UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA

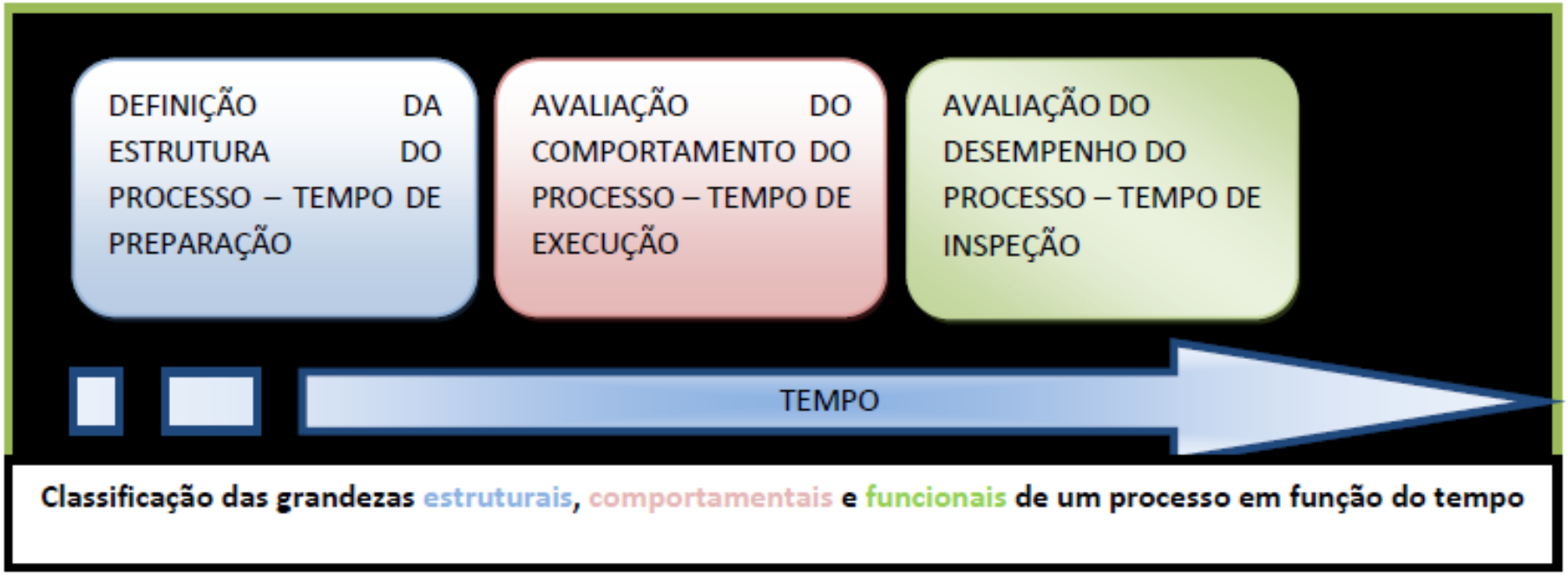
CURSO DE ESPECIALIZAÇÃO em ENGENHARIA INDUSTRIAL 4.0

Laboratório de usinagem: www.labusig.ufpr.br

Disciplina: Manufatura Inteligente

IMPLEMENTAÇÃO DO SISTEMA PARA MONITORAMENTO DE PROCESSOS DE USINAGEM VIA INTERNET

Monitoramento de processos de manufatura discreta

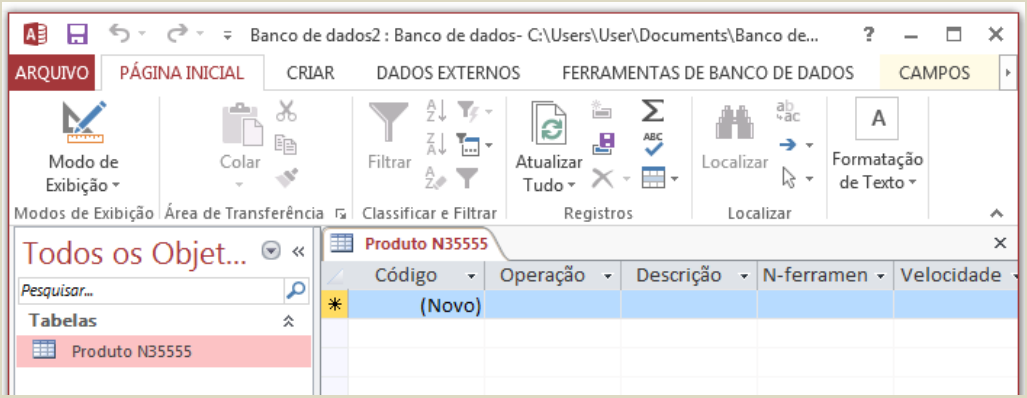


Classificação das grandezas estruturais, comportamentais e funcionais de um processo em função do tempo

MONITORAMENTO EM TEMPO DE PREPARAÇÃO

Conferência

do plano de processos



Base de dados contendo as grandezas estruturais e seus valores

Em máquinas equipadas com CLP/CNC é possível fazer a conferência dessas grandezas em tempo de execução

MONITORAMENTO EM TEMPO DE PREPARAÇÃO

grandezas programadas podem sofrer ligeiras modificações durante o tempo de execução.

