclusivos temporários sobre uma invenção, permitindo que inventores e empresas aproveitem seus esforços criativos. Ao buscar patentes, exploramos os detalhes técnicos das inovações, compreendendo seus métodos de operação e fabricação. Isso é crucial para inventores, empresas e pesquisadores que desejam se manter atualizados com as tecnologias mais recentes antes de desenvolverem suas próprias soluções.

Na busca por patentes relacionadas, destacam-se três delas. A patente de registro BR 10 2020 010686 4 A2 (PANASCO; MIRANDA, 2021) apresenta um *datalogger* com múltiplas opções de comunicação, incluindo Wi-Fi, *Bluetooth* e GPRS, permitindo aos usuários escolher até quatro interfaces diferentes. No entanto, nossa semelhança com essa patente se restringe à capacidade de comunicação GPRS. A patente BR 10 2012 029199 1 A2 (REGIS; DIAS, 2016) se concentra na parametrização de *dataloggers* programáveis para monitoramento automático, não abordando a transmissão de dados. A terceira patente, BR 10 2012 018224 6 A2 (FONSECA FILHO, 2016), trata de um controle remoto de sistemas eletromecânicos com o uso de GPRS, aplicado no

setor químico, mas não compartilha muitas semelhanças com nosso projeto, além da comunicação GPRS.

Sendo assim, após uma análise cuidadosa de cada patente identificada, chegamos à conclusão de que não existem trabalhos semelhantes ao nosso projeto. Cada trabalho parece diferir em termos de finalidade do sistema, equipamentos utilizados e locais de implementação. Em relação à classificação das patentes analisadas, foi observado que a maioria delas se enquadra na classe H04 (Eletricidade, Técnica de Comunicação Elétrica), seguida pela classe G01 (Física, Medição e Teste).

3 METODOLOGIA

Este trabalho de conclusão de curso foi desenvolvido em quatro etapas:

- Escolha do microcontrolador
- Desenvolvimento de algoritmo para receber e segmentar o *status* do módulo SIM900
- Circuito eletrônico para *reset* do módulo SIM900
- Design e confecção da PCI na CNC Router

3.1 Escolha do microcontrolador

O microcontrolador é responsável por formatar, enviar e receber as *strings* para/do *datalogger* e para/do SIM900. Neste trabalho implementamos a solicitação e recebimento de *status* do módulo SIM900 realizando a notificação ao operador via SMS, permitindo que o operador reinicie o sistema via SMS, se necessário.

Para a escolha do microcontrolador deve ser levada em conta a quantidade de dispositivos conectados a ele. Sendo assim, como serão conectados o módulo GPRS e *datalogger*, ele deve ser capaz de se comunicar com dois dispositivos e, portanto, deve dispor de duas portas de comunicação serial. É desejável ainda duas portas adicionais para comunicação com o PC e com um módulo *Bluetooth*. Pela disponibilidade no campus, optamos pelo Arduino Mega.

O Arduino Mega é uma placa de microcontrolador amplamente conhecida e utilizada na comunidade de entusiastas, *makers* e engenheiros. Baseado no microcontrolador ATmega2560, ele é um dos membros populares da família Arduino, conhecido por ter custo acessível para prototipagens, com boa relação entre custo e quantidade de pinos de entrada e saída, bem como entre custo e periféricos.

Devido às suas características, é uma escolha adequada para uma variedade de aplicações, incluindo robótica, automação industrial, controle de motores, impressoras 3D, máquinas CNC e sistemas de monitoramento. E devido à sua popularidade, o Arduino Mega possui uma comunidade ativa de desenvolvedores e uma variedade de recursos online, incluindo tutoriais, bibliotecas e exemplos de código.

Em resumo, o Arduino Mega é uma placa de microcontrolador versátil, com a capacidade computacional necessária e expansível, ideal para uma ampla gama de proje-

tos. Sua riqueza de I/O, capacidade de memória e recursos de comunicação o tornam uma escolha adequada para este projeto.

3.2 Leitura do *status* do SIM900

É importante reconhecer os possíveis problemas e erros que podem surgir ao trabalhar com o SIM900. Alguns dos problemas e erros comuns que podem ocorrer ao usar o SIM900 incluem:

- Problemas de conexão: O SIM900 pode enfrentar dificuldades em estabelecer uma conexão com a rede GSM ou GPRS. Isso pode ocorrer devido a problemas de cobertura de sinal inadequada, problemas com a antena ou configuração inadequada do módulo.
- Erros de comando: Ao enviar comandos AT, que são comandos específicos usados para interagir com o módulo, para o SIM900, podem ocorrer erros de sintaxe, problemas de formatação ou na transmissão do comando via comunicação serial. Esses erros podem resultar em respostas inesperadas ou nenhuma resposta do módulo.
- Falha na leitura do cartão SIM: Se o módulo não conseguir ler o cartão SIM corretamente ou não o reconhecer, ele não poderá se registrar na rede e, portanto, não conseguirá realizar nenhuma operação.
- Falhas de hardware: Como qualquer componente eletrônico, o SIM900 está sujeito a falhas de hardware devido a defeitos em componentes, soldas ou outros problemas relacionados.

Para lidar com esses possíveis erros, é essencial garantir que o módulo esteja corretamente configurado e operando dentro de suas especificações. No algoritmo implementado foram adicionados comandos para enviar o *status* por meio do monitor serial da Arduino IDE e via SMS para o operador do sistema, o qual pode realizar um “reset” remoto, também através de SMS.

3.2.1 Desenvolvimento de algoritmo para receber e segmentar o *status* do módulo SIM900

Ao ser energizado, o SIM900 envia uma *string* de dados que é recebida pelo Arduino. *Strings* de *status* também são retornadas sempre que há alguma configuração no módulo. Por fim, *Strings* de SMS também são retornadas do SIM900 sempre que uma nova mensagem chega ao *SIM Card*. No microcontrolador, essas informações são segmentadas e apresentadas ao usuário.

Por exemplo, de acordo com a Figura 3, a *string* recebida contém informações como a confirmação de que as funções do módulo foram liberadas (*status* CFUN), *status* do *SIM Card* (*status* CPIN), registro do dispositivo na rede (*status* CREG), indicações sobre a operadora (*status* PSNWID) e data e hora da conexão (*status* PSUTTZ). Com essas informações, são feitas as primeiras segmentações de *status* do módulo

de comunicação. O algoritmo divide a informação e trata separadamente cada um dos *status*. Para dividir essa *string* de *status*, foram utilizadas as funções *readStringUntil*, *startsWith* e *substring*.

