

## Sincronização de dados a partir de um módulo GPS para unidades de medição fasorial sincronizadas

### Data Synchronization from a GPS module to synchronized phasor measurement units

#### RESUMO

**Cristian Roberto Pastro**  
[Cristian-pastro@hotmail.com](mailto:Cristian-pastro@hotmail.com)  
Universidade Tecnológica Federal do Paraná, Pato Branco, Paraná, Brasil

**Gustavo Weber Denardin**  
[gustavo@utfpr.edu.br](mailto:gustavo@utfpr.edu.br)  
Universidade Tecnológica Federal do Paraná, Pato Branco, Paraná, Brasil

PMU é um dispositivo utilizado para mensurar magnitude e ângulo de fasores elétricos, utilizando uma fonte de tempo comum para sincronização. Essa base de tempo pode ser obtida de forma precisa através de sistemas de posicionamento global (GPS), que fornecem informações temporais precisas e comuns no mundo inteiro. O presente trabalho visa a implementação de uma PMU de baixo custo, com foco na sincronização dos dados de tempo recebidos do GPS com os dados de medição obtidos da rede elétrica. Este trabalho propôs uma fila para sincronizar dados fasoriais e temporais. A PMU proposta envia as informações a um concentrador de dados por meio de uma interface *ethernet* assim que as informações temporais estão disponíveis. O envio se dá pelo uso do protocolo IEEE C37.118.2-2011. As informações são coletadas por meio de software para análise posterior. Resultados experimentais mostram que há uma correta sincronia entre os dados coletados e a base de tempo obtida através do GPS.

**PALAVRAS-CHAVE:** Sistema de Posicionamento Global. Medição. Fase.

**Recebido:** 19 ago. 2019.

**Aprovado:** 01 out. 2019.

**Direito autoral:** Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



#### ABSTRACT

PMU is a device used to measure the magnitude and phase angle of an electrical phasor in the power grid, using a common time source for synchronization. This time base can be obtained through global positioning systems (GPS), which provides accurate and common information worldwide. The present work aims at the implementation of a low cost PMU, focusing in the synchronization of the time data received from the GPS with the measure data obtained from the power grid. This work proposed a queue to synchronize phasor and time data. The proposed PMU sends the phasor data to a phasor data concentrator through an ethernet interface as soon as the data receives its time tag, by the use of the IEEE C37.118.2-2011 protocol. The information is collected through computer software for further analysis. Experimental results shows that there is a correlation between data and a time base through GPS.

**KEYWORDS:** Global Positioning System. Measurement. Phase.

## INTRODUÇÃO

PMU (*Phasor Measurement Unit*) é um dispositivo utilizado para mensurar fasores. Aos fasores são anexadas informação de hora e data (GUPTA et al. 2006). Fasor é um número complexo contendo o valor de raiz quadrática média da magnitude, e o ângulo de fase na forma polar da forma de uma onda sinusoidal elétrica de tensão ou corrente (GUPTA et al. 2006). Fasores medidos em uma base de tempo comum são chamados de sícrofasores (KUMAR, 2016). A base de tempo pode ser obtida com precisão por meio de um sistema de um sistema de GPS (*Global Positioning System*), pois estes possuem referência temporal precisa e comum ao mundo todo, obtida por meio do relógio atômico dos satélites

É possível formar um Sistema de Medição Fasorial Sincronizada (SMFS) por meio de uma rede de PMUs espalhadas pela rede elétrica. As medidas são enviadas a um *Phasor Data Concentrator* (PDC) responsável pelo processamento e armazenamento das informações recebidas (GRANDO, 2016).

Com o uso de PMUs, é possível observar o comportamento dinâmico da rede elétrica. Além disso, PMUs têm uso em redes inteligentes e sistemas de verificação de distúrbios em redes elétricas (GRANDO, 2016).

O presente trabalho visa a construção de uma PMU de baixo custo. A iniciativa já se encontrava em andamento, porém a sincronização dos dados de tempo com os dados oriundos da rede elétrica ainda não era garantida. Portanto, este trabalho objetiva a sincronização destes dados.

