**MÓDULO 2:**

**IDENTIFICADOR DE CURTO-CIRCUITO IEC E FUNÇÕES DE PROTEÇÃO EM TEMPO REAL**

**Documento TR-01  
Versão 1.0 de março de 2024  
Projeto da disciplina STR  
www.feelt.ufu.br**

**Alan Petrônio Pinheiro**

Coordenador do projeto – UFU/LRI

**Execução e pesquisa:**

Thiago Henrique Arbuini Rodrigues - 11911ECP002

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1. DATA VERSÃO ORIGINAL**  19-03-2024 | | **2. DATA ÚLTIMA ATUALIZAÇÃO**  19-03-2024 | | **3. DATA COBERTA**  MAR/24 ATÉ MAI/24 |
| **4. TÍTULO DESTE DOCUMENTO**  REPORTE TÉCNICO DO IDENTIFICADOR DE CURTO-CIRCUITO | | | | **5a. PROCESSO SEI DO P&D**  - |
| **5b. NÚMERO PROJETO P&D**  - |
| **6. AUTOR(ES)**  THIAGO HENRIQUE ARBUINI RODRIGUES | | | | **5c. ETAPA DO PROJETO**  TODAS |
| **5d. TIPO DE PRODUTO**  DOCUMENTAÇÃO TÉCNICA DE SOFTWARE DE DISCIPLINA STR |
| **7. ENDEREÇO** | | | | **8. NÚMERO DO DOCUMENTO** |
| AV. JOÃO NAVES DE ÁVILA, 2121, BLOCO 3N – UBERLÂNDIA - MG | | | | **TR-****01** |
| **9. DISTRIBUIÇÃO DESTE DOCUMENTO**  DISTRIBUIÇÃO ABERTA A TODOS OS INTERESSADOS. | | | | |
| **10. NOTAS COMPLEMENTARES**  - | | | | |
| **11. RESUMO**  ESTE DOCUMENTO DESCREVE O FUNCIONAMENTO DE UM MÓDULO DE UM SISTEMA SUPERVISÓRIO PARA O SETOR ELÉTRICO: UMA APLICAÇÃO DE IDENTIFICAÇÃO DE CURTO-CIRCUITO COM FUNÇÕES DE PROTEÇÃO ANSI. | | | | |
| **12. PALAVRAS-CHAVE**  P&D; IOT; SISTEMA EM TEMPO REAL, MERGE UNIT, SISTEMA SUPERVISÓRIO, MEDIÇÃO EM SUBESTAÇÕES DE ENERGIA. | | | | |
| **13. CLASSIFICAÇÃO SEGURANÇA:**  **ABERTA** | **14. NÚMERO DE PÁGINAS**  - | | **15. NOME DO RESPONSÁVEL PRINCIPAL E CONTATO**  THIAGO HENRIQUE ARBUINI RODRIGUES  EMAIL: tharbuini@gmail.com | |

****

**HISTÓRICO DE VERSÕES DESTE TR**

**Tabela 1** – *Histórico de versões deste reporte técnico.*

|  |  |  |
| --- | --- | --- |
| **Versão** | **Data** | **Modificações** |
| 1.0 | março/2024 | * Principais elementos de projeto * Requerimentos básicos * Modelagem de pacotes e fluxo de pacotes * Interfaces básicas |

|  |
| --- |
| **SUMÁRIO** |
| [RESUMO GERAL 4](#_Toc158894411)  [1 – Introdução: visão geral da solução 4](#_Toc158894412)  [1.1 – Propósito e escopo 4](#_Toc158894413)  [1.2 – Produto: perspectivas e funções 5](#_Toc158894414)  [1.3 – Restrições do produto e considerações 6](#_Toc158894415)  [2 – Requisitos 7](#_Toc158894416)  [2.1 – Cenários de uso 7](#_Toc158894417)  [2.2 – Requisitos e validação 8](#_Toc158894418)  [2.3 – Versionamento 9](#_Toc158894419)  [2.4 – Elementos de projeto 10](#_Toc158894420)  [2.4.1 – Módulo de medição 10](#_Toc158894421)  [2.4.2 – Máquina de estados 10](#_Toc158894422)  [2.4.3 – Interfaces de usuário 11](#_Toc158894423)  [2.5 – Interfaces de comunicação e infra de TIC 12](#_Toc158894424)  [3 – Modelagem-métodos 13](#_Toc158894425)  [3.1 – Blocos de elementos principais 13](#_Toc158894426)  [3.2 – Tabela geral de objetos IPSO e recursos de URI 15](#_Toc158894427)  [3.3 – Modelagem de recursos 15](#_Toc158894428)  [3.3.1 – Eventos de carregamento da tela principal 15](#_Toc158894429) |