Figura 3 – String de status do SIM900

```
RDY
+CFUN: 1
+CPIN: READY
+CREG: 0
+CREG: 2
Call Ready
+CREG: 1
*PSNWID: "724", "05", "CLARO BR", 0, "", 0
*PSUTTZ: 2023, 8, 8, 10, 37, 31, "-12", 1
DST: 1
```

Fonte: Autoria Própria (2023)

Exemplificando o código, considerando a *string* apresentada na Figura 3, e que o caractere “\n” (nova linha) existe ao final de cada linha da *string*, este caractere foi utilizado como delimitador na função *readStringUntil*. Em seguida, por meio da função *startsWith*, é verificado se a linha começa com “+CFUN:”, “+CPIN:” ou “+CREG:”, que são as linhas que consideramos relevantes para identificação se houve falhas. Por fim, com a função *substring*, apresentando o número 7 como argumento, extraímos os sete primeiros caracteres da linha (a partir do caractere “+”). Com isso é possível identificar o tipo do retorno do SIM900 (CFUN, CPIN, CREG) e armazenar no Arduino, em variáveis distintas, os valores retornados pelo SIM900. Procedimento semelhante é realizado para ler o PSNWID e PSUTTZ, apenas variando o início da linha, para “*PSNWID:” ou “*PSUTTZ:”, e a quantidades de caracteres a serem extraídos, para 8.

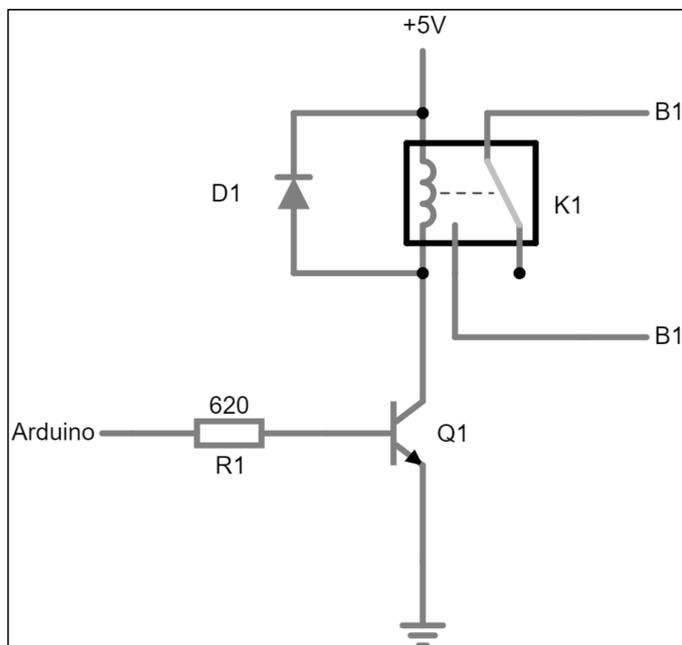
3.3 Circuito eletrônico para reset do módulo SIM900

Como qualquer dispositivo eletrônico, o SIM900 pode ser suscetível a falhas. Por exemplo, foi verificado que flutuações na tensão de alimentação causa travamento do módulo, assim como limitação na corrente de pico fornecida para o módulo e ruídos no canal de comunicação serial entre o SIM900 e o Arduino. Considerando ainda os ambientes suscetíveis a intempéries previstos para a instalação deste projeto, é necessário um sistema capaz de “destravar” o módulo SIM900.

Com base em observações experimentais, foi constatado que uma simples reinicialização do módulo SIM900 é suficiente para resolver os problemas de travamento do módulo. Considerando essa necessidade, foi desenvolvido um circuito eletrônico para

realizar essa reinicialização através do circuito do botão “power” do módulo. O circuito desenvolvido neste TCC pode ser visualizado na Figura 4.

Figura 4 – Circuito para reset do módulo SIM900



Fonte: Autoria Própria (2023)

Quando o Arduino envia um pulso de 5 V, o transistor Q1 entra na região de saturação e aciona a bobina do relé K1. Quando o relé é acionado, seus contatos são comutados e conectam os terminais do botão de comunicação (B1) do SIM900. É importante observar que, para que essa conexão seja realizada, os contatos de K1 estão em paralelo com o botão B1. Observar o diodo de proteção D1 que permite um caminho para a corrente durante a desenergização da bobina, evitando picos de tensão no coletor do transistor.

Foi utilizado o Relé SRD-5VDC-SL-C, que era o disponível no LabMaker Pesqueira. Este relé suporta até 10 A no lado do contator, que é suficiente para acionar o botão “power” do SIM900 visto que este botão aciona um circuito de alta impedância, e é necessário 71,4 mA no lado da bobina para seu acionamento. Considerando o transistor BC548, disponível no LabMaker, que suporta corrente de coletor até 100 mA, e que a bobina será acionada por 5 s de forma não periódica, calculamos a corrente de base em 7,1 mA ao considerar, de forma conservadora, o beta de saturação 10. Considerando a saída do Arduino 5 V, R1 foi calculado em 602 ohms. Pela proximidade, foi utilizado um resistor comercial de 620 ohms. Com este valor, calculamos a corrente fornecida pelo Arduino é de 7,68 mA. Segundo os dados do fabricante, o microcontrolador é capaz de suprir até 40 mA. Portanto, o circuito dissipa uma potência de aproximadamente 0,4 W durante seu funcionamento.