Exemplos:

Fonte: https://blog.paraisodasbombas.com.br/entenda-o-giro-do-motor-eletrico-e-aprenda-a-inverter-a-rotacao/

Variação da rotação do motor em função da variação da carga

Rotação síncrona (Ns) é função da frequência (f) e do número de par de polos (p)

$$N_s = \frac{120f}{p}$$

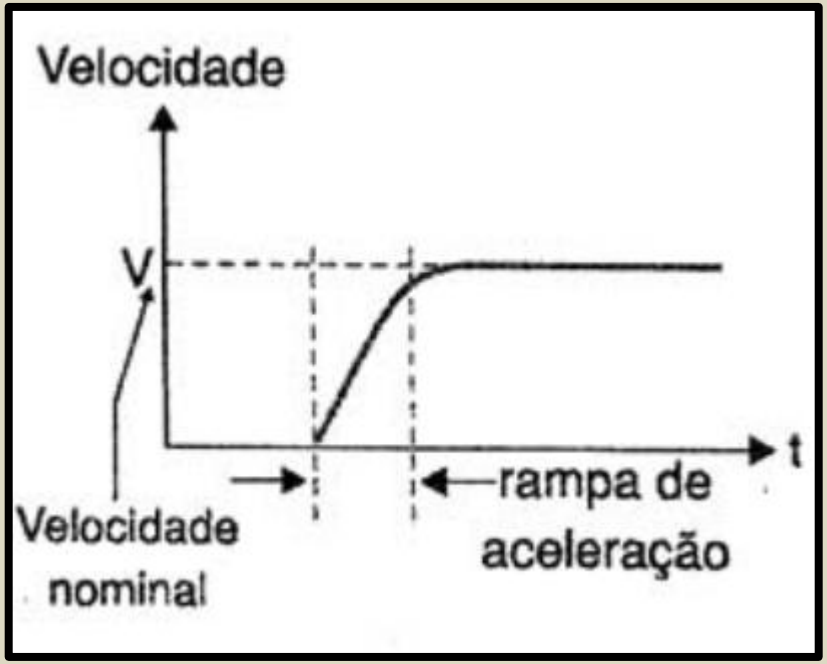
Escorregamento (ε) é função da carga suportada pelo motor em tempo de execução, o que afeta a sua rotação real (N_r) da seguinte maneira

$$N_r = N_s(1 - \epsilon)$$

grandezas programadas podem sofrer ligeiras modificações durante o tempo de execução.

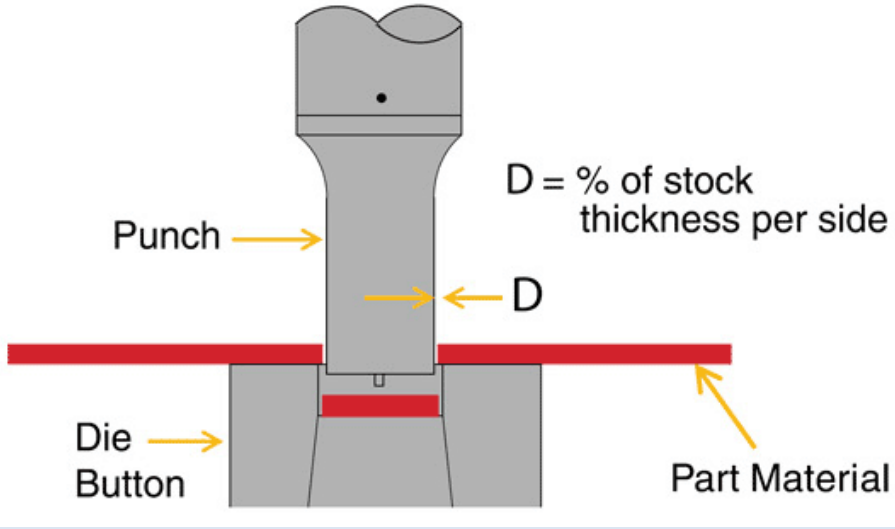
Exemplos:

Variação do tempo de posicionamento em função da variação das rampas de aceleração



Fonte:https://webautomacaoindustrial.blogspot.com/2017/10/os-inversores-de-frequencia.html

Variação do tempo de processo em função de desgaste/avarias do ferramental



Fonte:
https://www.daytonlamina.com/node/1050

MONITORAMENTO EM TEMPO DE INSPEÇÃO

Avaliação da qualidade final

Inspeção rápida: calibradores passa-não-passa

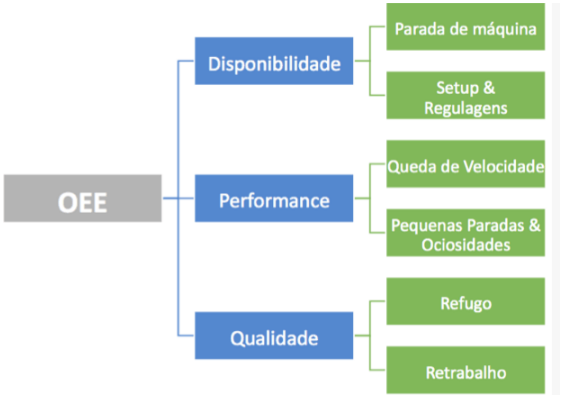
Inspeção de forma e rugosidade

Fonte https://www.taylor-hobson.com/products/roundness-form

MONITORAMENTO EM TEMPO DE INSPEÇÃO

Indicativos econômicos (tempo & custo)

