RODRIGO MARTINS ROMEIRA SAKAI

**Rede Serial para Comunicação de Dados e Controle em Sistema Embarcado:
Estudo de Implementação da ISO 11783.**

Dissertação apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo para obtenção do título de Mestre em Engenharia Mecânica.

Área de Concentração: Manufatura
Orientador: Ricardo Yassushi Inamasu

São Carlos
2008

Aos meus queridos pais, Mitio e Maria, pelo amor e pelo apoio incondicional.

AGRADECIMENTOS

Ao professor Ricardo Yassushi Inamasu pela amizade e pela orientação no desenvolvimento deste trabalho.

Ao professor Arthur José Vieira Porto pela amizade e pela contribuição que o Laboratório de Simulação e Controle do Departamento de Engenharia Mecânica da EESC/USP proporcionou para esta pesquisa.

Ao Rafael Viera de Sousa, pelo companheirismo e pela disposição demonstrada em diversos momentos deste estudo.

À todos os colegas, professores e funcionários do Laboratório de Simulação e Controle, pela amizade e pela cooperação.

À minha família e a minha namorada, pelo apoio e pelo incentivo em todos os momentos.

À todas as empresas que auxiliaram direta e indiretamente na montagem e testes dos protótipos desenvolvidos neste trabalho, com máquinas, equipamentos, conectores e local para testes: AGCO, ATB BALDAN, ENALTA, ORIGINAL, POWELL e VALTRA.

Ao grupo Força Tarefa ISOBUS, por apoiar e promover o desenvolvimento da norma ISOBUS no Brasil.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq, pela bolsa de estudo concedida.

À Financiadora de Estudos e Projetos – FINEP, pelo projeto “Distribuidor de Insumo Localizado” (FINEP nº 3498/04, Edital/Contrato MCT-RBT/FINEP/EMBRAPA CT-AGRO 01/2004).

RESUMO

SAKAI, R. M. R. **Rede Serial para Comunicação de Dados e Controle em Sistema Embarcado: Estudo de Implementação da ISO 11783**. 2008. 119f. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2008.

As redes digitais demonstraram ser uma solução eficaz em automação. A conexão de diferentes módulos de diferentes fabricantes em um único barramento para a troca de dados e controle é um desafio para a indústria brasileira de máquinas agrícolas, apesar desta tecnologia estar consolidada em automóveis, aeronaves e em chão de fábrica. As vantagens obtidas com redes digitais são evidentes, porém necessitam de implementação de protocolos de redes. Na área agrícola, a norma internacional ISO 11783 apresenta forte potencial para tornar-se referência de padrão para a troca de dados entre módulos em tratores e implementos agrícolas. Esta norma, também conhecida como ISOBUS, está no estágio avançado de desenvolvimento. Contém quatorze documentos e o seu desenvolvimento está apoiado por grupos denominados “Força Tarefa”, na Europa, nos EUA e recentemente no Brasil. Implementações deste padrão já estão sendo apresentadas no mercado internacional, em feiras e demonstrações de aplicação desta tecnologia. O Brasil deve investir e dominar a tecnologia, em busca de compatibilidade internacional tanto no ponto de vista tecnológico como comercial. Neste contexto, este trabalho encoraja o desenvolvimento nacional em aplicações com a norma ISOBUS, apresentando dois estudos de casos, cujos módulos comunicam com equipamentos de mercado compatíveis com a norma. Estas experiências práticas complementam trabalhos acadêmicos relativos a este tema, que surgiram nos últimos anos no Brasil.

Palavras-chave: Automação de máquina agrícola, eletrônica embarcada, ISO 11783, ISOBUS, protocolo de comunicação, fieldbus, CAN, SAE J1939, LBS, terminal virtual.

ABSTRACT

SAKAI, R. M. R. **Serial Control and Communication Data Network on Embedded Systems: Study of Implementation of ISO 11783**. 2008. 119f. Dissertation (Master's degree) – School Engineering of São Carlos, University of São Paulo, São Carlos, 2008.

The digital networks demonstrated to be an effective solution in automation. The connection of different modules from different manufacturers into a single bus for the exchange of data and control is a challenge for the agricultural machinery Brazilian industry, although this technology is consolidated in automobiles, aircraft and the factory floor. The benefits obtained with digital networks are obvious, but they need implementation of protocols networks. In the agricultural area, the international standard ISO 11783 shows strong potential to become the reference standard for the exchange of data between modules on tractors and agricultural implements. This standard, also known as ISOBUS, is in the advanced stage of development. It contains fourteen documents and its development is supported by groups called "Task Force" in Europe, USA, and recently in Brazil. Implementations of this standard are now being presented in the international market, in fairs and demonstrations of application of this technology. Brazil should invest and dominate the technology, to inquire after international compatibility in both the technological point of view as commercial. In this context, this work encourages the national development in applications with ISOBUS standard, presenting two studies of cases, whose modules communicate with equipment market compatible with the standard. These practical experiences complement academic works on this subject, which emerged in recent years in Brazil.

Keywords: Automation in agricultural machinery, embedded electronic, ISO 11783, ISOBUS, communication protocol, fieldbus, CAN, SAE J1939, LBS, virtual terminal.

LISTA DE SIGLAS

ACK	Acknowledgment
ASCII	American Standard Code for Information Interchange
BAM	Broadcast Announce Message
CAN	Controller Area Network
CM	Connection Management
CRC	Cyclic Redundancy Check
CTS	Clear to Send
DA	Destination Address
DIN	Deutsches Institut für Normung
DLC	Data Length Code
DP	Data Page
DT	Data Transfer
ECU	Electronic Control Unit
EOF	End of Frame
ETP	Extended Transport Protocol
EUA	Estados Unidos da América
FMIS	Farm Management Information System
FTI	Força Tarefa ISOBUS
GPS	Global Positioning System
I2C	Inter-Integrated Circuit
IBBC	Implement Bus Breakaway Connector
IBC	Implement Breakaway Connector
IDE	Identifier Extension
IGI	ISOBUS Group Implementation
IHM	Interface Homem-Máquina
ISO	International Organization for Standardization
LBS	Landwirtschaftliches BUS System
LIN	Local Interconnect Network
MGP	Módulo de Guiagem e Propulsão
MOST	Media Oriented System Transport
NAIITF	North American ISOBUS Implementation Task Force
NIU	Network Interconnection Unit

OP	Object Pool
OSI	Open System Interconnection
PCMCIA	Personal Computer Memory Card International Association
PDA	Personal Digital Assistants
PDU	Protocol Data Unit
PDUF	Protocol Data Unit Format
PDUS	Protocol Data Unit Specific
PGN	Parameter Group Number
PTO	Power Take-Off
PWM	Pulse Width Modulation
RAM	Robô Agrícola Móvel
RTS	Request to Send
SA	Source Address
SAE	Society of Automobile Engineers
SD	Secure Digital
SOF	Start of Frame
SPI	Serial Peripheral Interface
SRR	Substitute Remote Request
TBC	Terminating Bias Circuit
TBC_GND	Terminating Bias Circuit Ground
TBC_PWR	Terminating Bias Circuit Power
TC	Task Controller
TECU	Tractor ECU
TP	Transport Protocol
USART	Universal Synchronous Asynchronous Receiver Transmitter
VT	Virtual Terminal
WS	Working Set
WSM	Working Set Master
XML	Extensible Markup Language

LISTA DE TABELAS

TABELA 1 – PARTES DA NORMA ISOBUS, EM 1991. (STONE, 1999)	29
TABELA 2 – ESTADO ATUAL DAS PARTES DA NORMA ISOBUS. (ISO, 2007)	30
TABELA 3 – MENSAGEM <i>GROUND-BASED SPEED AND DISTANCE</i>	37
TABELA 4 – FORMATO PDU 1 E FORMATO PDU 2 (ADAPTADO DE ISO 11783, 1998)	38
TABELA 5 – MENSAGENS DE GERENCIAMENTO DE REDE (ADAPTADO DE ISO 11783, 2001).....	42
TABELA 6 – PGN PARA IDENTIFICAÇÃO DO <i>WORKING SET MASTER</i> E <i>WORKING SET MEMBER</i>	56
TABELA 7 – PEDIDO “EDUCADO” PARA CONFIGURAÇÃO DE ENDEREÇOS.	73
TABELA 8 – TROCA DE MENSAGENS ENTRE WSM E VT, NO PROCEDIMENTO DE CONFIGURAÇÃO DO WS.....	75
TABELA 9 – COMPARAÇÃO DO TAMANHO (<i>BYTES</i>) DE FIGURAS COM DIFERENTES TAMANHOS E RESOLUÇÕES.	80
TABELA 10 – CARACTERÍSTICAS DOS PROTOCOLOS DE TRANSPORTE.	81
TABELA 11 – FUNÇÕES DO PGN TP.CM (ISO 11783, 1998).....	82
TABELA 12 – TROCA DE MENSAGENS ENTRE WSM E VT, NO PROCEDIMENTO DE ENVIO DO OP.	84
TABELA 13 – FUNÇÕES DESENVOLVIDAS PARA WSM.....	89