## MATERIAIS E MÉTODOS

Implementou-se a PMU em duas plataformas *DISCO* da *STMicroelectronics*: *STM32F746G* e *STM32F769NI*. Ambos os kits possuem um *St-link* embutido, facilitando a depuração em tempo real, além de uma interface *ethernet* inclusa. Realizou-se o desenvolvimento da PMU utilizando a linguagem C, por possuir um bom desempenho, várias bibliotecas prontas e ser de fácil portabilidade. Utilizou-se a IDE *Eclipse Neon* por possuir boas ferramentas gráficas para o desenvolvimento e depuração do código. O compilador escolhido para o projeto foi o GCC, por ser gratuito e possuir vasta documentação.

A PMU foi desenvolvida com o sistema operacional *FreeRTOS*. A utilização de um RTOS em um sistema embarcado facilita a programação de multitarefas, além disso, o *FreeRTOS* é um sistema gratuito e com ampla documentação.

Utilizou-se o módulo GPS *V.KEL VK16U6*. Uma das vantagens deste módulo é que sua antena integrada elimina atrasos de comunicação com o receptor.

O módulo GPS *VK16U6* envia pulsos periódicos denominados PPS. Os pulsos são sincronizados satélites e sua frequência é configurável. Por padrão, o GPS emite um pulso por segundo. Logo após o pulso ser emitido, as informações temporais correspondentes são enviadas por meio de uma interface UART.

Foram utilizados também osciloscópios, geradores de função, multímetros e outros materiais de uso básico disponíveis nos laboratórios da universidade.

Para os testes, utilizou-se uma bancada com três geradores de função *Tektronix AFG3021C*. Os geradores foram configurados para emitir senoides que simulem a rede elétrica passando por um circuito de instrumentação. As senoides geradas possuem frequência 60 Hz, *offset* 900 mV e amplitude 1,5 V<sub>pp</sub>. Com o auxílio de um osciloscópio *TekTronix TBS 1062*, alterou-se a fase das senoides até que estivessem defasadas 120° entre as mesmas.

Após a bancada montada, um PDC foi concebido utilizando dois softwares. O primeiro é o *PMU ConnectionTester*, que permite verificar e testar a conexão entre PMU e PDC, além de permitir a exportação em formato CSV de cada pacote recebido. O segundo software é o *openPDC*, que permite a configuração completa de um PDC em uma rede local. Além disso, para uma melhor análise, os dados recebidos foram capturados utilizando o software *WireShark*.

A comunicação entre PMU e PDC foi realizada utilizando o protocolo IEEE C37.118.2-2011, por ser um protocolo comercial já utilizado em PMUs comerciais. São realizadas e transmitidas 30 medições fasoriais por segundo.

## ETAPAS DE DESENVOLVIMENTO

Como o projeto já se encontrava em andamento, inicialmente analisou-se todos os métodos e algoritmos implementados até o presente momento. Verificou-se a presença de uma fila que armazena informações de medições fasoriais enquanto não houver estampa de tempo disponível para ser associada.

Buscando falhas no atual processo, verificou-se que informações enviadas pelo GPS pela interface UART não eram recebidas com sucesso pela PMU. O erro verificado advinha de incompatibilidades nas configurações da comunicação entre microcontrolador e GPS. A configuração foi corrigida no microcontrolador.

A seguinte etapa de desenvolvimento focou-se no cálculo da estampa de tempo. Uma informação importante para o cálculo da estampa de tempo é o SOC. O SOC baseia-se na marca temporal *Unix*, que conta os segundos passados desde o dia 1 de janeiro de 1970 às 00:00:00 do Tempo Universal Coordenado (UTC) no calendário gregoriano proléptico. A metodologia escolhida para o cálculo do SOC verifica o tipo de pacote recebido do GPS e realiza validações a fim de impedir que pacotes inválidos sejam processados de forma errônea.

Pacotes contendo medições fasoriais devem incluir um campo denominado Fração de Segundo. Esse campo diferencia cada uma das 30 medições fasoriais realizadas durante um mesmo segundo. Implementou-se a fração de segundo na etapa seguinte de acordo com o protocolo IEEE C37.118.2-2011.