OEE (**Overall Equipment Effectiveness**)



Fonte: https://www.prodwin.com.br/pt/blog/como-calcular-o-oee/

Fonte: https://www.dreamstime.com/mes-manufacturing-execution-system-concept-keywords-letters-icons-flat-vector-illustration-isolated-mes-manufacturing-image135042001

MONITORAMENTO EM TEMPO DE EXECUÇÃO

Requisitos:

- Medições em tempo real
- Controle adaptativo (correções em tempo real)
- Baixa interferência

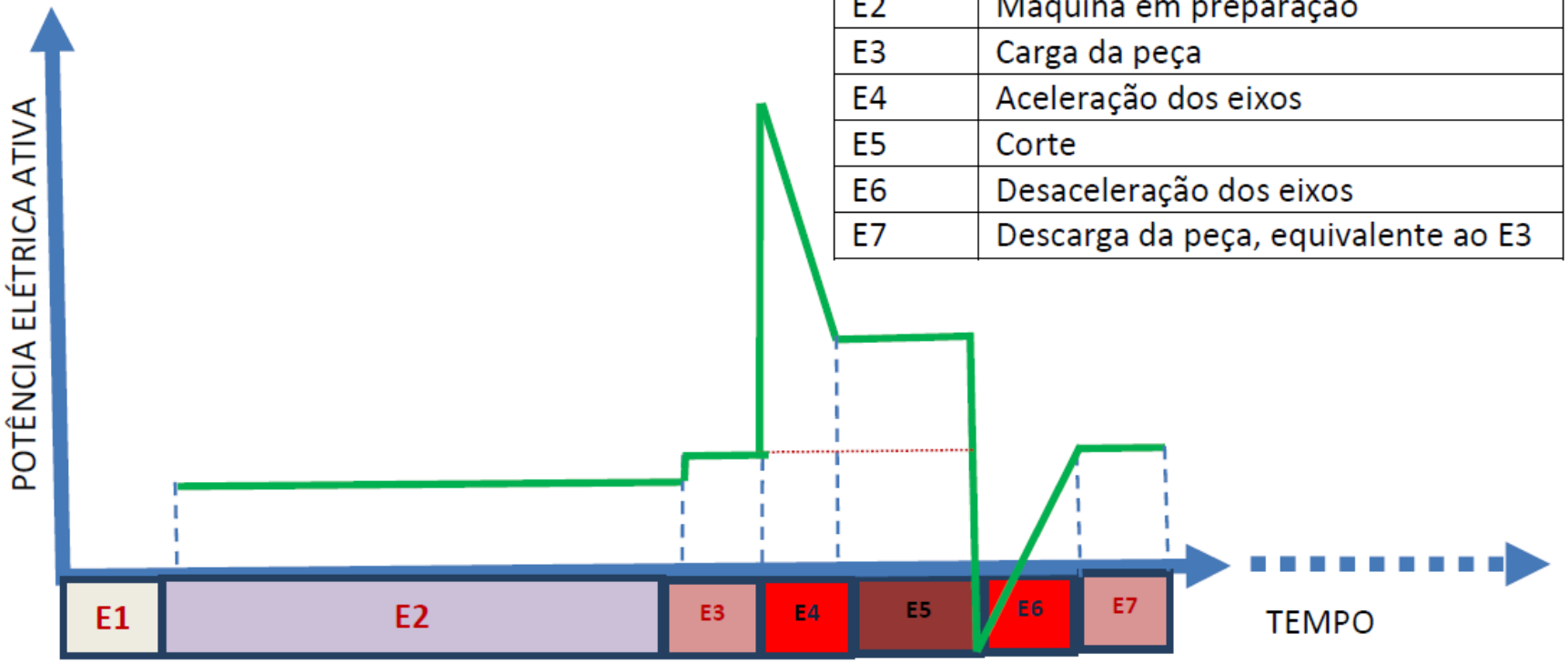
Fonte: Teti R, Jemielniak K, O'Donnel G, Dornfeld D (2010) Advanced Monitoring of Machining Operations. CIRP Annals Manufacturing Technology 59(2):717–739. https://doi.org/10.1016/j.cirp.2010.05.010

MONITORAMENTO EM TEMPO DE EXECUÇÃO

Técnicas	Interferência com o processo	Tempo de resposta	Confiabilidade	Processos
Desvios dimensionais	média	alto	alta	Usinagem, estampagem, conformação volumétrica
Avarias de ferramentas	média	alto	alta	Usinagem, estampagem, conformação volumétrica, injeção,...
Vibrações	baixa	muito baixo	média	Usinagem
Emissão acústica	baixa	muito baixo	alta em alguns casos	Usinagem, soldagem.
Análise por Imagens	baixa		em desenvolvimento. Baixa confiabilidade em alguns processos	Diversos
Forças	alta	baixo	alta	Usinagem, conformação.
Temperatura – infravermelho	baixa	baixo	em desenvolvimento. Baixa confiabilidade em alguns processos	Diversos
Temperatura - termopar	alta	baixo	alta	Diversos
Corrente/Potência elétrica	baixíssima	muito baixo	aceitável em alguns casos.	Soldagem, usinagem

Monitoramento em tempo real de processos de usinagem por meio da potência elétrica

Estágio	Descrição
E1	Máquina desligada
E2	Máquina em preparação
E3	Carga da peça
E4	Aceleração dos eixos
E5	Corte
E6	Desaceleração dos eixos
E7	Descarga da peça, equivalente ao E3