LISTA DE ILUSTRAÇÕES

FIGURA 1 – CABINE DE UM TRATOR COM DIVERSOS MÓDULOS ELETRÔNICOS (NISSEN, 2007A).	13
FIGURA 2 – CAMADAS DO MODELO DE REFERÊNCIA OSI E DO PROTOCOLO CAN (ADAPTADO DE SOUSA, 2002).	27
FIGURA 3 – QUADRO DE UMA MENSAGEM CAN (2.0 B), COM IDENTIFICADOR DE 29 <i>BITS</i> (ADAPTADO DE ISO 11783, 1998).	27
FIGURA 4 – CONECTORES (A) IBBC, (B) IBC E (C) TBC (POWELL, 2007).	32
FIGURA 5 – TOPOLOGIA DE REDE EM UM TRATOR.	33
FIGURA 6 – TOPOLOGIA DE REDE EM UM IMPLEMENTO AGRÍCOLA. (LEGENDA NA FIGURA 5).	33
FIGURA 7 – TOPOLOGIA DE REDE EM UM TRATOR ENGATADO COM UM IMPLEMENTO AGRÍCOLA. (LEGENDA NA FIGURA 5)	34
FIGURA 8 – OS SEIS CAMPOS CONTIDOS NO CAMPO IDENTIFICADOR.	35
FIGURA 9 – TERMINAL VIRTUAL (ADAPTADO DE ISO 11783, 2004).	45
FIGURA 10 – TOPOLOGIA DE REDE EM UM TRATOR EM DETALHES.	47
FIGURA 11 – ELEMENTO CONTIDO NO DICIONÁRIO DE DADOS (ISO 11783, 2007B).	51
FIGURA 12 – PÁGINA DE REGISTRO NA WEB, PARA ACESSO AO DICIONÁRIO DE DADOS. (ISOBUS, 2007)	51
FIGURA 13 – TOPOLOGIA DE REDE COM SISTEMA DIAGNÓSTICO CONECTADO (ADAPTADO DE ISO 11783, 2007C).	53
FIGURA 14 – CONECTOR PADRÃO DE DIAGNÓSTICO (POWELL, 2007).	54
FIGURA 15 – DIVISÃO VIRTUAL DE ECUs EM UM GRUPO, DENOMINADO <i>WORKING SET</i>	56
FIGURA 16 – CONJUNTO DE <i>BYTES</i> EXEMPLIFICANDO TRÊS OBJETOS EM UM OP.	57
FIGURA 17 – CAMPOS DO OP, VARIÁVEL COM O TIPO DE OBJETO.	58
FIGURA 18 – SEQÜÊNCIA DE PROCEDIMENTOS DE CONFIGURAÇÃO DE UM WS COM UM TERMINAL VIRTUAL. ..	61
FIGURA 19 – METODOLOGIA PARA A IMPLEMENTAÇÃO DO PROCEDIMENTO DE CONFIGURAÇÃO DE UM WS.	62
FIGURA 20 – CIRCUITO ELETRÔNICO COM INTERFACE SERIAL E INTERFACE CAN, DESENVOLVIDA POR SOUSA (2002).	66
FIGURA 21 – CARTÃO PCMCIA PARA INTERFACE DO PROGRAMA CANOE (VECTOR, 2007) COM O BARRAMENTO CAN.	67
FIGURA 22 – TERMINAL GTA (AGCO, 2007).	68
FIGURA 23 – CONJUNTO TRATOR E APLICADOR DE CALCÁRIO (BALDAN, 2007).	68
FIGURA 24 – DISPUTA DE ENDEREÇO: ECU A ENTRA NA REDE E GANHA DISPUTA (ADAPTADO DE ISO 11783, 2001).	71
FIGURA 25 – DISPUTA DE ENDEREÇO: ECU A ENTRA NA REDE E PERDE DISPUTA (ADAPTADO DE ISO 11783, 2001).	72
FIGURA 26 – DIAGRAMA COM SEQÜÊNCIA DE MENSAGENS DE CONFIGURAÇÃO DE UM WS COM UM VT.	74
FIGURA 27 – DIAGRAMA COM SEQÜÊNCIA DE MENSAGENS PARA ENVIO DO OP.	76
FIGURA 28 – TELA PARCIAL DO PROGRAMA <i>MASK GENERATOR</i> (WTK, 2007)	77
FIGURA 29 – TELA PARCIAL DO PROGRAMA DESENVOLVIDO PARA GERAR O ARQUIVO OP.	78

FIGURA 30 – FIGURAS ILUSTRANDO UM IMPLEMENTO AGRÍCOLA. FIGURA NO FORMATO “.BMP” (A) E A MESMA FIGURA INTERPRETADA SEGUNDO A NORMA ISOBUS.	79
FIGURA 31 – QUATRO IMAGENS COM DIFERENTES TAMANHOS E RESOLUÇÕES. EM (A), TAMANHO NORMAL E RESOLUÇÃO DE 256 CORES. EM (B), TAMANHO REDUZIDO E RESOLUÇÃO DE 256 CORES. EM (C), TAMANHO REDUZIDO E RESOLUÇÃO DE 16 CORES. EM (D), TAMANHO REDUZIDO E MONOCROMÁTICO....	80
FIGURA 32 – DIAGRAMA COM SEQUÊNCIA DE MENSAGENS DO PROTOCOLO DE TRANSPORTE (TP).	83
FIGURA 33 – ROBÔ MÓVEL AGRÍCOLA (A) E APLICADOR DE CALCÁRIO (B).....	85
FIGURA 34 – DIAGRAMA ILUSTRANDO A MONTAGEM DO BARRAMENTO PARA O PROTÓTIPO RÁPIDO.	87
FIGURA 35 – MÁSCARA DE DADOS DO OP CRIADO PARA O PROTÓTIPO RÁPIDO.....	88
FIGURA 36 – PROTÓTIPO RÁPIDO MONTADO EM BANCADA NO PRIMEIRO WORKSHOP ISOBUS BRASIL (FTI, 2007). PC COM PROGRAMAÇÃO DO WSM (1), INTERFACE SERIAL-CAN (2), CONTROLADOR DO MGP (3), MGP (4).....	92
FIGURA 37 – DIAGRAMA ILUSTRANDO A MONTAGEM DO BARRAMENTO PARA O SEGUNDO PROTÓTIPO.	94
FIGURA 38 – MÁSCARA DE DADOS DO OP DESENVOLVIDO PARA O CONTROLADOR DE CALCÁRIO. EM (A), TELA INICIAL. EM (B), TELA DE DADOS E CONTROLE.	95
FIGURA 39 – MONTAGEM DO CONTROLADOR DE CALCÁRIO. (1) ECU DO WSM, (2) VÁLVULA COM CONTROLE DE FLUXO EM FUNÇÃO DO SINAL PWM, (3) MOTOR HIDRÁULICO, (4) EIXO PRINCIPAL, (5) SENSOR INDUTIVO PARA LEITURA DA ROTAÇÃO DO EIXO PRINCIPAL, (6) RADAR III.....	96
FIGURA 40 – SENSOR RADAR III (DICKY-JOHN, 2007).	96
FIGURA 41 – DADOS REGISTRADOS NO TESTE DO SEGUNDO PROTÓTIPO.	98

SUMÁRIO

1. INTRODUÇÃO	12
1.1 MOTIVAÇÃO	14
1.2 OBJETIVO	15
1.3 ESTRUTURA	16
2. REVISÃO DE LITERATURA	18
3. PROTOCOLOS	26
3.1 CAN – <i>CONTROLLER AREA NETWORK</i>	26
3.2 A NORMA ISO 11783	28
3.2.1 <i>Padrão Geral</i>	31
3.2.2 <i>Camada Física</i>	31
3.2.3 <i>Camada de Enlace de Dados</i>	35
3.2.4 <i>Camada de Rede</i>	39
3.2.5 <i>Camada de Gerenciamento de Rede</i>	41
3.2.6 <i>Terminal Virtual</i>	43
3.2.7 <i>Mensagens de Implemento</i>	46
3.2.8 <i>Mensagens de Trem de Força (Power Train)</i>	46
3.2.9 <i>ECU do Trator</i>	47
3.2.10 <i>Controlador de Tarefas</i>	49
3.2.11 <i>Dicionário de Dados</i>	50
3.2.12 <i>Sistema de Diagnóstico</i>	52
3.2.13 <i>Servidor de Arquivos</i>	54
3.2.14 <i>Funções Automatizadas</i>	55
3.2.15 <i>Working Set (Conjunto de Trabalho)</i>	55
3.2.16 <i>Object Pool (Conjunto de Objetos)</i>	57
4. MATERIAIS E MÉTODOS	60
5. RESULTADOS E DISCUSSÕES	70
5.1 CONFIGURAÇÃO DE UMA ECU NA REDE	70
5.2 COMUNICAÇÃO VT – WS	74
5.3 CONSTRUINDO UM OP	76
5.4 TRANSMISSÃO DO OP UTILIZANDO O TP	81
5.5 ESTUDOS DE CASO	85
5.5.1 <i>Implementação de um controlador ISOBUS compatível para o controle de um Módulo de Guiagem e Propulsão (MGP)</i>	86
5.5.2 <i>Implementação de um controlador ISOBUS compatível para o controle de um aplicador de calcário</i>	93
6. CONCLUSÃO	100
7. TRABALHOS FUTUROS	104
REFERÊNCIAS	106
APÊNDICES	112

1. INTRODUÇÃO

A crescente utilização de redes digitais para automação e controle contribuiu para o desenvolvimento de novas tecnologias em diversos setores da sociedade.

Estas redes digitais, também conhecidas como *fieldbus* (barramento de campo), são caracterizadas por dispositivos eletrônicos que se comunicam entre si, através de um barramento (conjunto de linhas de comunicação que interligam componentes e periféricos), redes sem fio (*wireless*) (AKYILDIZ, 2004) e também redes híbridas com fio/ sem fio (*wired/ wireless*) (ALVES, 2002) com a finalidade de controle remoto e troca de informações em tempo real.

As principais vantagens obtidas com esta tecnologia são: o uso de cabos com número reduzido de fios, a facilidade de instalação, facilidade de manutenção e flexibilidade de expansão. Contudo, há um aumento na complexidade do sistema, causado pela implementação de protocolos de rede.

A utilização do conceito *fieldbus* se expandiu tanto para automação industrial quanto para eletrônica embarcada, com o desenvolvimento de diversos protocolos como o CAN – *Controller Area Network* (CANBUS, 2007), CANopen (CANOPEN, 2007), SAE J1939 (SAE J1939, 2007), ISO 11783 (ISO 11783, 2006), FlexRay (FLEXRAY, 2007) em veículos e Profibus (PROFIBUS, 2007), Modbus (MODBUS, 2007), DeviceNet (DEVICENET, 2007) na indústria.

O protocolo CAN, criado por Bosch (BOSCH, 1991), foi desenvolvido para promover a interconexão de dispositivos de controle em automóveis e rapidamente se difundiu para outros veículos e também para a indústria, dando origem a outros protocolos citados acima (CANopen, SAE J1939, ISO 11783, DeviceNet). Segundo

Sousa (2002), a robustez, a confiabilidade e a flexibilidade conferida aos padrões desenvolvidos com o protocolo CAN são hoje reconhecidas e essas características têm sido responsáveis pelo crescente número de aplicações deste protocolo.

Na área agrícola é crescente a utilização de equipamentos eletrônicos que buscam um aumento de produtividade no campo, desde a década de noventa (STONE et al., 1999). Inicialmente, a diversidade de fabricantes destes módulos eletrônicos e a falta de um padrão de comunicação geraram um problema de incompatibilidade entre implementos¹ agrícolas. As informações registradas em tratores e implementos agrícolas pelos módulos eletrônicos não puderam ser compartilhadas. Além disso, cada equipamento necessitava de uma interface própria, para visualização de dados e ações de controle, como exemplifica a Figura 1.

Figura 1 – Cabine de um trator com diversos módulos eletrônicos (NISSEN, 2007a).