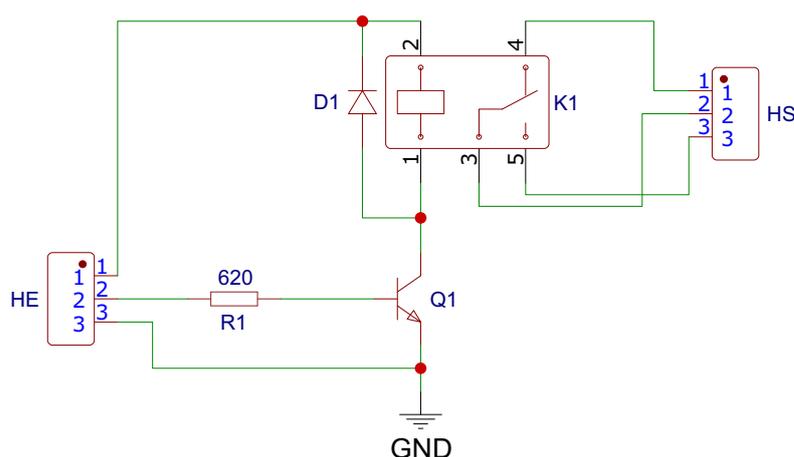
O SIM900 consome uma corrente significativa quando está conectando e transmitindo dados. No entanto, como só é necessário, neste projeto, o envio de dados por

poucos segundos a cada hora, não há necessidade de manter o módulo GPRS ligado o tempo todo, mesmo considerando o modo “*sleep*” do módulo. Portanto, o circuito também pode ser utilizado para ligar e desligar o módulo, ativando a comunicação apenas alguns segundos antes e depois do envio de dados a cada hora, economizando energia quando a comunicação não é necessária. Contudo, vale lembrar que enquanto o módulo estiver desligado, de fato não será possível o recebimento de SMS do operador neste período. A decisão de manter o módulo ligado ou desligado caberá ao programador em etapas futuras deste projeto.

3.4 Design e confecção da PCI na CNC Router

Terminada a prototipagem e testes do circuito, foi inicialmente testado desenvolver o projeto da PCI no software EasyEDA (ver Figura 5). O EasyEDA é uma plataforma web onde o usuário pode projetar, simular e encomendar a manufatura de circuitos eletrônicos. (AMARAL, 2015).

Figura 5 – Esquema elétrico do circuito de reset do SIM900



Fonte: Autoria Própria (2023)

Finalizado o esquema elétrico, a próxima etapa é o *design* da PCI e, em seguida, gerar os arquivos G-Code com os percursos para manufatura da PCI. O G-Code é uma linguagem de programação que controla máquinas CNC, sendo responsável por informar às máquinas CNC como realizar operações de usinagem, como corte, perfuração e fresagem, de forma automatizada e precisa (KALATEC AUTOMAÇÃO, s.d.). Entretanto, foi observado que no EasyEDA não é possível gerar os arquivos G-Code, sendo necessária a utilização de outro software.

A segunda alternativa analisada foi o Eagle, da Autodesk. O Eagle é um aplicativo que permite ao usuário criar diagramas esquemáticos e layouts de projetos eletrônicos. Para isso, dispõe de uma ampla biblioteca de componentes eletrônicos (BEZERRA, 2018).

Apesar de sua vasta biblioteca, o Eagle carece de alguns componentes eletrônicos, como foi o caso do relé K1, modelo SRD-05VDC-SL-C, do circuito de *reset*. Para solu-

cionar esse problema, foi necessário recorrer ao *download* do pacote do componente no GitHub (DUMAN, 2014).

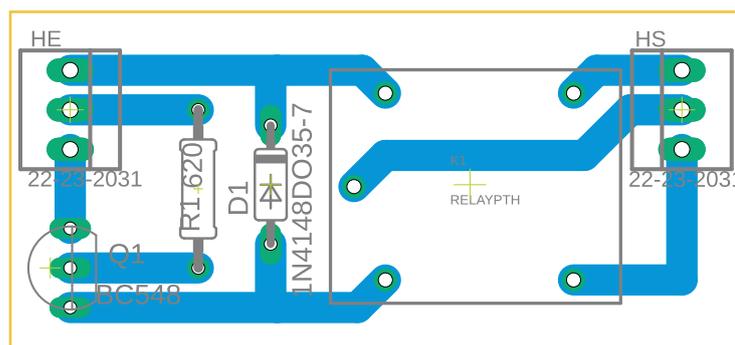
Após o *download* da biblioteca do componente, o circuito da Figura 4 pôde ser redesenhado no Eagle. E para gerar os G-Code no Eagle, foi necessário utilizar a extensão PCB-GCODE, que se encontra disponível no GitHub (JOHNSON, 2022). Os parâmetros utilizados na CNC Router podem ser vistos no Apêndice A.

4 RESULTADOS E DISCUSSÕES

4.1 Manufatura da PCI

Como parâmetros principais, foram adotados furos com 1 mm de diâmetro e trilhas de 2 mm de espessura. Como o circuito é pequeno, não foi utilizado o recurso de *autorouter* do *software*. Após gerar os G-Codes, os arquivos com os percursos seguiram para o *software* da CNC Router para manufatura da PCI. O resultado desse procedimento pode ser visualizado na Figura 6.

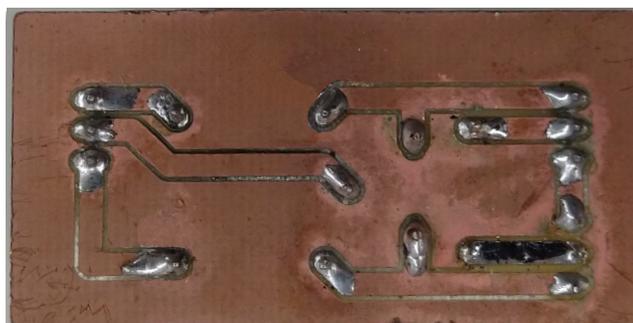
Figura 6 – PCI do circuito de *reset* no Eagle



Fonte: Autoria Própria (2023)

A Figura 7 mostra resultado da PCI.

Figura 7 – Vista inferior da PCI



Fonte: Autoria Própria (2023)

4.2 Testes do equipamento completo

Após conectar a PCI ao equipamento, iniciamos os testes com o equipamento para verificar se o seu funcionamento estava de acordo com o previsto. Os resultados desses testes serão descritos adiante, nas próximas subseções.

4.2.1 Status do módulo SIM900

A *string* original que o SIM900 envia imediatamente ao iniciar e as informações segmentadas e armazenadas em variáveis independentes, podem ser observadas na Figura 8.

Figura 8 – String dos status de operação do SIM900 após tratamento de informações. Esquerda: String original. Direita: Informações obtidas, segmentadas.

RDY	----
+CFUN: 1	Iniciando o módulo SIM900
+CPIN: READY	----
+CREG: 0	CFUN: 1
+CREG: 2	----
Call Ready	CPIN: READY
+CREG: 1	----
*PSNWID: "724", "05", "CLARO BR", 0, "", 0	CREG: 0
*PSUTTZ: 2023, 8, 8, 10, 37, 31, "-12", 1	----
DST: 1	CREG: 2

	Dispositivo pronto para realizar chamadas

	CREG: 1

	PSNWID: "724", "05", "CLARO BR", 0, "", 0

	PSUTTZ: 2023, 11, 24, 23, 37, 10, "-12", 1

	DST: 1

Fonte: Autoria Própria (2023)

Para visualização posterior dos dados de status, basta enviar o caractere 's' pelo monitor serial ou módulo Bluetooth que as informações são mostradas na tela do computador ou do dispositivo móvel. Ao enviar 's', o retorno do microcontrolador pode ser visto em Figura 9.