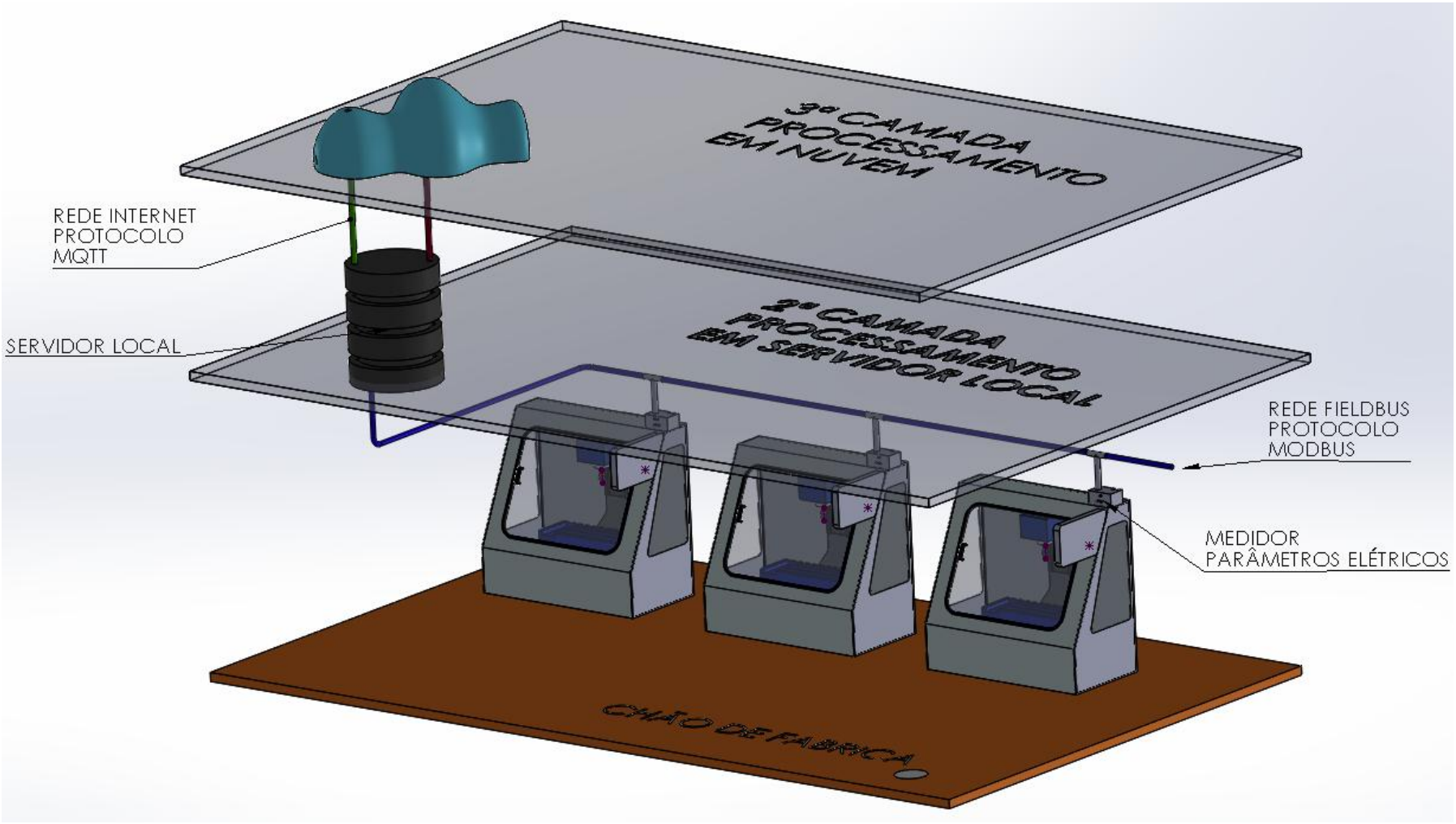


Implementação de um sistema para monitoramento via Internet da potência elétrica em processos de usinagem

Projeto MPEU

Sistema baseado em um multimedidor de parâmetros elétricos, uma rede tipo Fieldbus com protocolo Modbus-RTU e rede de Internet com protocolo MQTT

Projeto MPEU



Características do laboratório de usinagem da UFPR

- Diversificação razoável de máquinas
- Máquinas convencionais e CNCs
- A maioria do maquinário é composta por motores CA com alimentação trifásica em 220V. A exceção é o torno CNC (Mazak QTN100) que possui transformador interno para alimentação do motor EM 400V e demais servodrives e periféricos em 220V
- Além disso, os cabos de alimentação das máquinas possuem diâmetros e comprimentos diferentes.

Em função essa variação na alimentação elétrica das máquinas e na diversidade de acionamentos (conversores de energia) presentes, optou-se pela utilização de um sistema simples para monitoramento baseado em instrumentos universais (genéricos) ligados em uma rede do tipo Fieldbus.

Instrumento para medição da potência elétrica

Neste projeto, optou-se por montar o sistema de monitoramento de usinagem baseado na medição de parâmetros elétricos (tensão, corrente, fator de potência, potência e energia ativa). Para isso, vem sendo utilizados multimedidores da marca Kron.