Neste contexto, um esforço internacional iniciou o desenvolvimento de uma norma, denominada ISO 11783, no início da década de noventa, para atender as

¹ Neste trabalho, o termo “implemento” refere-se a dispositivo ou máquina que realiza uma operação específica e que é normalmente fixada em um trator (ISO 11783, 2007a).

necessidades da comunicação eletrônica em máquinas e implementos agrícolas, sob o suporte da ISO – *International Organization for Standardization* (ISO, 2007). A norma ISO 11783, também chamada norma ISOBUS, ou padrão ISOBUS, faz uso do protocolo CAN. A norma ISO 11783 tem por objetivo especificar um sistema de comunicação aberto para a comunicação e troca de dados entre equipamentos eletrônicos de diferentes fabricantes. A troca de informações de forma padronizada permite aos fabricantes de módulos eletrônicos de tratores e implementos agrícolas a simplificação de projetos de circuitos eletrônicos (*hardware*).

1.1 MOTIVAÇÃO

A norma ISOBUS tem sido desenvolvida ao longo dos últimos dezesseis anos. Atualmente, onze das quatorze partes já foram publicadas. Além disso, dez destas onze partes foram publicadas apenas nos últimos sete anos, sendo três partes no ano de 2007 (ISO, 2007). Verificamos que este padrão está no auge do desenvolvimento, com suporte de grandes grupos internacionais: John Deere, AGCO e CNH (ISOBUS, 2007).

Segundo Fellmeth (2003), somente após dez anos do início da norma, foram apresentados os protótipos iniciais de equipamentos, em novembro de 2001, na *Agritechnica* (2007) *Trade Show*, em Hanover, na Alemanha, e posteriormente em maio de 2002 na *Agricultural Machinery Conference – AMC* (2007) *Show*, em Iowa, nos Estados Unidos (EUA).

O desenvolvimento de equipamentos compatíveis com o padrão, ou ISOBUS compatíveis, se tornou uma tendência global. Os “*plugfest*”, encontros de

desenvolvedores para testar a comunicação entre seus equipamentos, são cada vez mais comuns. Segundo Rudolph (2007), coordenador da força tarefa norte americana que promove o padrão ISOBUS, chamada *North American ISOBUS Implementation Task Force* (NAITF), atualmente existem dois laboratórios no mundo que certificam equipamentos segundo o padrão ISOBUS, DLG na Alemanha e Wyle Labs nos EUA, e um interesse da Itália em criar um terceiro laboratório com esta competência. Esta tendência também é verificada no desenvolvimento de programas (*softwares*), ferramentas e bibliotecas que auxiliam no desenvolvimento de aplicações ISOBUS (SAKAI et al, 2007b).

No Brasil, a utilização de máquinas e implementos agrícolas com o padrão ISOBUS ainda está em fase de desenvolvimento. Na edição de 2007 da Agrishow, uma feira com grande importância no contexto de máquinas e implementos agrícolas, realizada em Ribeirão Preto desde 1994, não houve registro de implementos agrícolas compatível com o padrão.

Neste contexto, surge a oportunidade de estudar as partes já publicadas e realizar a implementação de equipamentos ISOBUS compatíveis.

1.2 OBJETIVO

O objetivo principal deste trabalho é o estudo do processo de implementação do protocolo de comunicação de eletrônica embarcada ISOBUS em uma máquina agrícola. Desta maneira, este trabalho busca exercitar a aplicação do recente padrão, com pretensão de complementar experiências práticas no Brasil, em um ambiente acadêmico. O desenvolvimento progressivo deste trabalho explora em

detalhes partes específicas da norma, indicando os pontos essenciais para determinados procedimentos. Por outro lado, realiza a sistematização de um procedimento macro, que é a configuração de uma máquina agrícola com um terminal do trator, ao vincular diversos procedimentos dispersos nos documentos da norma ISOBUS.

1.3 ESTRUTURA

Este trabalho está dividido em sete capítulos, organizados na sequência dos estudos realizados e da implementação da norma ISOBUS.

O **capítulo 1 – Introdução** – apresenta as justificativas que motivaram este trabalho assim como o objetivo e a organização da estrutura.

No **capítulo 2 – Revisão de Literatura** – são apresentadas as etapas de desenvolvimento da norma ISOBUS e os eventos e trabalhos que contribuíram para a divulgação, sistematização e implementação do padrão ISOBUS.

No **capítulo 3 – Protocolos** – é apresentada a síntese do protocolo CAN e da norma ISOBUS utilizados no trabalho.

No **capítulo 4 – Materiais e Métodos** – é apresentada a metodologia utilizada para o desenvolvimento desta pesquisa e os padrões, *softwares* e *hardwares* utilizados em cada etapa da implementação. Eventos também são considerados neste item, responsáveis por “catalisar” o desenvolvimento deste trabalho.

No **capítulo 5 – Resultados e Discussões** – são apresentados os resultados da implementação da norma ISOBUS em dois estudos de caso, segundo a metodologia adotada.

No **capítulo 6 – Conclusão** – são apresentadas as conclusões obtidas com este trabalho.

No **capítulo 7 – Trabalhos Futuros** – são apresentadas propostas para trabalhos futuros.

2. REVISÃO DE LITERATURA

No início do desenvolvimento da norma ISO 11783, na década de noventa, foram relativamente poucas as publicações técnicas e científicas sobre o padrão ISOBUS como pode ser constatado em trabalhos de revisão sobre padrões de comunicação para eletrônica embarcada em máquinas e implementos agrícolas (STRAUSS, 2001; SOUSA, 2002). No ano 2000, com uma parte publicada e outras partes ainda no estado de produção, a norma ISOBUS já possibilitava estudos e implementações básicas do padrão. Conseqüentemente, foram realizados estudos sobre este padrão relacionados a temas como a utilização de eletrônica embarcada em máquinas agrícolas e agricultura de precisão. Nos últimos anos também se verificam trabalhos sobre aplicações da norma em implementos agrícolas com nível limitado de compatibilidade como apresentado a seguir neste Capítulo.

A norma ISO 11783 teve seu início em 1991, patrocinada pela ISO. Os trabalhos são coordenados pelo Comitê Técnico 23, Sub-Comitê 19, Grupo de Trabalho 1 (TC23/SC19/WG1). Desde então, este grupo se reúne periodicamente para discutir e votar o desenvolvimento das partes da norma. O desenvolvimento de cada parte é realizado em sete estágios diferentes. A cada estágio o projeto é votado, decidindo-se por sua continuação, correção ou abandono. O sétimo estágio é a publicação do documento como padrão internacional. Após este estágio, a norma fica sob revisão periódica, para que modificações necessárias possam ser realizadas (ISO, 2007).

Em conseqüência deste processo de desenvolvimento longo e contínuo, estes estudos ficaram reservados ao comitê da ISO por alguns anos e somente em 1998

foi realizada a primeira publicação de uma parte da norma: a Parte 3 – Camada de Enlace de Dados (*Data Link Network*). Nos anos seguintes novas partes foram publicadas, e atualmente 11 das 14 partes em desenvolvimento já constituem padrões internacionais, sendo as partes mais recentes a 1 (Padrão Geral), a 11 (Dicionário de Dados) e a 13 (Servidor de Arquivos), publicadas em 2007 (ISO, 2007).

Em 1999, o professor Marvin L. Stone, da *Oklahoma State University*, apresentou o protocolo ISO 11783 na *Agricultural Equipment Technology Conference*, em *Louisville*, nos EUA. Stone (1999) afirma a importância da padronização da comunicação eletrônica entre os sistemas de controle distribuídos nos tratores e implementos agrícolas, e aponta um fator crítico a favor da evolução da norma ISO 11783: as necessidades do consumidor, que envolvem aumento de desempenho, compatibilidade entre diferentes fabricantes, possibilidade de expansão (componentes modulares podem ser adicionados) e baixo custo.

Em novembro de 2001, dez anos após o início dos trabalhos da norma, os protótipos iniciais foram apresentados na *Agritechnica Trade Show* em Hanover. Nos EUA, a primeira exibição foi em *Cedar Rapids*, Iowa, em maio de 2002, na *Agricultural Machinery Conference (AMC) Show*. Estas primeiras apresentações buscaram disseminar e promover o padrão ISOBUS aos fabricantes de máquinas e implementos agrícolas, e também aos consumidores finais, demonstrando sua interoperabilidade e compatibilidade do sistema (FELLMETH, 2003). Contudo, ainda não existiam produtos ISOBUS compatíveis, nem tratores e nem implementos agrícolas.

Em 2001, Munack (2001) afirma que os objetivos da agricultura moderna, e.g. a produção suficiente de alimentos de qualidade e matérias-primas, podem ser

alcançados através da utilização de máquinas, equipamentos e processos com alta eficiência, e conclui que o padrão ISO 11783 deve ser expandido para satisfazer as crescentes tarefas e funções da agricultura de precisão.

No Brasil, Strauss (2001), Sousa (2002), Silva e Guimarães (2003) e Landi (2004) realizaram sistematizações e aplicações com o protocolo CAN e o padrão ISOBUS na área agrícola. Sousa (2002) gerou um documento referencial para o uso do CAN para aplicações agrícolas. Sousa (2002) também apresentou um protótipo eletrônico que promove a interconexão de um dispositivo ao barramento, denominado Electronic Control Unit (ECU) ou Unidade Eletrônica de Controle. Este sistema possui um microcontrolador programável, interface serial RS-232, interface com o barramento CAN e pode ser aplicado em projetos que utilizam o protocolo CAN como, por exemplo, o projeto desenvolvido no presente trabalho, que utiliza o padrão ISOBUS.

Guimarães (2003) analisou comparativamente protocolos de comunicação serial utilizados em aplicações agrícolas, e desenvolveu uma aplicação com um monitor de semeadora baseada na norma ISO 11783. Guimarães (2003) explicou as principais características do padrão ISOBUS, segundo cada parte da norma ISOBUS, na época contendo 11 partes. A proposta de arquitetura de implementação apresentada compreende duas ECU que representam o monitor de semeadora em um barramento ISOBUS. Os dados foram monitorados por um computador pessoal, através de um canal de comunicação serial RS-232 em cada ECU.

Landi (2004) estudou uma proposta de implementação de um Terminal Virtual e um Controlador de Tarefas, ECU definidas pela norma ISOBUS (serão detalhadas no capítulo 3 – Protocolos), em um Dispositivo Computacional Portátil, ou *Personal Digital Assistants* (PDA). Landi (2004) descreveu detalhadamente estas ECU,

Terminal Virtual e Controlador de Tarefas, referentes às partes 6 e 10 da norma ISOBUS, respectivamente, contribuindo para a compreensão deste padrão.

Em 2003, Fellmeth (2003), representante da empresa Vector Informatik, promoveu o padrão ISOBUS, afirmando este ser uma tendência causada pelo aumento do uso de eletrônica nos tratores e implementos. Fellmeth (2003) também afirmou que o custo adicional de uma rede de troca de dados representa uma pequena porção do custo total do desenvolvimento de eletrônica e ainda possibilita um significativo aumento no desempenho do sistema como um todo.