‘

# RESUMO GERAL

Este reporte técnico aborda os elementos do sistema identificador de curto-circuito que compõe a solução de sistema supervisório para o setor elétrico. O software a ser desenvolvido é responsável por ler pacotes da rede, com medições de correntes fornecidas por um módulo gerador de pacotes, e identificar curto-circuito seguindo a função 51 da tabela ANSI (sobrecorrente temporizada) e curva de curto-circuito IEC. Esse módulo é apenas uma parte do sistema supervisório.

# 1 – Introdução: visão geral da solução

## – Propósito e escopo

O sistema a ser desenvolvido tem o objetivo de, ao receber pacotes da rede contendo informações de corrente, analisá-las usando normas e curvas de curto-circuito e identificar eventos na rede.

|  |
| --- |
| ***Módulo 1:***  *Hardware de medição contínua MU*  **Módulo 2:**  Identificação curto-circuito IEC e funções de proteção  ***Módulo 4:***  *Filtragem de parâmetros/ eventos em tempo real*  ***Módulo 3:*** *Monitoramento visual séries históricas, parametrização e info. Outros módulos*  **Módulo 5:**  *IEDs atuadores* |
| **Figura 1.1.1:** Visão geral de escopo. |

Desta maneira, ainda que o *Módulo 2* tenha função de identificar evento de curto-circuito na rede, também atenderá características do módulo de atuação, em particular o *Módulo 5 – IEDs atuadores*.

É importante ressaltar que normalmente há mais de um dispositivo enviador de pacotes a ser observado. Portanto, o módulo deve ser capaz de monitorar tais dispositivos simultaneamente e identificar (ou não) os eventos que ocorrerem. A seguir é mostrado um exemplo da distribuição lógica de uma subestação:

|  |
| --- |
| Electronics | Free Full-Text | A Survey on Vulnerabilities and  Countermeasures in the Communications of the Smart Grid |
| **Figura 1.1.2:** Visão geral de cenário. |

## 1.2 – Produto: perspectivas e funções

Este produto tem como função monitorar a rede através da análise de pacotes recebidos com a maior precisão possível e atendendo requerimentos de tempo nas ocasiões especiais de eventos elétricos. As funções do sistema são:

1. Fazer análise a cada pacote recebido;
2. Em caso de evento, enviar o pacote de emergência em broadcast por rede ethernet em IP (confiável);
3. Responder as demandas dos requerimentos de tempo da norma IEC61850.

|  |
| --- |
|  |
| **Figura 1.2.1:** *Principais elementos do projeto.* |

Para entender o sistema, comecemos a análise observando a **Figura 1.2.1**. Com base nisto, descreve-se os elementos:

* **Módulo hardware gerador de dados:**  envia dados de medidas de corrente periodicamente para a rede.
* **Módulo algoritmo de análise das medições:**  faz a análise dos valores de corrente obtidos do primeiro módulo e identifica se houve um evento na rede através da curva de curto-circuito IEC.
* **Módulo gerente de pacotes:**  através dos dados analisados, monta e envia pacotes de alarme para a rede conforme urgência.

## 1.3 – Restrições do produto e considerações

A solução geral aqui prevista foi testada para condições específicas e nestas, foram identificadas as seguintes restrições ou limitações para os quais o sistema proposto não foi projetado para atuar. Estas restrições e limitações são mostradas na tabela da sequência.