KRON MULT-K 120

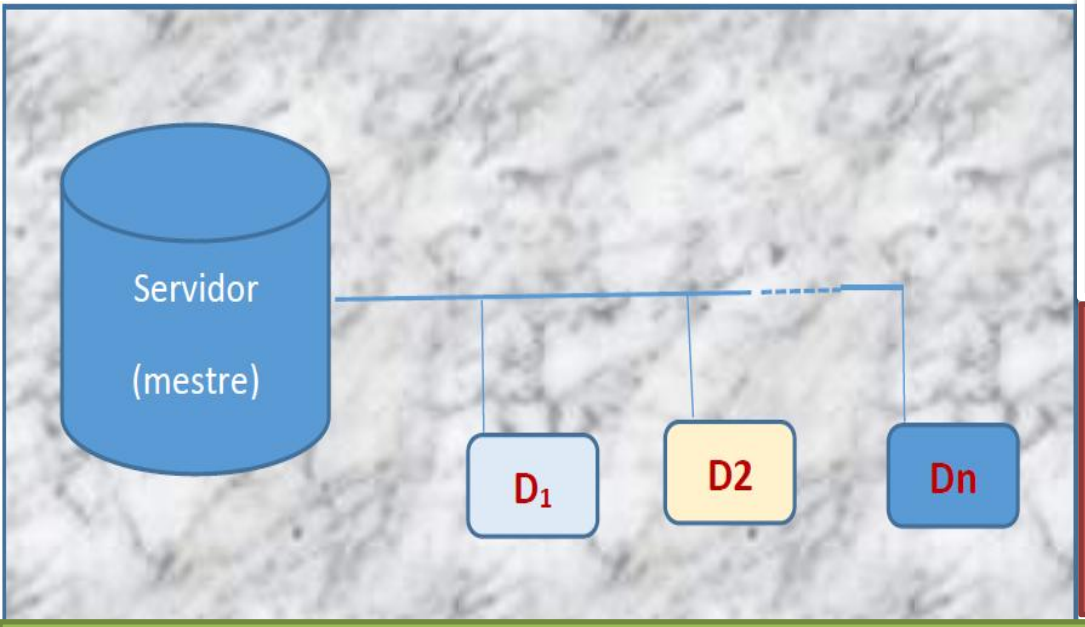
*CONEXÃO COM RS-485
*PROTOCOLO MODBUS
*LEITURA DE ATÉ 44 GRANDEZAS ELÉTRICAS

KRON KONECT

*CONEXÕES: RS-485 e ETHERNET
*PROTOCOLO: MODBUS-RTU; MODBUS TCP e MQTT
*WIRELESS TIPO BLUETOOTH
*LEITURA DE MAIS DE 50 GRANDEZAS ELÉTRICAS

Instalação dos instrumentos KRON no Laboratório de Usinagem

REDE FIELDBUS



Fieldbus é um conceito de rede para transmissão de dados gerados por dispositivos instalados em campo (field) através de um barramento (bus).

O *Fieldbus* pode ser implementado em diferentes arranjos (topologias), sendo que o arranjo mais simples assemelha-se a um varal sobre o qual são pendurados os dispositivos

Além do servidor e dos dispositivos, a rede *Fieldbus* ainda é composta pelo meio físico para transmissão e de um protocolo para formatação dos dados.

O meio físico mais utilizado é determinado pela RS-485 (ANSI TIA/EIA-485) que estabelece um esquema balanceado para transmissão de dados em longa distância dentro de ambientes ruidosos (ruídos gerados por campos eletromagnéticos).

PROTOCOLOS PARA REDES FIELDBUS

Existem vários protocolos no mercado e que atendem às especificações da norma IEC-61158

- AS-Interface
- CAN
- DeviceNet
- Foundation Fieldbus
- HART Protocol
- Industrial Ethernet
- Interbus
- LonWorks
- •Profibus
- •SERCOS
- Profinet
- Modbus

Cada protocolo deve atender plenamente aos requisitos de uma "*Communication Profile Families – CPF*", tal como definido pela norma IEC-61158

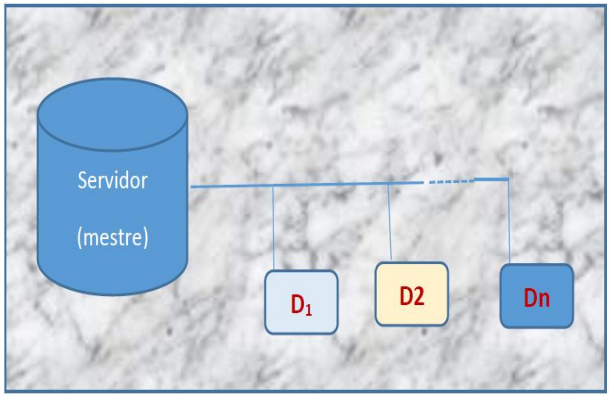
PROTOCOLO MODBUS

O Protocolo Modbus-RTU possui uma comunicação baseada no conceito “mestre-escravos”. Geralmente, a figura do mestre é criada por meio de um programa compilado para um microcomputador (PC) ou microprocessador (CLP). Os escravos são os dispositivos de campo, cujos processadores também possuem um programa capaz de interpretar, processar e responder às mensagens geradas pelo mestre.