Um grupo de pesquisadores do *Automation Technology Laboratory (Helsinki University of Technology – HTU)*, na Finlândia, desenvolveu sistemas para automação em máquinas agrícolas e utiliza o padrão ISOBUS para suas implementações. O projeto Agrix, como é chamado, foi iniciado com um protótipo rápido (*Fast Prototype*) em 2003 (OKSANEN et al, 2004), no qual foi utilizado um trator comercial com uma ECU do Trator ISOBUS compatível, um receptor de GPS (*Global Positioning System*, Sistema de Posicionamento Global) e um implemento com plantadeira e fertilizador. Toda a eletrônica básica deste implemento foi substituída, com o objetivo de obter um controle de taxa de aplicação variável, segundo as recomendações de um estudo prévio, ou seja, um mapa de aplicação. Neste primeiro protótipo, nem um Terminal Virtual nem um Controlador de Tarefas ISOBUS compatível foi utilizado. Ao invés disso, um PDA substituiu tais funcionalidades, porém suas características limitaram os testes. No segundo protótipo (OKSANEN et al, 2005), chamado *Basic Prototype*, foram utilizados dois Terminais Virtuais comerciais, porém o Controlador de Tarefas continuou sendo simulado com um PDA, com algumas diferenças de configuração para com a norma. Mensagens do padrão foram implementadas, porém a comunicação possuía um

barramento para cada componente do sistema. Estes dois protótipos demonstraram a dificuldade da implementação da norma, na época com poucas partes publicadas, e concluíram que a norma ISO 11783 será muito importante como um padrão de comunicação para máquinas e implementos agrícolas. OKSANEN et al (2005) afirmou que a parte mais complexa a ser implementada é uma ECU de implemento, pois é responsável por coordenar tarefas com outras ECU como o Terminal Virtual e o Controlador de Tarefas.

Em 2004, Auernhammer e Rothmund (2004) estudaram modelos de gerenciamento de informações automatizados em uma rede ISOBUS. O gerenciamento inclui aquisição, armazenamento, transferência e processamento de dados. Auernhammer e Rothmund (2004) concluíram que “a aquisição de dados automática dentro de uma arquitetura em um sistema de comunicação padronizado para máquinas agrícolas será um modo sensato de obter a base de dados para um guia de informações de produção no futuro”.

Em 2005, Benneweis (2005), publicou o estado da norma ISOBUS, com uma visão resumida de cada parte do documento, cujo número de partes havia aumentado para 13. Benneweis (2005) apresenta as implementações até então realizadas pelas organizações NAIITF, nos EUA, e *Implementation Group* ISOBUS (IGI), na Europa, e concluiu reafirmando o potencial gerado pelo padrão, em relação a futuras implementações.

Em 2006, Saraiva e Cugnasca (2006) revisaram as principais normas internacionais propostas para redes de comunicação serial em máquinas agrícolas, tais como CAN, LBS e ISOBUS. Saraiva e Cugnasca (2006) também ressaltaram a iniciativa brasileira no tema ISOBUS, que levou a criação do grupo denominado “Força Tarefa ISOBUS Brasil” (FTI) (FTI, 2007).

Em 2007, Sakai et al (2007a) apresentou uma sistematização do padrão ISOBUS considerando a troca de informações entre um implemento agrícola com um Terminal Virtual e um Controlador de Tarefas.

Além dos trabalhos que surgiram em função do avanço deste padrão, *softwares* e bibliotecas foram desenvolvidos para auxiliar a implementação de equipamentos ISOBUS compatíveis (SAKAI et al, 2007b). Dentre as ferramentas comerciais, a mais completa é o *software* proprietário Canoe.ISO11783 (VECTOR, 2007). Já a *Mask Generator* (WTK, 2007) se destaca pela aplicação específica no desenvolvimento da interface gráfica para aplicações com o Terminal Virtual. A Microchip (2007) disponibilizou uma biblioteca contendo funções básicas para a implementação de controladores CAN, baseada na norma SAE J1939. Outra biblioteca interessante é o ISOAgLib (2007). Este *software* livre disponível implementa diversas funcionalidades descritas na norma ISOBUS. Segundo Schmidt (2007), o ISOAgLib (2007) oferece diversos tutoriais e exemplos para quase todas as partes da norma ISOBUS.

Outra atividade relevante para o desenvolvimento da norma são os eventos denominados *plugfest*. São reuniões técnicas nas quais são testados os equipamentos ISOBUS compatíveis de diferentes fabricantes. Grande parte dos participantes desses eventos são profissionais em tecnologia e também membros do comitê de desenvolvimento da norma ISO 11783. O objetivo deste evento não é a certificação de um equipamento ISOBUS, e sim o teste dos parâmetros do protocolo, dos protocolos de troca de mensagens, dos protocolos de transporte, da operabilidade de um Terminal Virtual, de um Controlador de Tarefas, de um Servidor de Arquivos, entre outros. Os resultados obtidos nestas reuniões servem de

avaliação prática para o desenvolvimento da norma. Alguns problemas são verificados somente nos *plugfest*.

Segundo Sam Freesmeyer, representante da AGCO (2007), e ex-coordenador da NAIITF, o primeiro *plugfests* nos EUA contou com 27 membros de 16 diferentes empresas, entre essas grandes empresas multinacionais tais como AGCO, CNH, John Deere, Trimble e Vector Informatik. Durante os testes há relatos de que foram encontrados diversos problemas de interoperabilidade entre os equipamentos, cujas causas foram identificadas. Assim, os desenvolvedores puderam entender melhor a aplicação da norma, corrigindo os erros encontrados (PLUGFEST, 2007).

O nível técnico dos *plugfest* cresceu gradualmente. Nos primeiros eventos foram testados protótipos de Terminais Virtuais, Controladores de Tarefas, ECU do Trator e ECU de implementos, todos em bancadas de teste. Atualmente, os testes são realizados em campo com tratores e implementos realizando uma tarefa, e o foco dos testes são, por exemplo, comunicações através de portais (*gateways*) (NISSEN, 2007b).

Tão importante quanto a realização de *plugfest* são os laboratórios especializados para a certificação de ECU quanto ao padrão ISOBUS. Esta necessidade natural motivou dois laboratórios a certificarem o padrão ISOBUS: DLG na Europa e o Wyle Labs nos EUA. Após a certificação na DLG, os produtos e fabricantes são registrados para conhecimento (ISOBUS, 2007).

A norma ISOBUS é disseminada por um esforço mundial, no qual estão a frente grupos de indústrias, universidades e entidades, sem fins lucrativos, tais como a NAIITF (2007) nos EUA, a IGI (2007) na Europa e a FTI no Brasil, que promovem,

recomendam e encorajam o desenvolvimento de equipamentos compatíveis com a norma ISOBUS, além de organizarem os eventos tais como reuniões, oficinas, *plugfest* e apresentações em feiras e congressos.

3. PROTOCOLOS

3.1 CAN – *Controller Area Network*

O protocolo de comunicação serial digital CAN foi desenvolvido na década de oitenta por Robert Bosch Gmb (BOSCH, 1991), para promover a interconexão entre dispositivos de controle em automóveis. A robustez do protocolo CAN fez com que esta tecnologia migrasse para outras áreas.

As características principais deste protocolo são:

- Rede multimestre: o barramento pode ser utilizado por qualquer unidade quando o mesmo estiver livre;
- Taxa de comunicação variada, até 1 Mbps;
- Quadro de dados reduzido, de no máximo 8 *bytes*, para troca de informações concisa;
- Arbitragem para acesso ao meio sem colisão;
- Campos de *bits* na mensagem para identificação de erros;
- Possibilidade de adição, remoção e mudança de dispositivos, permitindo diferentes configurações do sistema;

O protocolo CAN possui camada física e de enlace de dados, de acordo com as sete camadas do modelo padrão *Open System Interconnection* (OSI) (TANENBAUM, 1997). As demais camadas deste modelo estão abertas para implementação de alto nível, segundo a necessidade de cada aplicação. Assim, diversos protocolos são baseados no protocolo CAN, e definem aplicações apenas

nas camadas superiores. A Figura 2 ilustra as camadas do modelo OSI e as camadas utilizadas pelo protocolo CAN.

Figura 2 – Camadas do modelo de referência OSI e do protocolo CAN (Adaptado de SOUSA, 2002).

A comunicação de dados em uma rede com o protocolo CAN é baseada em mensagens com formato fixo. As mensagens são formadas por vários campos de *bits* ou conjuntos de *bits* que possuem determinada função. A Figura 3 ilustra a mensagem CAN (formato CAN 2.0 B) e os campos dos quais ela é formada.

Figura 3 – Quadro de uma mensagem CAN (2.0 B), com identificador de 29 *bits* (Adaptado de ISO 11783, 1998).

De acordo com a especificação do protocolo CAN (BOSCH, 1991), existem duas versões que diferem quanto ao tamanho do campo do identificador. A versão

CAN 2.0 A determina um identificador com 11 *bits*, enquanto a versão estendida CAN 2.0 B determina um identificador com 29 *bits*.

Trabalhos anteriores (STRAUSS, 2001; SOUSA, 2002; GUIMARÃES, 2003) fornecem maior detalhamento deste protocolo já consolidado.

3.2 A norma ISO 11783

A norma ISO 11783 estabelece parâmetros para a comunicação entre ECU em máquinas e implementos agrícolas. Esta norma com catorze partes, sendo onze publicadas pela ISO, teve seu início em 1991 por um comitê da ISO a partir da união de duas outras normas, a DIN 9684 da Associação de Normas Alemã *Deutsches Institut für Normung* (DIN) e a SAE J1939 da Sociedade Norte Americana – *Society of Automotive Engineers* (SAE). A norma DIN 9684 – *Agricultural Tractors and Machinery*, ou norma LBS (*Landwirtschaftliches BUS System* – Barramento Agrícola Móvel) (LANDTECHNIK-VEREINIGUNG, 2002) foi desenvolvida na Alemanha por grupos de empresas e instituições associados à DIN. A primeira versão desta norma foi realizada em 1997, com cinco partes, das quais duas tiveram grande influência na norma ISO 11783. A norma SAE J1939 (2007) – *Recommended Practice for Truck and Bus Control and Communication Network* – foi desenvolvida pelo comitê SAE J1939 *Truck and Bus Control and Communications Subcommittee* para aplicações em veículos pesados, como ônibus, caminhões e veículos de construção civil. A descrição de cada parte da LBS e da SAE J1939 é detalhada por Strauss (2001) e Sousa (2002).