Caso o usuário envie por SMS o comando "status", irá receber também por SMS os *status* do módulo.

Figura 9 – Retorno do *status* do SIM900 após comando 's' via serial ou "status" via SMS

```

-----
STATUS DAS VARIÁVEIS
-----
Status CFUN: 1
-----
Status CPIN: READY
-----
Status CREG: 3
-----
Status Call Status:
-----
Status PSNWID: "724","05","CLARO BR",0,"",0
-----
PSUTTZ: 2023, 11, 24, 23, 37, 10, "-12",1
-----
Status DST: 1
-----
Última mensagem recebida:
-----
Status Configuração GPRS: Não Configurado
-----

```

Fonte: Aatoria Própria (2023)

4.2.2 Configuração e conexão da rede GPRS

O primeiro passo para o envio de dados é a configuração da rede GPRS. Pela serial, ao enviar o caractere 'c', o microcontrolador executa uma rotina de comandos AT para configurar a rede GPRS no módulo e, ao terminar, envia um SMS para o número previamente cadastrado. O andamento dessa configuração pode ser acompanhado através da mesma tela do monitor serial.

Adicionalmente, também foi implementado um comando via SMS para executar esse procedimento. Ao enviar a palavra "config", a mesma rotina de configuração é executada.

4.2.3 Desativando a conexão GPRS

Nos testes realizados no *LabMaker* foi visto a possibilidade de ocorrência de erro quando da conexão GPRS. E para esse caso, é possível apenas desativar a conexão para, enfim, ligá-la novamente. Sendo assim, também foi implementado no algoritmo um comando para desativar a rede GPRS. Quando o usuário envia o caractere 'd', o usuário pode acompanhar a rotina pelo monitor serial e ao seu término recebe um SMS informando que ela foi executada. Caso o usuário envie por SMS a palavra "desconfig", irá receber também por SMS o resultado do processo.

4.2.4 Reset do SIM900

Para realizar o *reset* do equipamento, o profissional pode enviar o caractere 'r' pelo monitor serial que a rotina de reinicialização será executada. Para executar de forma remota, via SMS, o comando é "reset". Observar que, no caso de travamento do módulo SIM900, não será possível a comunicação com o Arduino via SMS. A automatização do *reset* faz parte dos trabalhos futuros.

5 CONSIDERAÇÕES FINAIS / CONCLUSÕES

Através do desenvolvimento da interface de comunicação GPRS, o IFPE *Campus* Pesqueira poderá aumentar a eficiência e a confiabilidade dos sistemas de aquisição de dados.

Porém, são diversos os problemas de conectividade. Esse trabalho se empenhou em encontrar soluções para minimizar tais desafios. Um sistema de monitoramento do módulo de comunicação e um sistema de *reset* foram desenvolvidos, testados e prototipados com êxito. Além disso, é importante destacar que o dispositivo se mostrou eficaz, cumprindo com o objetivo de monitoramento do módulo GPRS, assim como de informar o usuário acerca de seus status de funcionamento. A partir deste trabalho, é possível realizar o monitoramento e intervenções no equipamento, sem depender da presença física constante de um profissional.

Quando todos os requisitos de desempenho e funcionalidade forem atendidos, é esperado que a Interface GPRS em desenvolvimento seja uma opção economicamente viável para ser implantada e utilizada em estações solarimétricas da REDESOLPE, assim como em outras aplicações, ao mesmo tempo em que colaborará para o avanço de novas pesquisas acadêmicas.

Reconhecemos a importância de considerar o pedido de registro de propriedade intelectual para proteger a propriedade de alguns itens desenvolvidos no projeto, com o objetivo de garantir sua exclusividade e incentivar inovações futuras.

Como sugestão, incluir em trabalhos futuros elementos como o desenvolvimento de um aplicativo para Android e iOS, o aprimoramento e automatização do sistema de detecção de erros, bem como a expansão desse sistema para abranger eventuais erros que o *datalogger* possa apresentar.

REFERÊNCIAS

AMARAL, Garcia. Embarcados. **EasyEDA – Simulador de circuitos on-line**. [S.l.: s.n.], 16 dez. 2015. Disponível em: <https://embarcados.com.br/easyeda/>. Acesso em 20 out. 2023.

BEZERRA, B. G. S. **IoT baseada em PHM aplicada a centrais condicionadoras de ar**. 2018. F. 90. Dissertação (Mestrado em Engenharia Elétrica) – Programa de Pós-Graduação em Sistemas de Comunicação e Automação, Universidade Federal Rural do Semi-Árido, Mossoró, RN.

DUMAN, David. **SRD-05VDC-SL-C.lbr**. [S.l.: s.n.], 14 set. 2014. Disponível em: <https://github.com/dvdnhm/Maker-Tools-Eagle-Libraries/blob/master/SRD-05VDC-SL-C.lbr>. Acesso em 16 out. 2023.

FONSECA FILHO, Sérgio de Aquino. **Processo de controle de sistemas eletromecânicos à distância**. Depositante: Empresa Brasileira de Segurança LTDA. Procurador: Escobar Advocacia - Propriedade e Intelectual e Direito Empresarial. BR 10 2012 018224 6 A2. Depósito: 23 jul. 2012. Concessão: 07 jun. 2016.

GARCIA, Fernando Deluno. Embarcados. **Introdução aos sistemas embarcados e microcontroladores**. [S.l.: s.n.], 19 jul. 2018. Disponível em:

<https://embarcados.com.br/sistemas-embarcados-e-microcontroladores/>. Acesso em 20 out. 2023.

IFPE. **IFPE-Pesqueira se torna Centro de Referência em Energia Solar**. [S.l.: s.n.], 7 jul. 2020. Disponível em:

<https://portal.ifpe.edu.br/pesqueira/noticias/ifpe-pesqueira-se-torna-centro-de-referencia-em-energia-solar/>. Acesso em 02 out. 2023.

JOHNSON, John T. **PCB-GCODE**. [S.l.: s.n.], 2 set. 2022. Disponível em:

<https://github.com/JohnAtl/pcbrcode>. Acesso em 16 out. 2023.