**Tabela 1.3.1:** *Restrições e limitações previstas para sistema.*

|  |  |  |
| --- | --- | --- |
| **Nº** | **Restrição/limitação** | **Descrição/detalhamento** |
| 1 | O sistema é voltado apenas para SE de energia que seguem o padrão IEC 61850 | Ele se aplica somente as normas de SE de média tensão da IEC, especialmente a IEC 61850, para tempos de atuação, arquitetura, configuração e formatação de dados. |
| 2 | O sistema é focado apenas para bay de processos | Sua medição e normalização é para atuação em transformadores de potência, disjuntores AT/MT, chaves seccionadoras e cubículos de medição. |
| 3 | O sistema precisa de um computador industrial instalado dentro da SE | Para execução é necessário que no ambiente da SE exista um computador industrial apto a operar neste tipo de ambiente e com os mecanismos de proteção corretos. Ainda, ter interfaces de comunicação com latência desprezível. |
| 4 | Sem ausência de redundância ou WDT | Caso o software falhe, não existe nenhum mecanismo de contingência dele. |
| 5 | O sistema emula no máximo 10 dispositivos | O sistema proposto consegue emular só até 10 unidades de ‘dispositivos identificadores’. |

# 2 – Requisitos

## 2.1 – Cenários de uso

Os seguintes cenários foram identificados para este sistema.

1. **Cenário 1 – operação em condições normais**: nele quando não há nenhum evento, o sistema opera em condições normais seguindo a sequência de passos indicada na figura da sequência. Ela ilustra como deve ser seu comportamento.

|  |
| --- |
|  |
| **Figura 2.1.1:** *Cenário de aplicação.* |

1. **Cenário 2 – operação em transitórios elétricos e/ou curto-circuito**: nele, o “algoritmo de comportamento” verificou que houve uma mudança abrupta na rede elétrica. Pode ser várias coisas. Uma delas é um princípio de curto-circuito (ainda a confirmar), ou um transitório ou qualquer coisa que aponte para uma mudança da rede, mas que ainda não seja possível indicar que de fato é um evento da rede.

|  |
| --- |
|  |
| **Figura 2.1.2:** *Cenário de aplicação.* |

## 

## 2.2 – Requisitos e validação

Com base nas avaliações de cenário de uso, desenvolveu-se na sequência a seguinte lista de requerimentos, vista na tabela da sequência.

**Tabela 2.2.1:** *Mapa de requerimentos.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Classe/**  **Componente** | **Nº req.** | **Requisito** | **Origem requisito** | **Prior.** | **Tipo validação** |
| 1 - Robustez | 1.1 | Ter alimentação auxiliar para não cair (bateria) | Não parar de funcionar durante eventos elétricos extremos | 1 | Testar em cenário comutando desligamento das fontes por X vezes e analisar disponibilidade do serviço. |
| 1.2 | Ter redundância na alimentação | 1 | Usar uma fonte AC e outra fonte DC. |
| 2 - Funcional | 2.1 | Recebe medidas de corrente por rede ethernet em taxas aceitáveis para normas proteção | Norma IEC | 1 | Implantar um módulo de analog front end e validar medidas usando maleta de relé para geração de sinais. |
| 2.2 | Evitar sobrecarregar a rede com pacotes (QoS) (taxa perda < 20%) | Um dispositivo em alarme não deve interferir em outro dispositivo. | 1 | Testar eventos usando um switch 10/100Mb com 10 dispositivos em evento de curto e ver a taxa de erro de pacotes |
| 2.3 | Na possibilidade de evento, mandar informações o ‘mais rápido possível’ visando garantir um tempo de atuação de sistema (recebimento de pacote + identificação) inferior a 2 ciclos 60Hz (<32ms) | Norma IEC | 1 | Fazer um recurso no software que aumenta o valor de corrente de modo que os outros módulos identifiquem este aumento, mandem as medidas instantaneamente e criamos uma rotina para medir tempo de envio e recebimento que deve ser inferior a 10% do tempo indicado como limite (~3ms) |
| 2.4 | A aplicação deve suportar 10 dispositivos emissores de pacotes | - | 2 | Fazer um outro módulo de software de teste que identifique os pacotes na rede e seja capaz de contabilizar a quantidade de dispositivos |
| 3 – Não funcionais | 3.1 | Alto MTBF (superior a 1 falha/ano com DEC < 2 mim) | Alta disponibilidade | 2 | Simulação em software |
| 3.2 | Sistema deve contribuir para decrementar DEC e FEC evitando falsos positivos | ANEEL | 2 | Avaliações feitas pelos usuários com base em cenários estatísticos contabilizando-se o tempo (e frequência) de desligamentos desnecessários em que o sistema aqui proposto poderia atuar evitandos tais desligamentos |