O grupo de trabalho (*Working Group* – WG1) da norma ISO 11783 teve as primeiras discussões em fevereiro de 1991 (STONE, 1999), abordando o desenvolvimento de um conector padrão e adotando o uso do recente padrão CAN 2.0B. A partir deste momento, foram estabelecidas as partes desta norma, e os respectivos documentos fonte, mostrados na Tabela 1.

Tabela 1 – Partes da norma ISOBUS, em 1991. (STONE, 1999)

Parte	Título	Documento Fonte
1	Padrão Geral	WG1
2	Camada Física	WG1 SAE J1939-13 (<i>Diagnostic connector</i>)
3	Camada de Enlace de Dados	SAE J1939-21 (<i>Data link layer</i>)
4	Camada de Rede	SAE J1939-31 (<i>Network layer</i>)
5	Gerenciamento de Rede	SAE J1939-81 (<i>Network management</i>)
6	Terminal virtual	DIN 9684 (<i>Virtual Terminal</i>)
7	Mensagens básicas de implemento	WG1
8	Camada de aplicação/ <i>Drivetrain</i>	SAE J1939-71 (<i>Applications layer</i>)
9	ECU do Trator	WG1
10	Controlador de Tarefas e interface do computador de gerenciamento	DIN 9684 (<i>Management computer interface</i>)

Os documentos fontes na Tabela 1 são partes das normas SAE J1939 e DIN 9684. Partes que não tiveram um documento anterior foram iniciadas pelo próprio grupo de trabalho WG1, como indicado.

Atualmente, a norma ISO 11783 é formada por catorze documentos. O estado de desenvolvimento atual da norma está descrito na Tabela 2.

Tabela 2 – Estado atual das partes da norma ISOBUS. (ISO, 2007)

Parte	Título	Estado
1	Padrão Geral	Padrão Internacional
2	Camada Física	Padrão Internacional
3	Camada de Enlace de Dados	Padrão Internacional
4	Camada de Rede	Padrão Internacional
5	Gerenciamento de Rede	Padrão Internacional
6	Terminal Virtual	Padrão Internacional
7	Camada de Aplicação – Mensagens de Implemento	Padrão Internacional
8	Camada de Aplicação – Mensagens de trem de força	Padrão Internacional
9	ECU do Trator	Padrão Internacional
10	Controlador de Tarefas e Sistema de Gerenciamento de Informações e Troca de Dados	Projeto Final registrado para aprovação formal
11	Dicionário de Dados	Padrão Internacional
12	Sistema de Diagnóstico	Projeto Final registrado para aprovação formal
13	Servidor de Arquivos	Padrão Internacional
14	Funções Automatizadas	Novo Projeto Aceito

Na Tabela 2, as partes concluídas estão com o estado “Padrão Internacional”. Estas partes já foram publicadas e estão disponíveis (ISO, 2007). As partes 10 e 12 estão em “Projeto Final registrado para aprovação formal”, o que significa que será votado e caso seja aceito, será encaminhado para o estágio de publicação. A parte 14 está no estado “Novo Projeto Aceito”. Isto significa que esta parte ainda será preparada e avaliada posteriormente, antes de ser publicada.

Os capítulos a seguir descrevem de forma geral as informações contidas nas partes da norma ISO 11783. Vale a pena lembrar que Sousa (2002), Guimarães (2003), Landi (2004) e Benneweis (2005) já descreveram as partes da norma ISO. Desde então houve avanços, com novas partes publicadas, novas edições e com inclusão de anexos. Assim, apresenta-se uma atualização na revisão do padrão, para contribuir com a difusão de novos conceitos e detalhes.

3.2.1 Padrão Geral

A parte 1 da norma ISOBUS constitui os alicerces deste padrão. Nesta parte são apresentados termos e definições utilizados nas treze partes restantes, abreviações de termos, a aplicação do modelo OSI ao padrão e requisitos gerais de uma rede ISOBUS.

Os anexos desta parte contêm todos os identificadores de mensagens, de endereços preferenciais e de grupos de indústrias, funções de controle e códigos de fabricantes. Além disso, há também formulários para requisição de códigos de fabricantes, novos identificadores ou até modificação dos mesmos.

3.2.2 Camada Física

A parte 2 da norma ISOBUS descreve a camada física. São descritos os parâmetros elétricos, conectores padrão, comportamento mínimo necessário da rede, em casos de perda de conexão ou falhas do barramento CAN, e em anexo, a subdivisão do tempo de *bit* e exemplos de circuitos elétricos.

A camada física é baseada no protocolo CAN 2.0 B e define a taxa de 250 kbps para a comunicação serial. O CAN 2.0 B também define o meio físico, ou o barramento, composto por quatro linhas: dois condutores de dados denominadas *CAN High* (CAN_H) e *CAN Low* (CAN_L), e dois condutores de referência elétrica, denominados *Terminating Bias Circuit Power* (TBC_PWR) e *Terminating Bias Circuit Ground* (TBC_GND).

A conexão padrão entre o trator e o implemento agrícola é realizada pelo conector *Implement Bus Breakaway Connector* (IBBC), que está localizado no trator. A função principal deste conector é agrupar o duto de dados com o duto de potência (energia elétrica). Assim, um implemento agrícola obtém do trator a conexão com o barramento CAN e alimentação 12V através de um conector apenas, o *Implement Breakaway Connector* (IBC), localizado no implemento agrícola. Ambos os conectores são mostrados na Figura 4.

Figura 4 – Conectores (a) IBBC, (b) IBC e (c) TBC (POWELL, 2007).

A parte 2 da norma ISOBUS também define uma terminação que deve estar localizada nos dois extremos de cada barramento, ou seja, tanto no barramento de trator quanto no barramento de implemento. Esta terminação, chamada *Terminating Bias Circuit* (TBC) tem a função de fornecer uma referência de nível elétrico entre os pinos CAN_H e CAN_L e o promover o casamento de impedâncias nos extremos da rede, através da alimentação fornecida pelos pinos TBC_PWR e TBC_GND. A topologia de uma rede ISOBUS em um trator é mostrada na Figura 5.

Figura 5 – Topologia de rede em um trator.

Como vemos na Figura 5, o trator possui dois barramentos: o barramento de trator e o barramento de implemento. Os dois barramentos são separados por uma ECU chamada ECU do Trator, ou *Tractor ECU* (TECU). Ambos possuem a terminação TBC em seus extremos. A topologia do implemento, por sua vez, é mostrada na Figura 6.

Figura 6 – Topologia de rede em um implemento agrícola. (Legenda na Figura 5).

Como vemos na Figura 6, o implemento possui em um dos seus extremos o IBC, visto anteriormente. Este conector padrão permite a comunicação do implemento agrícola com um trator que possua o conector IBBC. No outro extremo, o implemento deve conter um TBC, pois o fim do barramento de implemento se deslocou do trator para o implemento.

Figura 7 – Topologia de rede em um trator engatado com um implemento agrícola. (Legenda na Figura 5)

Como vemos na Figura 7, trator e implemento dividem o barramento de implemento, que possui o TBC corretamente em seus extremos.

Como pode ser observado nas três configurações anteriores, o conector padrão IBBC também tem a função de realizar a função de TBC. Quando não há nenhum implemento conectado ao IBBC, o próprio conector é um dos extremos do barramento, e por isso fornece a terminação TBC. Quando algum implemento se conecta ao trator pelo IBBC, a terminação TBC é desativada automaticamente, uma vez que este local não é mais considerado o extremo do barramento, agora localizado no fim do barramento de implemento agrícola. O implemento agrícola deve fornecer o TBC no extremo do barramento.

3.2.3 Camada de Enlace de Dados

A parte 3 da norma ISOBUS descreve a camada de enlace de dados. São descritos o formato do quadro de mensagens, a unidade de protocolo de dados, ou *Protocol Data Unit* (PDU), os tipos de mensagens, prioridade das mensagens, o mecanismo de acesso ao meio de comunicação, o processo de arbitragem, a detecção de erros, o protocolo de transporte e, nos anexos desta parte em questão, uma rotina para o processamento de mensagens, a seqüência da transferência de dados via protocolo de transporte, ou *Transport Protocol* (TP) e exemplos de modo de comunicação.

A camada de enlace de dados também é baseada no protocolo CAN. O campo identificador é dividido em campos menores, conforme a norma SAE J1939, que adota o formato de quadro estendido, o CAN 2.0 B. Este formato define o identificador de 29 *bits*. O identificador é formado por 6 campos: prioridade (3 *bits*), página de dados (1 *bit*), reservado (1 *bit*), Formato de PDU (8 *bits*), PDU Específico (8 *bits*) e Endereço de origem (8 *bits*), mostrado na Figura 8.

Figura 8 – Os seis campos contidos no campo identificador.

A norma define uma entidade chamada “Número do Parâmetro de Grupo”, ou *Parameter Group Number* (PGN). O PGN é formado pelos campos reservado, página de dados, ou *data page* (DP), Formato de PDU, ou *PDU Format* (PDUF) e PDU Específico, ou *PDU Specific* (PDUS), totalizando 18 *bits*. Cada PGN é associado a uma, e apenas uma mensagem. Assim, as mensagens são identificadas pelos PGN, que se encontram no campo identificador de cada quadro. O campo de dados deve ser interpretado pelas ECU de acordo com o PGN identificado. Ou seja, para cada PGN existe um protocolo que define o conteúdo, a divisão e as unidades do campo de dados. Por exemplo, a mensagem *Ground-Based Speed and Distance*, que é enviada pela ECU do Trator no barramento de implemento para fornecer a qualquer sistema conectado na rede a velocidade e a distância percorrida em tempo real medida em relação ao solo, é descrita da seguinte forma:

Página de dados:	0;
Formato de PDU:	254 (FE ₁₆);
PDU Específico:	73 (49 ₁₆);
PGN:	65097 (00FE49 ₁₆);
<i>bytes</i> 1, 2:	velocidade;
3 a 6:	distância percorrida;
7:	reservado;
8:	<i>bits</i> 1, 2: sentido;
	<i>bits</i> 3 a 8: reservado;

Por sua vez, os parâmetros que estão no campo de dados também são definidos na norma e, neste caso específico, estão descritos da seguinte forma:

velocidade:

extensão de dados: 2 bytes;
 resolução: 0,001 m/s/bit;
 offset: 0 m/s;
 limite: 0 a 64,255 m/s;

distância percorrida:

extensão de dados: 4 bytes;
 resolução: 0,001 m/bit;
 limite: 0 a 4.211.081,215 m/s;

sentido:

extensão de dados: 2 bits;
 definição:
 00 p/trás
 01 p/ frente
 10 indicação de erro
 11 não disponível

Como exemplo, uma mensagem *Ground-Based Speed and Distance* é apresentada na Tabela 3. A primeira linha desta tabela indica o PGN e o campo de dados. O PGN é o identificador da mensagem, e é dividido em quatro campos: Página de Dados (DP – *Data Page*), Reservado (R), PDUF e PDUS, na segunda linha. O campo de dados, por sua vez, possui oito bytes: D0 a D7. A terceira linha apresenta valores respectivos aos campos da linha anterior.