KALATEC AUTOMAÇÃO. **Código G para Usinagem CNC: o que é, funções e modais**. [S.l.: s.n.]. Disponível em: <https://blog.kalatec.com.br/codigo-g/>.

Acesso em 16 out. 2023.

OLIVEIRA, Jefferson Rodrigues de. **Acionamento remoto de equipamentos domésticos**. 2016. F. 74. Monografia (Curso de Graduação em Engenharia de Controle e Automação) – Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF.

OLIVEIRA, José Raimundo de; SHIN-TING, Wu. **Metodologia de Projeto de Sistema Embarcado**. [S.l.: s.n.], ago. 2019. Disponível em:

<https://www.dca.fee.unicamp.br/courses/EA075/2s2019/notas/topico3.pdf>. Acesso em 16 out. 2023.

PANASCO, Gabriel Fabiano; MIRANDA, Oliver Augusto. **Coletor e registrador de dados com sensores variados**. Depositante: Auro Solucoes Tecnologicas Industria e Comercio de Equipamentos Ltda. Procurador: Cristiane Rumika Minowa Ozawa. BR 10 2020 010686 4 A2. Depósito: 27 maio 2020. Concessão: 07 dez. 2021.

PATIL, Mr Vijay R. A Low Cost Automated Irrigation System Using WSN and GPRS Module. **International Journal of Advance Research and Innovative Ideas in Education**, v. 8, 2022.

PEDROSA FILHO, Manoel H. de O. REDE SOLARIMÉTRICA DE PERNAMBUCO. **IX Congresso Brasileiro de Energia Solar**, 2022.

PRIAMBODO, A. S.; NUGROHO, A. P. Design & Implementation of Solar Powered Automatic Weather Station based on ESP32 and GPRS Module. **Journal of physics: Conference series**, v. 1737, p. 012009, 2021.

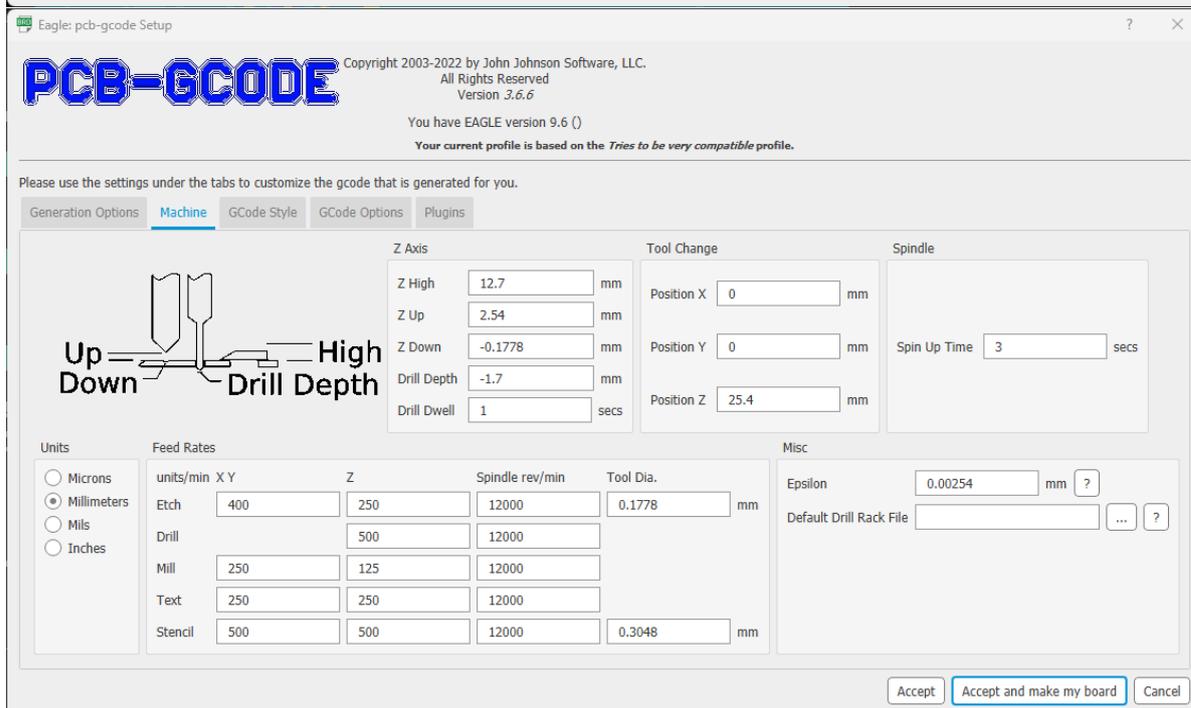
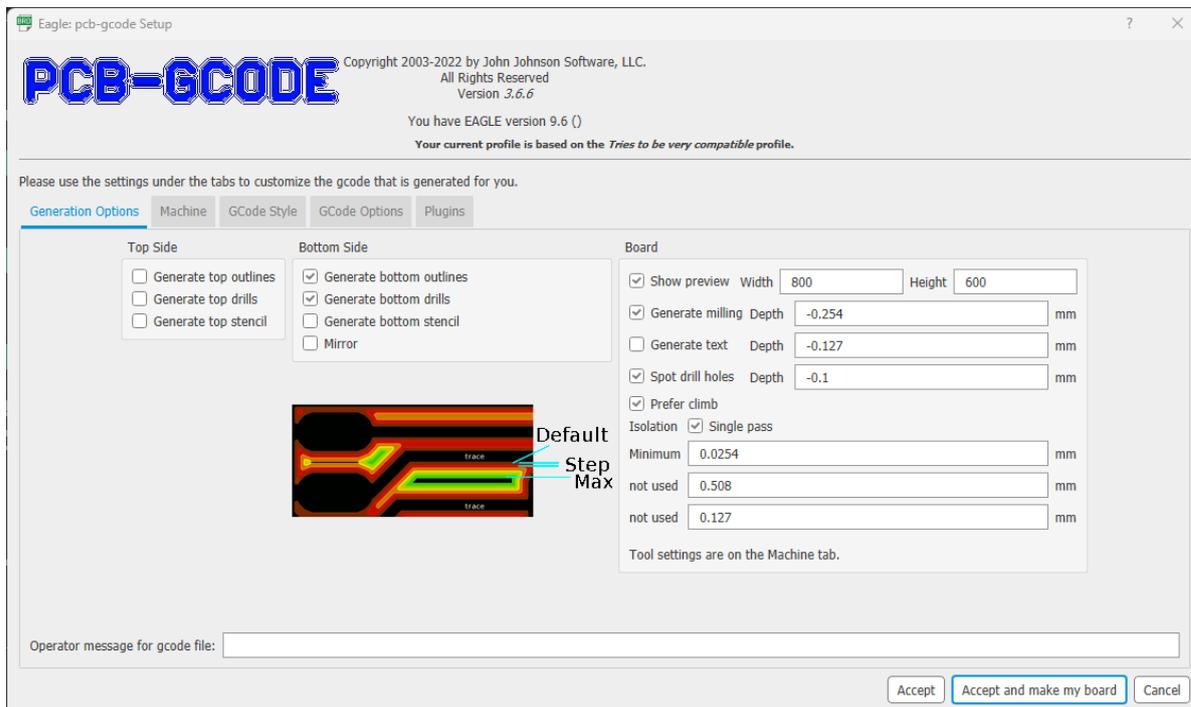
REGIS, Adriano; DIAS, Roberto Alexandre. **Sistema e método dinâmico de parametrização de registradores de dados programáveis**. Depositante: Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina. BR 10 2012 029199 1 A2. Depósito: 14 nov. 2012. Concessão: 29 mar. 2016.