## 2.3 – Versionamento

Os recursos do software são distribuídos em versões conforme estimado pela tabela na sequência.

**Tabela 2.3.1:** *Tabela de recursos do sistema e versão.*

|  |  |
| --- | --- |
| **Versão** | **Recurso** |
| **1.0**  *(mar/24)* | ■ Receber pacotes da rede  ■ Analisar segundo a curva de curto-circuito IEC |
| **1.1**  *(abril/24)* | **□** Plotar visualmente os dados enviados e recebidos  **□** Ter suporte a mais de um dispositivo gerador de pacotes  **□** Fazer análise multithread  **□** Enviar pacote alerta com o número de dispositivo  **□** Estimar QoS |

## 2.4 – Elementos de projeto

### 2.4.1 – Máquina de estados

Baseado nos cenários identificados e requerimentos construídos, tem-se a seguinte proposição para a máquina de estados de uma MU.

|  |
| --- |
|  |
| **Figura 2.4.2.1:** *Máquina de estados de uma unidade de monitoramento.* |

Todavia, essa unidade de monitoramento é executada dentro uma aplicação que pode ter mais de uma. Logo, o sistema também deve ter estes recursos:

1. Criação dinâmica de unidades
2. Recepção de dados da rede
3. Estimador de QoS e métricas de tempo
4. Avaliação de disponibilidade de rede

Esses recursos serão incluídos na segunda versão do projeto.

## 2.4.3 – Interfaces de usuário

Para fins de caracterização do sistema, a figura na sequência ilustra a interface desta aplicação indicando alguns de seus recursos previstos.

|  |
| --- |
|  |
| **Figura 2.4.3.1:** *Interface principal da aplicação* |

Na sequência, uma breve descrição destes principais elementos:

|  |  |
| --- | --- |
| ❶ | Caixas de texto identificando: corrente, tempo máximo de espera (atuação), timer em funcionamento (provisório para uma thread) e o IP de conexão. |
| ❷ | Plot de gráfico para visualização dos dados recebidos (provisório para uma thread). |

## 2.5 – Interfaces de comunicação e infra de TIC

O sistema poder ter duas interfaces de rede que podem ser usadas para dois cenários: (i) redundância ou (ii) cada interface alimenta separadamente um grupo de IEDs separados por VLANs.

No último caso, vale lembrar que o tráfego produzido pela interface 1 não chega aos dispositivos que estão ligados a VLAN 2. Logo, os tráfegos são separados pelas VLANs, evitando degradação de QoS. Ao mesmo tempo, damos ao MU a capacidade de operar com múltiplos MUs e múltiplas medições elétricas dentro de uma SE, porém, em um mesmo dispositivo, se observada a arquitetura definida na figura da sequência.

|  |
| --- |
| **Sistema MU**  **Inter. ethernet 1 ... N**  **Inter. ethernet N+1... N+M**  VLAN1  SWITCH DENTRO SE  VLAN 2  IEDs ou apps  IEDs ou apps |
| **Figura 2.5.1 –** Ligação básica das interfaces de rede do sistema MU. |

# 3 – Modelagem

## 3.1 – Blocos de elementos principais

Na sequência é mostrado um conjunto de diagramas de blocos para exemplificar a arquitetar do sistema. Cada bloco é um objeto e estes são os principais objetivos previstos na solução. As setas indicam o fluxo das informações.