Quando um pulso emitido pelo GPS é identificado pela PMU, um novo processo de aquisição de dados da rede elétrica é realizado. Todas as medições fasoriais realizadas entre o pulso e a recepção da base de tempo correspondente a esse pulso, são armazenadas em uma fila. Assim que a base de tempo é recebida e processada, todas as medições fasoriais realizadas sem essa base de tempo devem ser atualizadas. Nesta etapa do desenvolvimento, alguns pontos do projeto original foram corrigidos, tais como:

- a) substituição das funções de alocação de memória da biblioteca *stdlib* pelas funções do *FreeRTOS*: tal substituição se faz necessária porque as funções originais da *stdlib* possuem alto tempo de execução, tempo de execução não determinístico, além de não ser *thread safety*;
- b) não utilização de funções que liberam regiões de memória alocadas dinamicamente: implica no possível travamento da PMU por falta de memória após certo tempo de execução;
- c) envio repetido de pacotes: um pacote depois de enviado permanecia na fila, sendo atualizado a cada um segundo com um SOC incoerente. Uma alteração na forma de manipulação da fila teve de ser feita.

Na etapa conseguinte, realizou-se a portabilidade do projeto para o kit contendo o microcontrolador STM32F769. A vantagem desse kit é que, diferente do outro, os três A/Ds do microcontrolador possuem conexão direta com os pinos disponíveis ao usuário, permitindo assim, mensurar as três fases da rede elétrica. As alterações a partir desse ponto foram realizadas apenas nesse kit.

Como o GPS é configurado por padrão para emitir apenas um pulso por segundo, as 29 medições fasoriais seguintes devem ser acionadas através de um *timer* interno do microcontrolador. Com o intuito de verificar uma possível melhora na precisão das medidas, realizou-se uma nova opção de configuração, na qual o GPS foi configurado para emitir 30 pulsos por segundo, cada pulso gerando uma medição fasorial.

Com essa alteração, a relação de sincronia entre o pulso e a base de tempo enviada pelo GPS é alterada. Com o auxílio de um osciloscópio e um módulo GPS auxiliar de mesmo modelo, verificou-se que a estampa de tempo é enviada próximo ao 27º pulso de cada correspondente SOC. Para inserir tal característica no código, adicionou-se uma macro para indicar qual pulso corresponde a primeira estampa de tempo recebida.

## RESULTADOS E DISCUSSÃO

Montou-se a bancada de testes utilizando o kit contendo o microcontrolador STM32F769 e a configuração de 1 PPS. A PMU exibiu a tela conforme mostra a Figura 1.

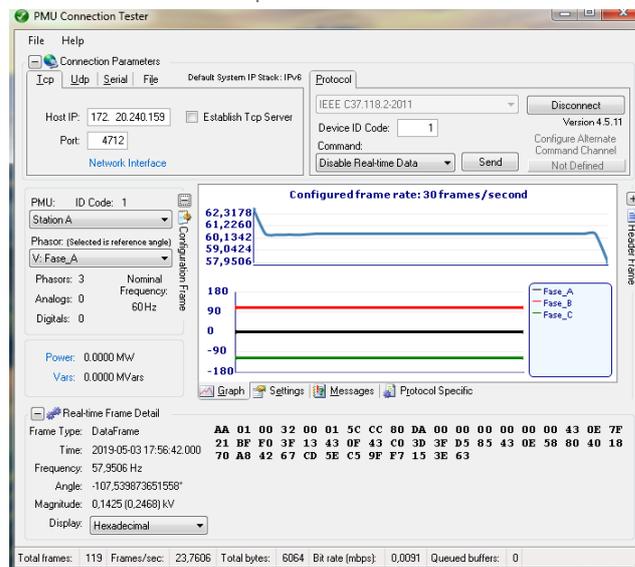
Figura 1 – Tela inicial da PMU após a conexão com a bancada de testes.



Fonte: Autoria Própria.

Depois de realizada a conexão através do software *PMU Connection Tester*, é exibida a tela conforme mostra a Figura 2.