Tabela 3 – Mensagem *Ground-Based Speed and Distance*.

PGN				Campo De Dados							
R	DP	PDUF	PDUS	D0	D1	D2	D3	D4	D5	D6	D7
0	0	FE ₁₆	49 ₁₆	A9 ₁₆	0D ₁₆	59 ₁₆	C4 ₁₆	00 ₁₆	00 ₁₆	FF ₁₆	7F ₁₆

A mensagem apresentada na Tabela 3 pode ser interpretada da seguinte forma:

Velocidade: $0DA9_{16} = 3,497 \text{ m/s}$;
 Distância percorrida: $C459_{16} = 50,265 \text{ m}$;
 Sentido: $7F_{16} = \underline{0}111\ 1111_2 = \text{p/ frente}$;

As mensagens são divididas em dois formatos quanto ao endereço destino. As mensagens podem ser enviadas para um endereço específico, em Formato de PDU 1, ou para o endereço global (*broadcast*), em Formato de PDU 2. A identificação dos dois tipos é feita no campo PDUF. Quando o valor do PDUF é menor do que 240 ($F0_{16}$), a mensagem é enviada a um endereço específico. Neste caso, o campo PDUS é utilizado para o endereço específico, e não é considerado no PGN. No segundo caso, o valor do PDUF é maior ou igual a 240, e o campo PDUS é uma extensão para a identificação do PGN. Neste caso não há destino específico, e todas as ECU devem receber a mensagem. A Tabela 4 exhibe a interpretação dos dois formatos de PDU.

Tabela 4 – Formato PDU 1 e formato PDU 2 (Adaptado de ISO 11783, 1998)

Formato	Formato PDU	PDU Específico	PGN
Formato de PDU 1	0	endereço destino, ou <i>destination address</i> (DA)	0 (0x0000)
	1	DA	256 (0x0100)
	...	DA	...
	238	DA	60928 (0xEE00)
	239	DA	61184 (0xEF00)
Formato de PDU 2	240	0	61440 (0xF000)
	240	1	61441 (0xF001)
	...	...	...
	255	254	65534 (0xFFFE)
	255	255	65535 (0xFFFF)

Como o PGN considera o campo página de dados, o número máximo de PGN possível a ser utilizado é duplicado. No total, são possíveis $(240 + 16 * 256) * 2 =$

8672 PGN. Todos os PGN atribuídos estão descritos no anexo A da parte 1 da norma ISO 11783.

A prioridade das mensagens, o acesso ao meio, a arbitragem e a detecção de erros estão de acordo com o protocolo CAN.

O TP é descrito na parte da camada de enlace de dados. O TP tem como objetivo o empacotamento e desempacotamento de mensagens com tamanho entre 9 e 1785 *bytes*, e também o gerenciamento da conexão do transporte. Existem dois PGN atribuídos para o TP: gerenciamento de conexão, ou *Connection Management* (CM) e a transferência de dados, ou *Data Transfer* (DT).

3.2.4 Camada de Rede

A camada de rede é definida na parte 4 da ISOBUS. Esta parte descreve os serviços e os requisitos necessários para a comunicação entre ECU em diferentes segmentos de uma rede ISOBUS. São descritas as ECU especiais, denominadas unidades de interconexão de rede, ou *Network Interconnection Unit* (NIU), para a conexão de sub-redes. As funções realizadas pelas NIU são:

- Encaminhamento: repassar uma mensagem de um barramento a outro, na mesma taxa de transmissão. Todas as mensagens são repassadas. É permitido atraso máximo de 10% no tempo de cada *bit*, ou seja, 400 ns a 250 kbps;

- Filtragem: repassar uma mensagem de um barramento a outro condicionalmente. É permitido atraso de 50 ms na filtragem de mensagens;
- Translação de endereços: re-mapear os endereços das mensagens entre sub-redes. Desta maneira, um barramento inteiro pode ser representado como uma simples ECU em outro barramento.
- Remontar parâmetros no campo de dados de mensagens, para facilitar a transferência, recepção e interpretação pelas ECU.

Na filtragem, translação de endereços e remontagem, é necessário um banco de dados para executar tais funções automaticamente.

Existem cinco tipos de NIU em uma rede ISOBUS: repetidor, ponte (*bridge*), roteador, *gateway* e a TECU. Um repetidor realiza apenas a função de encaminhar mensagens. Uma ponte encaminha e filtra mensagens. Um roteador realiza as funções anteriores, e também a translação de endereços. Um *gateway* realiza todas as quatro funções descritas. Finalmente, a TECU é considerada uma ECU única, realizando a função de *gateway* entre o barramento de trator e o barramento de implemento, somando outras funções que serão apresentadas adiante. Todas as NIU devem ser transparentes na rede para qualquer ECU.

3.2.5 Camada de Gerenciamento de Rede

A camada de gerenciamento de uma rede ISOBUS é definida na parte 5 da ISO 11783. Esta parte descreve o gerenciamento de endereços para ECU, os tipos de ECU, tanto com relação às funcionalidades quanto à configuração de endereço, e aos procedimentos de inicialização.

A configuração de um endereço é o processo realizado por uma ECU para a obtenção de um endereço que a identifica na rede. Há três tipos de ECU com relação às funcionalidades: ECU padrão, ECU de diagnóstico ou desenvolvimento e ECU de interconexão de redes, descritos por Sousa (2002).

Há quatro tipos de ECU, com relação à configuração de endereços: ECU de endereço não configurável, ECU de endereço configurável em manutenção, ECU de endereço configurável por comando e ECU de endereço auto-configurável, descritos por Sousa (2002).

Nesta parte da norma também é definida uma estrutura chamada *NAME*. Esta estrutura de dados é composta por 64 *bits* divididos em diversos campos, os quais descrevem a função e o posicionamento de uma ECU na rede ISOBUS. O valor numérico do *NAME* também é utilizado na arbitragem de disputas por um mesmo endereço, de forma que as funções de maior relevância tenham maior prioridade. Assim, toda ECU deve ter um *NAME* para obter um endereço na rede e poder descrever suas características e funcionalidades a todas as outras ECU em uma rede ISOBUS. Os campos do *NAME* e a descrição dos mesmos são detalhados por Sousa (2002) e Guimarães (2003).

A negociação para a obtenção de um endereço é realizada através de quatro mensagens específicas, que estão detalhadas na Tabela 5.

Tabela 5 – Mensagens de gerenciamento de rede (Adaptado de ISO 11783, 2001)

Mensagem	PGN	Endereço de Origem, ou <i>Source Address</i> (SA)	Endereço Destino	Dados (nº de bytes)
<i>Address Claim</i> (Reivindicação de Endereço)	EE00 ₁₆	endereço pretendido	FF ₁₆ (global)	NAME (8)
<i>Cannot Claim Source Address</i> (Impossibilidade de Obtenção de Endereço)	EE00 ₁₆	FE ₁₆ (nulo)	FF ₁₆ (global)	NAME (8)
<i>Request for Address Claimed</i> (Pedido de Endereços Utilizados)	EA00 ₁₆	endereço de origem	endereço pretendido	PGN 60928 (3)
<i>Commanded Address</i> (Configuração de Endereço)	FED8 ₁₆	endereço de origem	global (PDU 2)	NAME e novo endereço de origem (9)

A mensagem “*Address Claim*” é utilizada para reivindicar, declarar, ou sustentar um endereço na rede. A reivindicação ocorre quando uma ECU entra na rede e procura obter um endereço. A declaração ocorre quando o endereço que já é utilizado por uma ECU é pedido por outra ECU (mensagem “*Request for Address Claimed*”). A sustentação ocorre em resposta à outra ECU, quando esta reivindicar seu endereço, mas possui menor prioridade.

A mensagem “*Cannot Claim Source Address*” é utilizada para declarar a impossibilidade de obtenção de endereço na rede por uma ECU. O PGN é idêntico ao da mensagem “*Address Claim*”, porém a diferença está no endereço de origem utilizado. Como a ECU não possui um endereço na rede, ela deve utilizar o endereço nulo. Neste estado, a ECU deve somente aguardar uma configuração remota (mensagem “*Commanded Address*”), caso esta seja habilitada a receber tal comando. A ECU também pode responder a pedido global de endereços (mensagem “*Request*”).

for Address Claimed”), porém com o endereço nulo, ou seja, a própria mensagem “*Cannot Claim Source Address*”.

A mensagem “*Request for Address Claimed*” é utilizada para verificar os endereços que já estão sendo utilizados. Caso o endereço destino seja global, todas as ECU devem responder com a mensagem “*Address Claim*”, declarando a utilização de seus endereços. Se houver ECU sem endereços definidos, elas também podem responder com a mensagem “*Cannot Claim Source Address*”.

A mensagem “*Commanded Address*” é utilizada para configurar o endereço de uma ECU, caso seja necessário. Esta mensagem pode ser enviada por uma NIU, tal como uma ponte, ou por uma ECU de manutenção. Esta mensagem possui nove (9) dados e, portanto deve-se utilizar um protocolo de transporte para enviá-la.

O procedimento de inicialização de uma ECU para a obtenção de um endereço de origem também é definido na parte 5 da norma, e será descrito em detalhes posteriormente, no Capítulo 5.1.

3.2.6 Terminal Virtual

O Terminal Virtual, ou *Virtual Terminal* (VT), é um dispositivo de interação do operador com as funcionalidades do trator e do implemento agrícola. O VT é uma ECU baseada no conceito de Interface Homem-Máquina (IHM). Este dispositivo deve localizar-se na cabine do trator e deve conectar-se ao barramento de implemento. A parte 6 da norma ISOBUS apresenta definições, características físicas do dispositivo e o comportamento dinâmico em relação a procedimentos de

inicialização, procedimentos de atualização de dados, tratamento de alarmes e manipulação de diferentes tipos de objetos. Os anexos desta parte da norma contêm informações diversas sobre os objetos, os eventos, os comandos, as mensagens entre ECU e VT, e o protocolo de transporte estendido, ou *Extended Transport Protocol* (ETP).

Para os fabricantes deste tipo de terminal, o projeto de *hardware* tornou-se limitado aos requisitos da norma, quanto à quantidade de teclas, tamanho de tela e quantidade de cores. Estas restrições permitem alcançar a compatibilidade entre diferentes fabricantes de VT e ECU de implementos agrícolas. No entanto, a norma ISOBUS especifica entradas que são programáveis, como as *soft key* (teclas com funções programáveis) e controles auxiliares. Estas opções permitem que cada dispositivo execute uma função diferente, de acordo com o programa estabelecido.