|  |
| --- |
| **COMPUTADOR INDUSTRIAL**  **Criação dinâmica MUs**  - Cria, para e configura parâmetros das MUs virtuais  **Hardware medição MU1**  - ver módulo medição  **Hardware medição MU1**  - ver módulo medição  **Hardware medição MU1**  - ver módulo medição  **THREAD(MáquinaEstadoMU 1)**  - ver algoritmo máquina estado  **THREAD(Máquina Estado UM 2)**  - ver algoritmo máquina estado  **THREAD(Máquina Estado UM 3)**  - ver algoritmo máquina estado  **THREAD (Recepção dados rede)**  - Recebe pacotes IP na porta 333  - faz despacho dos pacotes  **Timer (estimador QoS)**  - estoura a cada 30seg - envia 10 pacotes UDP e recebe eles medindo taxa perca e latência, principalmente  **WatchDogTimer e keep alive de MUs**  - a definir  **MockUp medidas para MUs**  -  **MockUp medidas para MUs**  -  **MockUp medidas para MUs**  -  **ParametrizacaoRemota()**  - Recebe JSON com configuração de parâmetros de MU e faz o ‘set’ de seus parâmetros em tempo real  Cria, configura e exclui MU  switch  Socket IP  IP: xxx porta 333 |
| **Figura 1.2.1:** Componentes básicos do Digital Twins. |

## 3.2 – Tabela geral de objetos IPSO e recursos de URI

Todos os recursos previstos entre as aplicações do TWINS são feitos por códigos IPSO[[1]](#footnote-1) em padrão URI. Eles podem ser a designação de uma conexão para um evento, o formato de um pacote, o formato de um JSON específico dentre outras coisas. Ajudam a racionalizar e identificar facilmente o recurso e a estruturar o pensamento. Estes códigos estão dispostos na tabela da sequência.

**Tabela x.:** *Tabela IPSO de recursos do projeto.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objeto** | **Recurso** | | **Significado** | **Link** |
| **Nível 1** | **Nível 2** |
| **99**  *Mensagens com medidas* | **1**  Medidas periódicas |  | Medidas elétricas trafegando na rede quando o MU está operando em condições normais. |  |
| **2**  Medidas em situação alteração |  | Medidas elétricas trafegando na rede quando o MU identificou uma grande mudança nos parâmetros e ai começa a transmitir mensagens com maior frequência. |  |
| **100** *Controle e configuração* | **10**  Controle do protocolo MU |  | Mensagens onde um outro programa pode controlar alguns recursos do protocolo de medição do MU. |  |
| **11**  Configura parametrização MU |  | Mensagens onde um outro programa pode alterar remotamente alguns parâmetros do algoritmo do MU. |  |

## 3.3 – Fluxo geral de mensagens

A figura na sequência ilustra resumidamente as mensagens que são trocadas, em diferentes circunstâncias, entre o sistema MU e demais módulos.

|  |
| --- |
| **Módulo 2 – identificação curto-circuito**  - recebe medida - AO IDENTIFICAR CURTO, ENVIA PACOTE CONTROLE AO MU INDICANDO RECONHECIMENTO  Controle protocolo  100/10  **MU**  - GERA MEDIDAS SEGUNDO MODELO DE MÁQUINA DE ESTADOS PROPOSTA  Medidas 99/1 ou 99/2  **Módulo 3 – Monitoramento visual e parametrização**  - RECEBE DADOS E PLOTA - PERMITE AO USUARIO PARAMETRIZAR DADOS DO MU REMOTAMENTE  Parametrização 100/11 |
| **Figura 3.3.1:** Diagrama ilustrando a ideia de como o front tem suas informações atualizadas dinamicamente |

Como observado, o MU envia para a rede em regime de broadcasting o pacote 99/1 (quando ele não percebe nenhuma alteração na rede significativa) ou o 99/2 (quando ele percebe uma modificação nos parâmetros elétricos e envia as medidas mais rapidamente).