A definição completa do protocolo Modbus é aberta ao público e está disponível em: http://www.modbus.org/

PROTOCOLO MODBUS

Nesse tipo de conceito “mestre-escravos”, apenas o mestre pode iniciar uma transação, também denominada por pergunta. Após formatar a pergunta, o mestre irradia-a pela rede. Essa pergunta pode ser despachada para todos os dispositivos (endereço 0) ou para um dispositivo específico (endereço informado na mensagem).



Todos os dispositivos (escravos) irão receber e ler as mensagens (perguntas) enviadas. No caso de mensagens irradiadas para todos os dispositivos (endereço 0), não haverá respostas. Esse tipo de mensagem serve apenas para escrita (alteração de parâmetros).

PROTOCOLO MODBUS

No caso de mensagens endereçadas com o numero de um dispositivo, apenas o destinatário (dispositivo cujo endereço esteja na mensagem) irá processá-la e dar ao mestre uma resposta à pergunta enviada.

O cadastro dos dispositivos na rede fica a cargo do usuário final. Ao todo, poderão ser incluídos um total de 247 dispositivos, os quais devem ser endereçados entre 1 a 247.

With Parity Checking

Start	1	2	3	4	5	6	7	8	Par	Stop

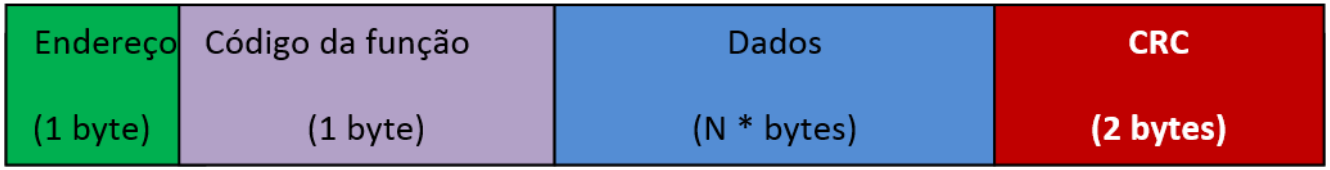
No Modbus-RTU (Remote Terminal Unit) as mensagens são transmitidas sequencialmente (bit-a-bit) e podem variar um pouco em relação ao controle de paridade.

Without Parity Checking

Start	1	2	3	4	5	6	7	8	Stop	Stop

PROTOCOLO MODBUS

No caso do Modbus RTU, uma mensagem é organizada em um quadro de quatro campos



O endereço deve ser um número escolhido no intervalo “0-247”, sendo que o “0” é utilizado para uma comunicação com todos os dispositivos (broadcasting).

O código da função deve ser fornecido pelo desenvolvedor do dispositivo e geralmente é informado no seu manual de instruções.

O campo “Dados” depende da origem da mensagem (mestre ou escravo) e da função utilizada. Isto também deve ser informado pelo desenvolvedor do dispositivo.

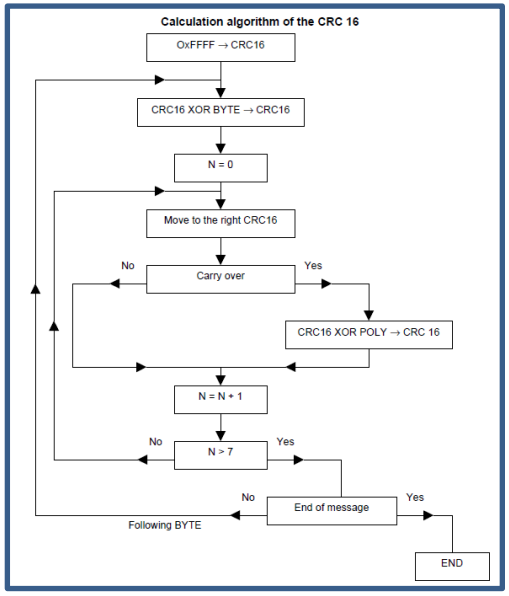
PROTOCOLO MODBUS

Após a definição dos 3 primeiros campos, tanto o mestre quanto o escravo devem acrescentar um quarto campo denominado "CRC".

Esse campo tem um tamanho de 2 bytes (16 bits) e DEVE SER CALCULADO em função do conteúdo dos três primeiros campos.

O CRC é uma forma de conferir a integridade da mensagem transmitida. Dessa maneira, o recebedor da mensagem (mestre ou escravo) deve calcular o CRC e verificar se o valor calculado confere com o valor armazenado no quarto campo.

CRC significa *Cyclical Redundancy Checking* e o seu cálculo deve ser feito por um procedimento (algoritmo) definido no protocolo Modbus. Para maiores detalhes vide: http://modbus.org/docs/Modbus_over_serial_line_V1_02.pdf



PROTOCOLO MQTT

MQTT é um acrônimo para Message Queue Telemetry Transport e refere-se a um protocolo para transferência de dados por Internet (sobre os protocolos TCP/IP). MQTT é um protocolo muito leve e simples, quando comparado, por exemplo, ao protocolo Modbus discutido anteriormente.