Qualquer ECU, ou conjunto de ECU que juntas realizam uma determinada função, denominado *working set* (WS), ou conjunto de trabalho, poderá utilizar o VT para exibir suas funcionalidades ao operador. É determinado um procedimento para que um WS envie um arquivo, contendo um grupo de objetos, denominado *Object Pool* (OP), ou Conjunto de Objetos. Cada objeto tem um comportamento específico e definido, e possui um conjunto específico de atributos, semelhante à orientação objeto. Existem trinta e um tipos de objetos definidos no anexo A.1 da parte 6 da norma ISOBUS. Landi (2004) apresenta a descrição de todos os objetos definidos pela norma ISOBUS.

A função básica do VT é a interface entre o operador e a máquina. Por isso, este dispositivo deve ser capaz de apresentar imagens e dados através de um monitor e receber comandos do operador, através de teclas ou tela sensível a toque (*touch screen*). Este monitor deve conter endereçamento por *pixel* (menor elemento

em um dispositivo de exibição de imagem), ou seja, os objetos são dimensionados e posicionados em função da quantidade de *pixels*. A Figura 9 ilustra o monitor, que é dividido em duas áreas: uma para máscara de dados e outra para máscara de *soft key*.

Figura 9 – Terminal Virtual (Adaptado de ISO 11783, 2004).

O tamanho mínimo para a máscara de dados é de 200 x 200 *pixels*. Outros valores são possíveis: 240 x 240, 320 x 320 e 480 x 480 *pixels*. O tamanho mínimo para a máscara de *soft key* é de 60 x 32 *pixels*. Cada WS pode solicitar informações sobre o tamanho do *display* do VT, antes de enviar o OP. É recomendável que todo WS seja capaz de redimensionar os objetos para apresentar um OP adequado ao tamanho do *display*. Outra maneira é conter em sua memória quatro OP, um para cada tamanho de *display*, porém, dependendo da memória disponível, isto seja inviável.

Os procedimentos de inicialização de um WS com um VT e o procedimento para a utilização do ETP estão descritos no Capítulo 5 – Resultados.

3.2.7 Mensagens de Implemento

A parte 7 da norma ISOBUS especifica um conjunto de mensagens utilizadas entre tratores e implementos agrícolas. Estas mensagens implementam as necessidades básicas, como as informações do trator para os implementos agrícolas, ou como o controle de funções realizadas entre os mesmos. As informações suportadas são: tempo (data/ hora), velocidade em relação ao solo, distância, navegação, parâmetros do eixo de tomada de potência do trator, ou *Power Take-Off* (PTO), engate de três pontos, processamento de dados (*process data*) e parâmetros das funções de iluminação.

Os anexos desta parte apresentam definições de parâmetros, PGN e mensagens citadas acima para controle do trator pelo implemento.

Guimarães (2003) apresenta maiores detalhes de faixas de valores dos possíveis parâmetros de dados, parâmetros discretos e parâmetros de controle que formam as mensagens.

3.2.8 Mensagens de Trem de Força (*Power Train*)

A parte 8 da norma ISOBUS descreve mensagens necessárias ao monitoramento e controle do trem de força em tratores e implementos auto propelidos. Estas mensagens não estão explicitadas na norma ISOBUS, pois fazem referência integral ao documento SAE J1939/71, com ressalva a possíveis PGN

duplicados, que neste caso, deve-se optar pelo PGN apresentado na norma ISOBUS.

Estas mensagens trafegam no barramento de trator, separadamente das mensagens de implemento, que por sua vez trafegam no barramento de implemento. A ECU do Trator, que é um *gateway* entre os dois barramentos, é responsável por encaminhar as informações relevantes de um barramento a outro, realizando as modificações de PGN, endereço origem e endereço destino relativos a cada barramento.

3.2.9 ECU do Trator

A parte nove da ISO 11783 descreve a ECU do Trator, ou *Tractor ECU* (TECU). Esta ECU é um *gateway* entre o barramento de trator e o barramento de implemento, como mostrado na Figura 10.

Figura 10 – Topologia de rede em um trator em detalhes.

A TECU representa o trator e suas funcionalidades. As informações do trator, como por exemplo velocidade, posição do engate e PTO devem ser coletadas do

barramento de trator e enviadas no barramento de implemento, com o endereço de origem da TECU. Por outro lado, mensagens de implementos para controlar funcionalidades do trator são enviadas com endereço destino para a TECU, que gerencia e redireciona os comandos no barramento de trator, utilizando seu próprio endereço.

Existem três classes de TECU, de acordo com um conjunto mínimo de mensagens suportadas. A classe 1 fornece medições básicas do trator. O fato de permitir a conexão de sensores existentes em uma ECU simples conectada em um barramento habilita fabricantes de tratores se tornarem ISO compatíveis rapidamente. Esta classe não é apropriada para novos projetos, e visa apenas adaptações de tratores antigos. A classe 2 fornece um conjunto completo das medições do trator. A classe 3 abrange TECU que aceitam comandos de implementos, por exemplo, do controle da posição do engate, ou rotação do PTO.

A TECU também é responsável por controlar a iluminação e fornecer potência para os implementos. O controle de iluminação compreende o uso de mensagens de iluminação especificadas na parte 7 da ISO 11783. A potência é fornecida sob dois condutores, PWR e ECU_PWR, através do conector IBC. A TECU envia mensagens indicando a condição da potência, e também pode receber controle da potência, caso esta seja originada no trator.

3.2.10 Controlador de Tarefas

O Controlador de Tarefas, ou *task controller* (TC), é responsável pelo gerenciamento de tarefas a serem realizadas no campo. Esta ECU envia comandos para o implemento agrícola dependendo da tarefa a ser realizada, por exemplo, para o controle de taxas de aplicação de um determinado insumo agrícola, ou manejo do solo. O TC gerencia diversas informações durante uma tarefa como, por exemplo, velocidade, posicionamento global, mapas de aplicação e aquisição de dados da aplicação real. A parte 10 da ISO 11783 apresenta definições, procedimentos de inicialização do TC e configuração com WS, aquisição de dados de tarefas e transferência de dados em formato XML (*Extensible Markup Language*). O formato XML, adotado pela ISOBUS, é uma linguagem de marcação de dados que provê um formato para descrever dados estruturados (XML, 2001).

Os anexos desta parte da norma apresentam uma lista descrevendo todos os tipos de objetos (máquinas ou componentes) que possuem alguma funcionalidade, definições de mensagens, atributos dos elementos XML e um diagrama de relações entre os mesmos.

As tarefas a serem realizadas no campo são preparadas anteriormente a partir de diversos dados como, por exemplo, dados coletados do solo e dados de colheitas anteriores. As tarefas são gerenciadas em um computador da fazenda, definido pela norma como *Farm Management Information System* (FMIS). Um mapa da tarefa é gerado, e as informações devem estar no formato XML, de acordo com a norma.

A forma de transferência dos dados do FMIS para o TC não é padronizada. Ela pode ser enviada por cartão de memória, discos removíveis ou transferência de dados sem fio.

O TC utiliza o VT para a interação com o operador, ou seja, o TC se torna um cliente do VT, equivalente a um WS. Desta maneira o operador pode escolher uma determinada tarefa, assim como seu início ou cancelamento, de acordo com sua necessidade.

Nas tarefas estão identificados os parâmetros para aquisição de dados, ou seja, as informações coletadas durante as tarefas que voltarão para o FMIS.

As mensagens relacionadas ao Controle de Tarefas utilizam principalmente um PGN específico: a mensagem “*Process Data*” (processamento de dados). O TC utiliza os dados de tarefas nestas mensagens para controlar os implementos agrícolas.

3.2.11 Dicionário de Dados

O dicionário de dados (*data dictionary*) é uma especificação de parâmetros de elementos contidos nas mensagens de processamento de dados, apresentadas na parte 7 e 10 da norma ISOBUS.

Esta especificação abrange número de identificação, definição, limites, resolução e unidades destes elementos. Um exemplo de elemento utilizado na é apresentado na Figura 11.

DD Entity	1 - Setpoint Volume Per Area Application Rate
Definition	Setpoint Application Rate specified as volume per area
Comment	test
Device Class(es)	4 - Planters /Seeders 5 - Fertilizer 6 - Sprayers 10 - Irrigation
Unit Symbol	mm ³ /m ² - Capacity per area unit
Resolution	0,01
SAE SPN	not specified
Range	0 to 2
Submit by	Part 10 Task Force
Submit Date	
Submit Company	89 - Kverneland Group, Electronics Division
Current Status	ISO-Published
Status Date	02.02.2005
Request Document	
Annex Release Date	01.03.2005

Figura 11 – Elemento contido no dicionário de dados (ISO 11783, 2007b).

A parte 11 da norma ISOBUS apenas indica o local de acesso deste documento, disponibilizado no *site* www.isobus.net.

VDMA **ISOBUS**
Communication System

About ISOBUS ISOBUS up-to-date ISOBUS standards Data Dictionary Press Downloads Contact

ISOBUS Data Dictionary

Member Login Area

Please login with your Username and Password.

Username

Password

Login

[Register Now](#) | [Forgot Password](#)

Information for first-time registrants:
To register your user access data for this online database, please proceed as follows:
1) Fill out the desired text for Username and Password and confirm by pressing the button 'Register Now'. The password should contain at least 6 characters.
Fill out the Registration Form and press the button 'Register'.
2) Send a request to unlock your user access data to the administrator of this online database (Email: gerhard.henninger@vdma.org). The administrator will inform you about your registration as soon as possible.

Figura 12 – Página de registro na web, para acesso ao dicionário de dados. (ISOBUS, 2007)

É necessário realizar um registro, preenchendo um formulário, e em seguida enviar um e-mail para o endereço eletrônico indicado para desbloquear o acesso ao documento, como mostra a Figura 12.

3.2.12 Sistema de Diagnóstico

A parte 12 da norma ISOBUS fornece um sistema padronizado para diagnóstico da rede. O sistema de diagnóstico solicita a todas as ECU de uma rede ISOBUS o fornecimento de informações específicas. Estas informações, padronizadas pela norma ISOBUS, permitem um operador ou técnico realizar o diagnóstico da rede e determinar quais ECU estão com problemas, ou estão funcionando, porém inadequadamente.

Há três níveis de capacidade de diagnósticos. O diagnóstico deve ser aplicado a cada ECU, em função de seu nível. Os níveis são: “0” (zero) – nível de diagnóstico não compatível com a norma ISOBUS, “1” (um) – nível básico de diagnóstico compatível com a norma ISOBUS e finalmente “2” (dois) – nível otimizado de diagnóstico, que suporta tanto o nível básico (1) especificado na norma ISOBUS quanto outras padrões como o ISO 15765 e SAE J1939-73.