Também, o MU recebe informações de outras aplicações. Uma delas é pacote de controle (pacote 100/10) que pode ser enviado por outro módulo solicitando a ele alguma mudança no seu comportamento de envio de mensagens, como por exemplo um comando avisando que ele pode parar de enviar 99/2 porque o evento já foi reconhecido e não faz mais sentido enviar tantos pacotes para a rede. Uma segunda opção seria um pacote de parametrização onde outra aplicação daria a oportunidade de configurar os parâmetros do algoritmo (usando o pacote 100/11) do MU remotamente por um operador.

Para entender melhor esta dinâmica, elas são tratadas individualmente com mais detalhes nas próximas subseções.

## 3.4 – Modelagem detalhada dos recursos

### 3.4.1 – Envio de mensagens de medição

A figura na sequência ilustra como deve ser o comportamento temporal da dinâmica de envio de pacotes de medidas elétricas do MU em dois diferentes cenários: (i) sem mudanças abruptas nas variáveis elétricas medidas e (ii) com mudanças abruptas.

|  |
| --- |
| **MU**  **IED qualquer ou módulo 2**  **CENÁRIO: OPERAÇÃO EM CONDIÇÕES NORMAIS**  **Evento1:** envio de mensagens  **99/1**  *... repete a cada 50ms ...*  **99/1**  **CENÁRIO: OPERAÇÃO COM GRANDE MUDANÇA VALOR**  **Evento2:** identificador curto  **99/2**  *... repete a cada 1ms com taxas incrementais tempo ...*  **99/2**  **100/10**  **99/1**  *... repete a cada 50ms ...* |
| **Figura 3.4.1.1:** Diagrama pacotes e eventos associados ao envio de mensagens. |

Com base no diagrama mostrado na figura anterior, descreve-se agora o comportamento dos principais eventos identificados.

**Tabela 3.4.1.1:** *Algoritmos equivalentes aos eventos vistos na Figura 3.4.1.1.*

|  |  |
| --- | --- |
| **Evento/Ação** | **Algoritmo** |
| **Evento 1**  (envio de mensagens) | **Este evento deve implementar a máquina de estado mostrada na figura Figura 2.4.2.1. Algoritmo básico:**   1. Switch(MáquinaEstadoMU.Estado)    1. Case Estado.E1:       1. Implementar aqui os comandos vistos no estado E1 da Figura 2.4.2.1       2. MáquinaEstadoMU.Estado = Estado.E2 (ou outro?)    2. Case Estado.E2:       1. Implementar aqui os comandos vistos no estado E1 da Figura 2.4.2.1       2. MáquinaEstadoMU.Estado = Estado.E3 (ou outro?)   **....**   * 1. Case Estado.E3:      1. Implementar aqui os comandos vistos no estado E1 da Figura 2.4.2.1      2. MáquinaEstadoMU.Estado = Estado.E1 (ou outro?) |

Já o formato do pacote de dados é mostrado na sequência.

**Tabela 3.4.1.2:** *Formato do pacote de dados 99/1*

|  |  |  |
| --- | --- | --- |
| **Campo** | **Valores** | **Significado** |
| URI | **99/1** | Pacote de envio de medidas regulares |
| idMU | int | Identificador de qual MU está gerando este pacote |
| idAtivo | String (opcional) | Identificado de qual ativo o ID está monitorando. |
| numPct | inteito | Número do pacote gerado incrementalmente |
| timpeStamping | Data UMC | Data e hora em que foi gerado o pacote |
| freqEnvioMS | int | Taxa de aquisição e intervalo de tempo em que foram coletas as medidas e calculada a media. |
| medidas | [MedidasEletricas] | Contem um array com objetos do tipo “MedidasEletricas” **médias** que tem o seguinte formato:   |  |  | | --- | --- | | fase | String (A,B,C,N) | | tensao | float | | corrente | float | | angTensao | Float | | potApaVA | float | | potReatVAr | float | | potRealW | Float | | fatorP | float | | freq | float | |

E nas ocasiões onde acontecem grandes variações, gera-se no lugar do 99/1 o pacote 99/2 que tem mais informações e deve ter mais prioridade na rede.