SILVA, Armando Vitor de Oliveira. **Alarme com ativação por sensor presencial e alerta via SMS**. 2016. F. 67. Monografia (Curso de Graduação em Engenharia da Computação) – Centro Universitário de Brasília, Faculdade de Tecnologia e Ciências Sociais Aplicadas, Brasília, DF.

SIMCOM. **SIM900 AT Command Manual**. [S.l.: s.n.], 2010. Disponível em: https://www.espruino.com/datasheets/SIM900_AT.pdf. Acesso em 30 ago. 2023.

XIONG, Yicheng et al. Remote monitoring system design of power substation based on GPRS. **Journal of Physics: Conference Series**, v. 2108, p. 012031, 2021.

APÊNDICE A - PARÂMETROS DA EXTENSÃO PCB-GODE PARA A CNC ROUTER DO LABMAKER



Eagle: pcb-gcode Setup

PCB-GCODE Copyright 2003-2022 by John Johnson Software, LLC.
All Rights Reserved
Version 3.6.6

You have EAGLE version 9.6 ()
Your current profile is based on the *Tries to be very compatible profile.*

Please use the settings under the tabs to customize the gcode that is generated for you.

Generation Options Machine GCode Style **GCode Options** Plugins

NC File Comments

- NC File Comment from Board
- Nc File Comment Date
- Nc File Comment Machine Settings
- Nc File Comment Pcb Defaults Settings

Other Options

- Use user gcode (from user-gcode.h) Compact gcode
- Debug Flag
- Do tool change with zero step Use line numbers? Format: N%05d
- Flip board in Y instead of X Use simple drill code

File Naming

Filename Base: \$BOARD_PATH/\$BOARD_NAME Extension: tap Word for 'top': top Word for 'bottom': bot

Word for 'etch': trilhas Word for 'drill': furos Word for 'mill': mill Word for 'text': text Word for 'stencil': stencil

Top (Component) Side Files

Etching	.\$SIDE.\$FILE.\$EXT	Drill	.\$SIDE.\$FILE.\$EXT
Mill	.\$SIDE.\$FILE.\$EXT	Text	.\$SIDE.\$FILE.\$EXT
Stencil	.\$SIDE.\$FILE.\$EXT		

Bottom (Solder) Side Files

Etching	.\$SIDE.\$FILE.\$EXT	Drill	.\$SIDE.\$FILE.\$EXT
Mill	.\$SIDE.\$FILE.\$EXT	Text	.\$SIDE.\$FILE.\$EXT
Stencil	.\$SIDE.\$FILE.\$EXT		

Test Filenames Help

Accept Accept and make my board Cancel

APÊNDICE B - FIRMWARE PARA DESENVOLVIMENTO DO SISTEMA DE MONITORAMENTO

```

/*
    Instituto Federal de Educação, Ciência e Tecnologia de
    Pernambuco - Campus Pesqueira
    Bacharelado em Engenharia Elétrica
    Trabalho de Conclusão de Curso - Sistema de identificação
    de erros em módulo GPRS com confecção de PCI em CNC Router
    Discente: Christian Araujo de Souza | Email:
    christian.araujo96@outlook.com
    Orientador: Prof. Dr. Ygo Neto Batista

    Observações:
    Caso precisem utilizar este código, mesmo que apenas em
    partes, por gentileza citar a fonte.
*/

// PINAGEM
#define rele 7 // Relé de acionamento conectado no pino 7

// VARIÁVEIS
// Variáveis para armazenar status de retorno e serial
char serial;
char x1 = ' '; char x2 = ' '; char x3 = ' '; char x4 = ' ';
char x5 = ' '; char x6 = ' '; char x7 = ' '; char x8 = ' ';
char x9 = ' ';
String cmdReturn;
String dadosRecebidos;

// Variáveis que armazenam os status de comunicação do SIM900
String cfuns; //Armazena a indicação de funcionamento do
módulo SIM900
String cpin; //Armazena a indicação de funcionamento do
cartão SIM
String creg; //Armazena a indicação de registro do SIM900 na
rede
String callStatus; //Armazena a indicação caso o módulo
esteja pronto para chamadas
String psnwid; //Armazena informações sobre a operadora
String psuttz; //Armazena informações sobre a data e hora
atual
String dstStatus; //Armazena a indicação sobre horário de
verão
String sms; // Armazena o SMS recebido

// Variáveis que armazenam os status de configuração do SIM900
int config_gprs = 0; // Variável que indica se foi configurada
a conexão GPRS do SIM900
int cont_ok = 0; // Variável para contar quantas vezes o
módulo retornou com "OK"

// CONFIGURAÇÕES INICIAIS