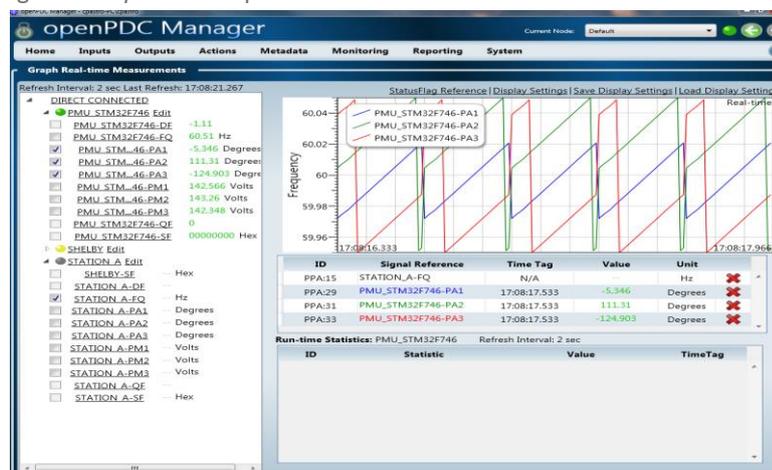
Figura 2 – PMU Connection Tester depois de conectado à PMU com a bancada de testes



Fonte: Autoria Própria.

Várias telas estão disponíveis no software *openPDC*. Uma das principais telas, mostrando várias informações importantes é mostrada na Figura 3.

Figura 3 – *openPDC* depois de conectado à PMU com a bancada de testes



Fonte: Autoria Própria.

As Figura 1 a 3 indicam a defasagem de cerca de  $120^\circ$  entre as fases, com pequenas diferenças devido à falta de um circuito de instrumentação. Esse resultado é esperado, pois é a mesma configuração realizada nos geradores de função.

Realizou-se uma análise mais detalhada no arquivo CSV gerado pelo software *PMU Connection Tester*, assim como nos pacotes capturados pelo *WireShark*. Na análise, verificou-se que o primeiro pacote de cada segundo possui fração de segundo zero. Ademais, cada pacote consecutivo possui fração de segundos diferenciado em 0.333 unidades. Além disso, recebe-se 30 pacotes por segundo. Tais características são esperadas no protocolo IEEE C37.118.2-2011, indicando o correto funcionamento e integração dos componentes do projeto.

## CONCLUSÃO

Este trabalho finalizou a criação de uma PMU de baixo custo, criando um protótipo funcional e fiel às especificações de projeto.

A utilização de um *Real Time Operating System* (RTOS) permite à PMU cumprir os prazos rígidos de aquisição, processamento e transmissão de dados. A utilização de kits que dispõe de interface *ethernet* e tela própria promove melhor integração e facilidade de programação destes componentes.

Verificou-se a dificuldade de realizar a correta sincronização na versão da PMU com 30 PPS. Ademais, mesmo usando macro para configurar a qual pulso corresponde a primeira estampa de tempo recebida, o problema da sincronização não é totalmente resolvido. É possível, por exemplo, que essa macro tenha de ser configurada de forma diferente para módulos GPS diferentes.

Esse problema não ocorre na com 1 PPS. Além disso, utilizar 1 PPS é mais viável, pois esta configuração é possível na maioria dos módulos GPS comerciais.

Em conjunto com o circuito de instrumentação já projetado anteriormente, a PMU pode ser ligada à rede elétrica, obtendo sinais reais. Tendo em vista estes pontos, é possível concluir que a PMU concebida por este projeto não é algo simplesmente teórico ou para fins acadêmicos. O presente projeto pode ser utilizado em ambientes reais e inclusive integrar uma rede de PMUs.

## REFERÊNCIAS

GUPTA, Abhishek; JAIN, Smriti; AGRAWAL, Vishal Kumar; AHAMED, Sohail; AGRAWAL, Saksham; KUMAR, Rishabh. Phasor measurement unit. **International Journal of Engineering and Management Research**, IJEMR, v. 6, n. 2, p. 221–224, 2006.

KUMAR, Jitender. Impact of phasor measurement unit on the state estimation of large power system. **International Journal of Advance Research and Innovation**, v. 4, p. 1–7, 03 2016.

GRANDO, Flavio Lori. **Arquitetura para o desenvolvimento de unidades de medição fasorial sincronizadas no monitoramento a nível de distribuição**. 2016. Dissertação (Programa de Pós Graduação em Engenharia Elétrica - PPGE) – Universidade Tecnológica Federal do Paraná, Pato Branco, 2016.

## AGRADECIMENTOS

Agradeço a Fundação Araucária e a Universidade Tecnológica Federal do Paraná pelo apoio financeiro destinado a este trabalho.