**Tabela 3.4.1.3:** *Formato do pacote de dados 99/2*

|  |  |  |
| --- | --- | --- |
| **Campo** | **Valores** | **Significado** |
| URI | **99/2** | Pacote de envio de medidas urgentes com grande variação |
| idMU | int | Identificador de qual MU está gerando este pacote |
| idAtivo | String (opcional) | Identificado de qual ativo o ID está monitorando. |
| numPct | inteito | Número do pacote gerado incrementalmente |
| timpeStamping | Data UMC | Data e hora em que foi gerado o pacote |
| medidas | [MedidasEletricas] | Contem um array com objetos do tipo “MedidasEletricas” **instantâneas** que tem o seguinte formato:   |  |  | | --- | --- | | fase | String (A,B,C,N) | | tensao | float | | corrente | float | | angTensao | Float | | potApaVA | float | | potReatVAr | float | | potRealW | Float | | fatorP | float | | freq | float | |
| variavelDescrepante | [String] | Indica qual é a variável que sofreu a grande variação e acarretou a ocorrência deste pacote. |
| faseDescrepante | [String] | Indica qual é a fase que sofreu a grande variação e acarretou a ocorrência deste pacote. |

Temos também o pacote de controle que a MU pode receber e tomar alguma providência, segundo sua máquina de estados. Seu formato de dados é visto na sequência.

**Tabela 3.4.1.4:** *Formato do pacote de dados 100/10*

|  |  |  |
| --- | --- | --- |
| **Campo** | **Valores** | **Significado** |
| URI | **100/10** | Pacote de envio de controle do protocolo de medição. |
| idMU | int | Identificador de qual MU está gerando este pacote |
| idAtivo | String (opcional) | Identificado de qual ativo o ID está monitorando. |
| comando | int | Indica o comando de uma aplicação terceira está pedido para a MU executar.   * 0 = indica que o objeto foi identificado e que ele pode voltar a sua condições de normalidade ou que providencias de tratamento de eventos já foram tomadas. * 1 = incrementar taxa de envio * 2 = decrementar taxa de envio segundo o valor do campo “parametro1” |
| parametro1 | int | Usando para indicar algum parâmetro numero de algum comando. |
| parametro2 | string | Usando para indicar algum parâmetro texto de algum comando. |

### 3.4.2 – Parametrização remota

A figura na sequência ilustra como deve ser a parametrização remota.

|  |
| --- |
| **MU**  **módulo 3**  **Evento2:** reconfiguração  **100/11**  **Evento1:** interface gráfica enviado novos parâmetros  **100/11.reconhecimento=true** |
| **Figura 3.4.2.1:** Diagrama pacotes e eventos associados à reconfiguração do MU |

Com base no diagrama mostrado na figura anterior, descreve-se agora o pacote de reconfiguração.

**Tabela 3.4.2.1:** *Algoritmos equivalentes aos eventos vistos na Figura 3.4.1.1.*

|  |  |
| --- | --- |
| **Evento/Ação** | **Algoritmo** |
| **Evento 2**  (envio de mensagens) | 1. monta pacote 100/11 2. Verifica chegada de confirmação    1. Aguarda 100ms    2. Se pacotesEnviados < 3       1. Envia novo pacote       2. Incrementa “pacotesEnviados++”    3. Senao       1. Parametrização falhou. Mostra na interface gráfica falha da parametrizacao |

Já o formato do pacote de dados é mostrado na sequência. Como pode ser visto, a principal limitação é que para cada parâmetro deve ser enviado 1 pacote. Uma modificação seria tornar os campos “parametro” e “valor” como vetores, permitindo assim múltiplas parametrizações em um único pacote.

**Tabela 3.4.1.2:** *Formato do pacote de dados 100/11*

|  |  |  |
| --- | --- | --- |
| **Campo** | **Valores** | **Significado** |
| URI | **100/11** | Pacote de envio de parametrização da MU |
| idMU | int | Identificador de qual MU está gerando este pacote |
| parametro | String | Nome do parâmetro |
| valor | float | Novo valor que a variável contida em “parâmetro” deve valer. |
| reconhecimento | bool | É o ACK da operação |

1. https://www.avsystem.com/blog/IPSO-smart-objects/ [↑](#footnote-ref-1)