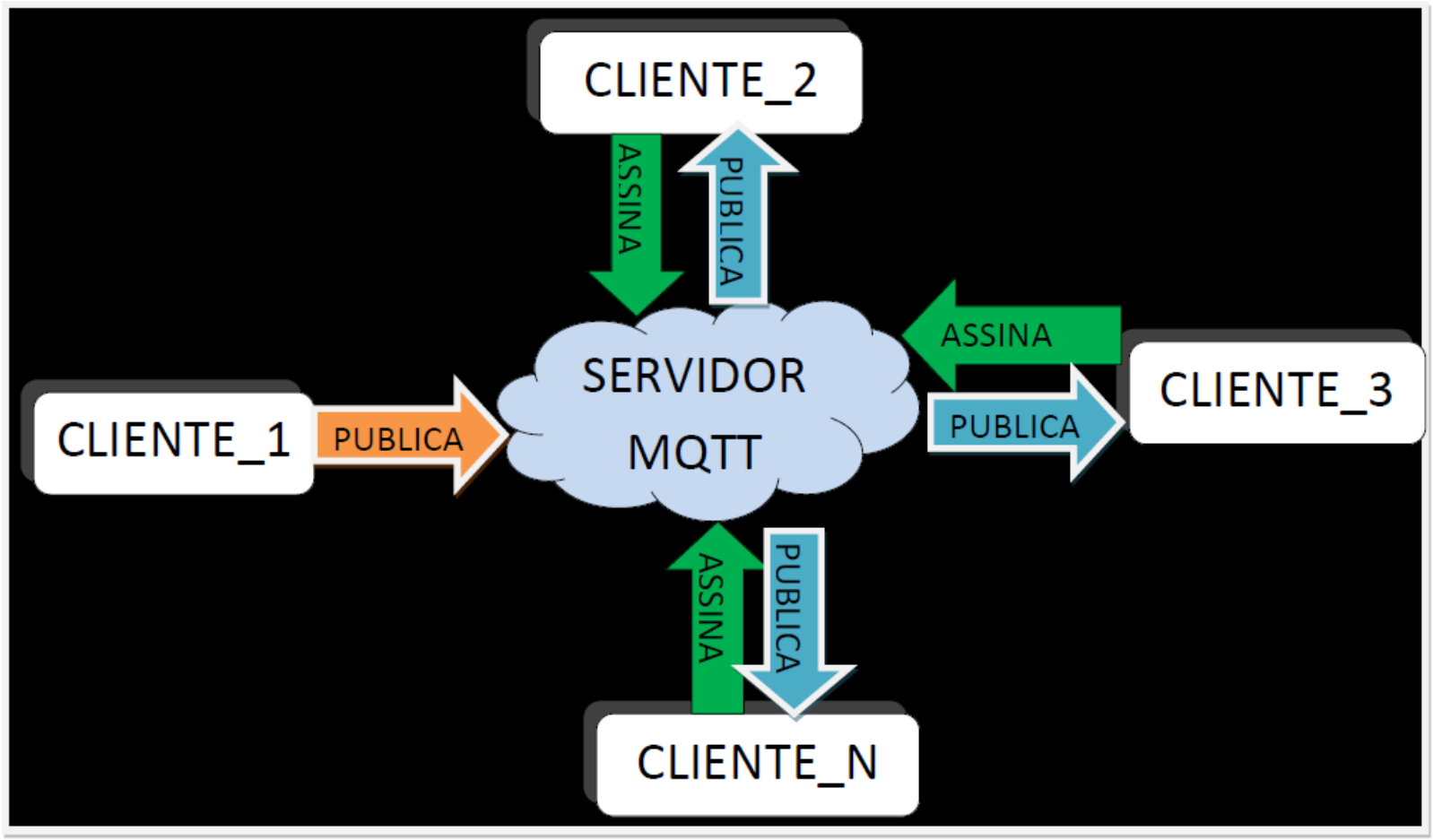
Esse protocolo foi concebido em 1999 por Andy Stanford-Clark (IBM) e Arlen Nipper (Cirrus Link) para aplicações de telemetria, nesse caso, para monitoramento de dutos transportadores de petróleo no deserto (fonte: Wikipedia https://en.wikipedia.org/wiki/MQTT).

PROTOCOLO MQTT

Pelo protocolo MQTT as mensagens são transmitidas na forma de textos “strings” seguindo o formato UTF-8 (Unicode Transformation Format) . O tamanho das mensagens pode variar de poucos bytes até um total de 256 Mb.

O protocolo MQTT é baseado no padrão “publicador/assinante” e possibilita uma distribuição de mensagens do tipo “**um-para-muitos**”, mas sem o acoplamento direto dos clientes, tal como esquematizado a seguir.

PROTOCOLO MQTT



O cliente 1 exerce o papel de “publicador” e os demais (clientes 2 a N) são os “assinantes”. O servidor exerce um papel de mediador (broker).

PROTOCOLO MQTT

O estabelecimento da comunicação entre o cliente publicador e o servidor e do servidor para os clientes assinantes é feito por meio de mensagens “empacotadas” (*packets*). Para isso, o protocolo MQTT define 15 tipos de packets de controle.