```

```

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, LOW);
  pinMode(rele, OUTPUT);
  digitalWrite(rele, LOW);
  Serial.begin(9600); //Inicia serial para comunicação com
  computador
  Serial1.begin(9600); //Inicia serial para comunicação com
  módulo SIM900
  resetar_variaveis_status();
  acionar_rele();
}

// PROGRAMAÇÃO PRINCIPAL
void loop() {
  if (Serial1.available()) {
    dadosRecebidos = Serial1.readStringUntil('\n'); // Lê a
    linha completa

    // Verifica se a linha contém um dos prefixos dos status
    if (dadosRecebidos.startsWith("+CFUN:")) { // Remove o
    prefixo e armazena o valor na variável cfuns
      cfuns = dadosRecebidos.substring(7);
      Serial.println("CFUN: " + cfuns);
    } else if (dadosRecebidos.startsWith("+CPIN:")) { //
    Remove o prefixo e armazena o valor na variável cpin
      cpin = dadosRecebidos.substring(7);
      Serial.println("CPIN: " + cpin);
    } else if (dadosRecebidos.startsWith("+CREG:")) { //
    Remove o prefixo e armazena o valor na variável creg
      creg = dadosRecebidos.substring(7);
      Serial.println("CREG: " + creg);
    } else if (dadosRecebidos.equals("Call Ready")) { //
    Identifica o status acerca da possibilidade de realizar
    chamadas
      callStatus = "Dispositivo pronto para realizar
      chamadas";
      Serial.println(callStatus);
    } else if (dadosRecebidos.startsWith("*PSNWID:")) { //
    Remove o prefixo e armazena o valor na variável psnwid
      psnwid = dadosRecebidos.substring(8);
      Serial.println("PSNWID: " + psnwid);
    } else if (dadosRecebidos.startsWith("*PSUTTZ:")) { //
    Remove o prefixo e armazena o valor na variável psuttz
      psuttz = dadosRecebidos.substring(8);
      Serial.println("PSUTTZ: " + psuttz);
    } else if (dadosRecebidos.startsWith("DST:")) {
      dstStatus = dadosRecebidos.substring(4);
      Serial.println("DST: " + dstStatus);
    } else if (dadosRecebidos.startsWith("+CMT:")) { //

```

```

Remove o prefixo e armazena o valor na variável sms
    sms = dadosRecebidos.substring(7);
    waitSMS();
} else if (dadosRecebidos.startsWith("NORMAL POWER
DOWN")) { // Identifica se o módulo está sendo desligado
    Serial.println("Desligando o módulo SIM900");
} else if (dadosRecebidos.startsWith("RDY")) { //
Identifica se o módulo está sendo ligado
    Serial.println("Iniciando o módulo SIM900");
} else if (dadosRecebidos.startsWith("")) { //
Identifica espaços vazios após as mensagens de retorno do
SIM900
    Serial.println("----");
} else {
    Serial.print("Recebido: ");
Serial.println(dadosRecebidos); // Caso identifique qualquer
mensagem não prevista acima, ela é mostrada por inteiro
}
}
if (Serial.available()) {
    serial = Serial.read();

    if (serial == 'c') { // Caso seja digitado o caractere
'c' no monitor serial...
        configGPRS();
    }

    if (serial == 'd') { // Caso seja digitado o caractere
'd' no monitor serial...
        desativar_GPRS();
    }

    if (serial == 'l') { // Caso seja digitado o caractere
'l' no monitor serial...
        acionar_rele();
    }

    if (serial == 'o') { // Caso seja digitado o caractere
'o' no monitor serial...
        enviar_SMS("OK");
    }

    if (serial == 'r') { // Caso seja digitado o caractere
'r' no monitor serial...
        reset_comunicacao();
    }

    if (serial == 's') { // Caso seja digitado o caractere
'S' no monitor serial...
        ver_status();
}

```



```

Serial.println("Configurando conexao para GPRS...");
Serial1.println("AT+SAPBR=3,1,\"Contype\",\"GPRS\");
waitOK(); Serial.println("----"); cont_ok = cont_ok + 1;

Serial.println("Configurando conexao com a rede
celular...");
Serial1.println("AT+SAPBR=3,1,\"APN\",\"claro.com.br\");
// Dados da operadora do SIM Card
waitOK(); Serial.println("----"); cont_ok = cont_ok + 1;

Serial.println("Ativando o GPRS...");
Serial1.println("AT+SAPBR=1,1"); // Comando AT para
ativar o GPRS
waitOK(); Serial.println("----"); cont_ok = cont_ok + 1;

Serial.println("Verifica se recebeu IP");
Serial1.println("AT+SAPBR=2,1");
waitOK(); Serial.println("----");
cont_ok = cont_ok + 1;
}
// Manda SMS para indicar que o GPRS foi configurado
if (cont_ok == 6) {
    config_gprs = 1;
    enviar_SMS("GPRS Configurado");
}
}

// Método para desativar o GPRS
void desativar_GPRS() {
    if (config_gprs == 1) {
        Serial.println("Desativando o GPRS...");
        cont_ok = 0;
        config_gprs = 0;
        Serial1.println("AT+SAPBR=0,1"); // Comando AT para
desativar o GPRS
        waitOK(); Serial.println("----");
        enviar_SMS("GPRS Desativado");
    }
}

// Método para retornar string do SIM900
void retorno() {
    cmdReturn = Serial1.readString();
    Serial.println("----");
    Serial.println(cmdReturn);
}

// Método para esperar uma resposta OK do SIM900
void waitOK() {

```

```

unsigned long tIni = millis();
while (millis() - tIni < 5000) {
  if (Serial1.available()) {
    x1 = Serial1.read();
    Serial.write(x1);
    if ((x1 == 'K') && (x2 == 'O')) {
      x1 = ' '; x2 = ' ';
      break;
    }
    x2 = x1;
  }
}
delay(10);
while (Serial1.available()) {
  Serial.write(Serial1.read());
};
}

// Método para espera de comandos SMS do usuário
void waitSMS() {
  unsigned long tIni = millis();
  while (millis() - tIni < 5000) {
    if (Serial1.available()) {
      x1 = Serial1.read();
      Serial.write(x1);
      if ((x1 == 't') && (x2 == 'e') && (x3 == 's') && (x4 ==
'e') && (x5 == 'r')) {
        // Se receber um sms com a palavra "reset"...
        reset_comunicacao();
        x1 = ' '; x2 = ' '; x3 = ' '; x4 = ' '; x5 = ' '; x6
= ' '; x7 = ' '; x8 = ' '; x9 = ' ';
        break;
      } else if ((x1 == 's') && (x2 == 'u') && (x3 == 't') &&
(x4 == 'a') && (x5 == 't') && (x6 == 's')) {
        // Se receber um sms com a palavra "status"...
        ver_status_SMS();
        x1 = ' '; x2 = ' '; x3 = ' '; x4 = ' '; x5 = ' '; x6
= ' '; x7 = ' '; x8 = ' '; x9 = ' ';
        break;
      } else if((x1 == 'g') && (x2 == 'i') && (x3 == 'f') &&
(x4 == 'n') && (x5 == 'o') && (x6 == 'c')){
        // Se receber um sms com a palavra "config"...
        configGPRS();
        x1 = ' '; x2 = ' '; x3 = ' '; x4 = ' '; x5 = ' '; x6
= ' '; x7 = ' '; x8 = ' '; x9 = ' ';
        break;
      } else if((x1 == 'g') && (x2 == 'i') && (x3 == 'f') &&
(x4 == 'n') && (x5 == 'o') && (x6 == 'c') && (x4 == 's') &&
(x5 == 'e') && (x6 == 'd')){
        // Se receber um sms com a palavra "desconfig"...