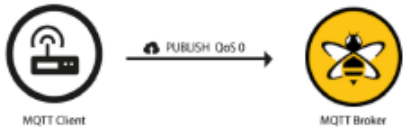
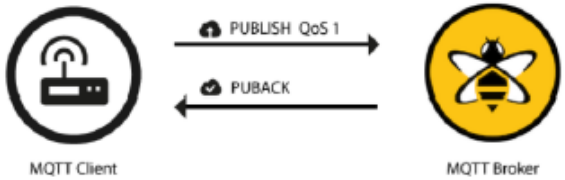
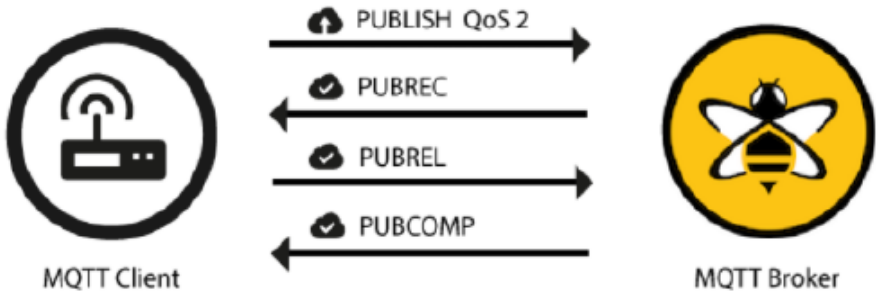
Nome	Valor	Direção do fluxo	Descrição
Reserved	0	Proibido	Reservado
CONNECT	1	Cliente-servidor	Solicitação de conexão
CONNACK	2	Servidor-cliente	Aceite de conexão
PUBLISH	3	Cliente-servidor ou Servidor-cliente	Publica mensagem
PUBACK	4	Cliente-servidor ou Servidor-cliente	Confirmação de publicação (QoS 1)
PUBRECK	5	Cliente-servidor or Servidor-cliente	Publicação recebida (QoS 2 part 1)
PUBREL	6	Cliente-servidor ou Servidor-cliente	Liberação de Publicação (QoS 2 part 2)
PUBCOMP	7	Cliente-servidor ou Servidor-cliente	Publicação realizada (QoS 2 part 3)
SUBSCRIBE	8	Cliente-servidor	Solicitação de assinatura
SUBACK	9	Servidor-cliente	Confirmação de assinatura
UNSUBSCRIBE	10	Cliente-servidor	Solicitação de suspensão de assinatura
UNSUBACK	11	Servidor-cliente	suspensão de assinatura confirmada
PINGREQ	12	Cliente-servidor	Solicitação de teste de conexão (PING)
PINGRESP	13	Servidor-cliente	Resposta do “PING”
DISCONNECT	14	Cliente-servidor ou Servidor-cliente	Notificação de desconexão
AUTH	15	Cliente-servidor ou Servidor-cliente	Troca de autenticação

Fonte: : https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.pdf

PROTOCOLO MQTT

Ao estabelecer uma comunicação, o cliente publicador pode definir um valor (0 a 2) para a qualidade do serviço (QoS). Da mesma forma, um cliente assinante pode definir uma QoS . A qualidade de serviço de uma publicação encaminhada para um assinante pode ser diferente da qualidade de serviço da publicação. O menor dos dois valores é usado para encaminhar uma publicação. Para maiores detalhes vide texto sobre o assunto em https://www.ibm.com/support/knowledgecenter/pt-br/SSFKSJ_8.0.0/com.ibm.mq.dev.doc/q029090_.htm.

PROTOCOLO MQTT

QUALIDADE DO SERVIÇO (QOS)	SIGNIFICADO	COMENTÁRIOS **Fonte** https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels/
0	No máximo uma vez ***(at most once)***	 QoS=0 é o modo de transferência mais rápido.
1	Pelo menos uma vez (***at least once***)	 **QoS=1** é o modo de transferência padrão.
2	Exatamente uma vez (***exactly once– QoS = 2***)	 **QoS=2** é o mais seguro e o modo de transferência mais lento.

PROTOCOLO MQTT

Na utilização de transmissões pelo protocolo MQTT, o servidor (broker) desempenha um papel fundamental. Os clientes em ambas extremidades do fluxo devem custear os serviço prestado pelo broker. Atualmente, várias empresas se especializaram nesse nicho de mercado.

Uma boa notícia, para os desenvolvedores acadêmicos que estão começando um projeto e ainda não dispõem de recursos financeiros, é a utilização de serviços gratuitos. Eles são serviços de nuvem que, embora limitados quanto ao tamanho das mensagens e ao número de clientes, possibilitam a validação de ideias e a utilização plena dos recursos MQTT.

PROTOCOLO MQTT

Além disso, vários desenvolvedores de aplicativos tem disponibilizado suas implementações (códigos) para que usuários iniciantes possam desenvolver suas ideias sem a necessidade de despender muito tempo construindo a interface de comunicação cliente-servidor.

Dois bons exemplos disso são o servidor **Eclipse Mosquitto™**, que provê um serviço gratuito para teste de projetos por meio do broker test.mosquitto.org.; e o cliente MQTT denominado **M2Mqtt** e disponibilizado por Paulo Patierno por meio do NuGet (serviço de compartilhamento de código do .NET framework da Microsof).

PROTOCOLO MQTT

Informações sobre o projeto M2Mqtt podem ser encontradas em https://github.com/eclipse/paho.mqtt.m2mqtt/blob/master/M2Mqtt/MqttClient.cs. Sobre o NuGet podem ser obtidos mais detalhes em: https://docs.microsoft.com/pt-br/nuget/what-is-nuget

O projeto **M2Mqtt** é um definido como sendo um *pacote* (notação UML) contendo várias DLL´s que executam e controlam os pacotes de mensagens do MQTT. Dessa forma, um usuário do sistema Windows poderá desenvolver aplicativos com baixíssimo custo de implementação para a troca de informações entre clientes através do servidor **Mosquitto™**.