```

```

        desativar_GPRS();
        x1 = ' '; x2 = ' '; x3 = ' '; x4 = ' '; x5 = ' '; x6
= ' '; x7 = ' '; x8 = ' '; x9 = ' ';
        break;
    }
    x9 = x7; x8 = x7; x7 = x6; x6 = x5; x5 = x4; x4 = x3;
x3 = x2; x2 = x1;
    }
}
delay(10);
while (Serial1.available()) {
    Serial.write(Serial1.read());
}
}

// Método para resetar as variáveis que armazenam os status
de comunicação do SIM900
void resetar_variaveis_status() {
    cfuns = " ";
    cpin = " ";
    creg = " ";
    callStatus = " ";
    psnwid = " ";
    psuttz = " ";
    dstStatus = " ";
}

// Método para realizar o acionamento do relé
void acionar_rele() {
    digitalWrite(rele, HIGH); //Para a PCB do relé
    digitalWrite(LED_BUILTIN, HIGH);
    delay(5000);
    digitalWrite(rele, LOW); //Para a PCB do relé
    digitalWrite(LED_BUILTIN, LOW);
}

// Método para resetar a comunicação do SIM900
void reset_comunicacao() {
    config_gprs = 0;
    Serial.println("----");
    Serial.println("INICIANDO RESET DA COMUNICAÇÃO DO SIM900");
    Serial.println("----");
    resetar_variaveis_status();
    Serial.println("Limpeza das variáveis: OK!");
    acionar_rele();
    delay(3000);
    acionar_rele();
    Serial.println("----");
}

```

```

// Método para mostrar o status das variáveis
void ver_status() {
  Serial.println("----");
  Serial.println("STATUS DAS VARIÁVEIS");
  Serial.println("----");
  Serial.println("Status CFUN: " + cfuns);
  Serial.println("----");
  Serial.println("Status CPIN: " + cpin);
  Serial.println("----");
  Serial.println("Status CREG: " + creg);
  Serial.println("----");
  Serial.println("Status Call Status: " + callStatus);
  Serial.println("----");
  Serial.println("Status PSNWID: " + psnwid);
  Serial.println("----");
  Serial.println("Status PSUTTZ: " + psuttz);
  Serial.println("----");
  Serial.println("Status DST: " + dstStatus);
  Serial.println("----");
  Serial.println("Última mensagem recebida: " + sms);
  Serial.println("----");
  if (config_gprs == 1) {
    Serial.println("Status Configuração GPRS: Configurado");
    Serial.println("----");
  } else if (config_gprs == 0) {
    Serial.println("Status Configuração GPRS: Não
Configurado");
    Serial.println("----");
  }
}

// Método para mostrar o status das variáveis e enviar via
SMS
void ver_status_SMS() {
  String mensagem;
  Serial.println("----");
  Serial.println("STATUS DAS VARIÁVEIS");
  Serial.println("----");
  Serial.println("Status CFUN: " + cfuns);
  Serial.println("----");
  Serial.println("Status CPIN: " + cpin);
  Serial.println("----");
  Serial.println("Status CREG: " + creg);
  Serial.println("----");
  Serial.println("Status Call Status: " + callStatus);
  Serial.println("----");
  Serial.println("Status PSNWID: " + psnwid);
  Serial.println("----");
  Serial.println("Status PSUTTZ: " + psuttz);
  Serial.println("----");
}

```

```
Serial.println("Status DST: " + dstStatus);
Serial.println("----");
Serial.println("Última mensagem recebida: " + sms);
Serial.println("----");
if (config_gprs == 1) {
    Serial.println("Status Configuração GPRS: Configurado");
    Serial.println("----");
    mensagem = "Status CFUN, CPIN, CREG e GPRS: " + cfun + ",
" + cpin + ", " + creg + ", configurado."; enviar_SMS(mensagem);
} else if (config_gprs == 0) {
    Serial.println("Status Configuração GPRS: Não
Configurado");
    Serial.println("----");
    mensagem = "Status CFUN, CPIN, CREG e GPRS: " + cfun + ",
" + cpin + ", " + creg + ", não configurado.";
enviar_SMS(mensagem);
}
}
```

ANEXO 1 - LISTA DE COMANDOS AT UTILIZADOS

- AT: Comando de verificação básica de comunicação, onde o módulo geralmente responde com "OK";
- AT+CCID: Verifica o número do cartão SIM inserido no módulo. O módulo responderá com o número ICCID (*Integrated Circuit Card Identifier*) do cartão SIM, que é um identificador único do cartão;
- AT+CPIN: Verifica o status do PIN (*Personal Identification Number*) do cartão SIM;
- AT+CREG: Consulta o status de registro na rede;
- AT+CSQ: Consulta a qualidade do sinal da rede;
- AT+CMGF: Define o modo de mensagem (texto ou PDU) para envio e recebimento de SMS;
- AT+CMGS: Envia uma mensagem de texto;
- AT+CMGL: Lista mensagens SMS armazenadas na memória do módulo;
- AT+CMGR: Lê uma mensagem SMS específica;
- AT+CMGD: Exclui uma mensagem SMS;
- ATD: Inicia uma chamada telefônica;
- ATH: Encerra uma chamada telefônica;
- AT+CGATT: Ativa ou desativa o serviço GPRS;
- AT+CGDCONT: Define os parâmetros da conexão GPRS;
- AT+CIICR: Inicia a conexão GPRS;
- AT+CIPSTART: Inicia uma conexão TCP ou UDP;
- AT+CIPSEND: Envia dados por meio da conexão TCP ou UDP;
- AT+CIPCLOSE: Fecha uma conexão TCP ou UDP;
- AT+SAPBR: Configura e gerencia a conexão GPRS.