Os anexos desta parte da norma apresentam definições de parâmetros de diagnósticos, definições de mensagens de diagnósticos, configuração de rede com um sistema de diagnóstico em funcionamento, e sugestões de telas de configurações.

Figura 13 – Topologia de rede com sistema diagnóstico conectado (adaptado de ISO 11783, 2007c).

A Figura 13 ilustra um diagrama que representa o sistema de diagnóstico operando em uma rede. Nota-se que três barramentos podem estar conectados no sistema de diagnóstico: o barramento de implemento (item 13, Figura 13), o barramento de trator (item 11, Figura 13) e até mesmo um barramento proprietário (item 9, Figura 13).

Figura 14 – Conector padrão de diagnóstico (POWELL, 2007).

A Figura 14 ilustra o conector de diagnóstico padrão especificado na parte 2 da norma ISOBUS. Este conector se refere ao item 5 da Figura 13. Após a realização de um diagnóstico na rede, este conector é desconectado do barramento, e o sistema retorna ao funcionamento normal.

3.2.13 Servidor de Arquivos

A parte treze da norma ISOBUS descreve uma ECU denominada Servidor de Arquivos (*File Server*). É uma ECU opcional que tem como objetivo armazenar dados de qualquer ECU da rede, via comunicação CAN. As mensagens para o envio dos dados e para realizar funções com os arquivos são definidas nesta parte. No anexo A desta parte encontram-se definições de parâmetros como grupos de comando, funções de comando e estado. No anexo B desta parte encontram-se definições das mensagens separadas em grupos, tais como mensagens para transferência de dados, volume de dados, acesso a arquivos, operações com arquivos e operações com diretórios.

3.2.14 Funções Automatizadas

A mais recente parte da norma ISOBUS, a parte quatorze, refere-se a funções automatizadas. Esta parte ainda está em estágio de desenvolvimento. Atividades e tarefas repetitivas no campo podem ser automatizadas para simplificar as tarefas de um operador, aumentando o conforto e diminuindo a suscetibilidade a erros. As funções automatizadas serão opcionais para os desenvolvedores e fabricantes, não sendo obrigatórios para um sistema ISOBUS compatível.

3.2.15 *Working Set* (Conjunto de Trabalho)

O *working set* (conjunto de trabalho) é definido como um grupo de ECU que em conjunto realizam funções relacionadas a uma tarefa comum. Um WS também pode ser representado por apenas uma ECU.

O conceito da divisão “virtual” das ECU em grupos reduz o fluxo de informações na rede, otimizando a comunicação. Um exemplo pode ser verificado em um implemento agrícola que aplica determinado insumo em seis linhas no campo (Figura 15). Se cada linha possuir uma ECU que controla a taxa de aplicação, cada ECU se comunicaria com o Terminal Virtual, o Controlador de Tarefas e a ECU do Trator. Considerando a aplicação deste insumo uma tarefa realizada em conjunto pelas seis ECU, a comunicação deste grupo pode ser realizada por apenas uma ECU, denominada mestre do WS, ou *Working set master* (WSM). A WSM coordena esta comunicação, trocando informações tanto com o trator (VT e TC) quanto com

as outras ECU do implemento, denominadas membros do WS, ou *Working set member*. Deste modo, evita-se a comunicação entre todas as ECU do implemento com o trator.

Figura 15 – Divisão virtual de ECU em um grupo, denominado *Working Set*.

A parte 7 da norma ISOBUS define duas mensagens para a identificação de *working set master* e *working set member*. Elas são apresentadas na Tabela 6, com indicação de localização (anexo) na parte 7 da norma ISOBUS.

Tabela 6 – PGN para identificação do *Working Set Master* e *Working Set Member*.

Mensagem	PGN	Descrição
<i>Working set master</i> (anexo B.23.1)	65037 (00FE0D ₁₆)	mensagem enviada por um WSM para indicar quantas ECU fazem parte do WS.
<i>Working set member</i> (anexo B.23.2)	65036 (00FE0C ₁₆)	mensagem enviada por um WSM para indicar um membro de um conjunto WS.

A mensagem “*Working Set Master*” é a identificação de um WS, o qual deve ser associado ao endereço de origem da ECU que o enviou, para comunicações posteriores.

3.2.16 *Object Pool* (Conjunto de Objetos)

O *Object Pool* é um arquivo de dados que contém um grupo de objetos que representa graficamente um WS em um Terminal Virtual. Através deste recurso, o WS é capaz de enviar dados para o operador, e também disponibilizar comandos para ativar funções ou mudar configurações do mesmo. Um WS poderá apresentar somente um OP no VT, cujos objetos podem ter seus atributos modificados a qualquer momento, por mensagens definidas. O OP é um conjunto de *bytes* no qual os objetos que o definem estão dispostos seqüencialmente. A interpretação dos *bytes* na seqüência direta decifra os limites de cada objeto. Na Figura 16 é apresentado um exemplo de conjunto de objetos.

0,0,0,1,1,232,3,1,0,1,249,42,5,0,20,0,69,78,232,3,1,1,160,15,3,0,178,54,10,0,6,0,178,54,10,0,80,0,4,43,10,0,18,0,184,11,3,180,0,60,0,0,4,0,253,42,12,0,6,0,22,4,4,6,18,0,30,0,176,54,0,0,0,0,254,42,110,0,35,0...

Figura 16 – Conjunto de *bytes* exemplificando três objetos em um OP

Neste conjunto de *bytes*, parte de um OP, há apenas três objetos, assim como a descrição de cada um deles. A extensão de cada objeto é variável. Apenas os três primeiros *bytes* são constantes: os dois primeiros são o identificador do objeto, e o terceiro é o tipo de objeto. Os campos seguintes são identificados segundo cada tipo de objeto. Após o último campo ser identificado, o próximo *byte* da seqüência é o primeiro *byte* do próximo objeto, se houver. Como exemplo, o primeiro objeto do conjunto de *bytes* apresentado anteriormente é detalhado *byte* a *byte* na Figura 17. Nesta figura, é verificado a identificação do objeto (número “0”) e o tipo de objeto (*Object Pool*). Este tipo de objeto é utilizado para identificar um OP no VT. Os parâmetros deste objeto são cor de fundo (um *byte*, lista de cores na tabela A.3,

anexo A da parte 6 da norma ISOBUS), selecionável (0 = falso, 1 = verdadeiro), máscara ativa (dois *bytes*, identificação da máscara de dados (*data mask*) inicial quando OP é selecionado no VT), número de objetos contidos (quantidade de objetos que estão contidos neste objeto), número de macros contidas (quantidade de macros que estão associadas a este objeto), número de idiomas contidos (quantidade de idiomas associados a este OP), objetos (seis *bytes* cada, se houver), macros (dois *bytes* cada, se houver) e idiomas (dois *bytes* cada, se houver).

Figura 17 – Campos do OP, variável com o tipo de objeto

Verificado o final de cada objeto, os próximos *bytes* serão interpretados da mesma maneira, e assim por diante até o fim do OP. Quando o VT recebe uma sequência de *bytes* relativos a um OP, ele deve ser capaz de separar os objetos, identificando o tipo e os dados de cada um, e verificar se aquele OP é válido para ser exibido no *display*. Se o VT encontra um erro em algum objeto, por exemplo, dados que não estão de acordo com a norma ou objetos referenciados, porém inexistentes, o VT deve invalidar o OP, com uma mensagem determinada e indicação do erro, para que o WS aceite a resposta.

4. MATERIAIS E MÉTODOS

A implementação de uma máquina agrícola compatível com a norma ISOBUS teve como principal objeto de estudo o protocolo CAN e a norma ISOBUS, detalhados no capítulo 3 – Protocolos.

A metodologia utilizada neste trabalho consistiu na implementação e testes de rotinas de programas para a realização de procedimentos da norma ISOBUS. A seqüência desta implementação foi baseada no procedimento de configuração de um WS com um Terminal Virtual. Segundo o capítulo 4.6.4.b – Inicialização de um WS – na parte 6 da norma ISOBUS, a seguinte seqüência deve ser realizada na configuração de um OP com um Terminal Virtual:

1. O WSM deve completar o processo de obtenção de endereço;
2. O WSM deve aguardar a transmissão da mensagem “VT *Status*” (Estado do VT);
3. O WSM deve identificar-se como mestre e também identificar os membros do grupo, se houver;
4. O WSM deve iniciar o envio periódico da mensagem “*Working Set Maintenance*” (manutenção do WS);
5. O WSM deve pedir ao VT a mensagem “*Language Command*” (Comando de Língua), que contém o idioma, formato e unidades utilizados na rede;
6. O WSM pode pedir ao VT o envio de parâmetros para identificar as suas capacidades;

7. O WSM pode pedir ao VT a informação sobre existência de OP em sua memória não-volátil;
8. A transferência do OP deve ser iniciada e completada. Isto pode ser feito pedindo ao VT para carregar o OP da memória não-volátil ou utilizando um protocolo de transporte definido na norma ISOBUS.

Esta seqüência foi organizada em um fluxograma (Figura 18).

Figura 18 – Seqüência de procedimentos de configuração de um WS com um Terminal Virtual.

A implementação dos códigos partiu de um programa existente, a biblioteca SAE J1939 (MICROCHIP, 2007). Esta biblioteca possui uma estrutura de dados e um conjunto de funções para a realização do primeiro passo da seqüência

apresentada na Figura 18. A partir deste programa, foram sendo implementadas novas variáveis e novas funções para a realização dos passos seguintes desta seqüência. No desenvolvimento de cada passo, o controlador foi submetido a testes de compatibilidade com equipamentos em conformidade com a norma ISOBUS. Os testes permitiram verificar a troca de mensagens entre os equipamentos, e assim foi possível encontrar erros durante a programação, corrigindo-os para a correta comunicação exigida pela norma. A metodologia adotada pode ser verificada na Figura 19.

Figura 19 – Metodologia para a implementação do procedimento de configuração de um WS.

A metodologia apresentada na Figura 19 foi utilizada em dois estudos de caso, que serão apresentados posteriormente. A seguir são apresentados os materiais utilizados para auxiliar na implementação de cada passo do procedimento de configuração de um WS com um Terminal Virtual.

O primeiro passo é a configuração de uma ECU na rede. Este é o procedimento para a obtenção de endereço. Para este procedimento foram utilizados os seguintes materiais: a parte 5 – Gerenciamento de rede – da norma ISOBUS e a biblioteca SAE J1939 (2007). A parte 5 da norma ISOBUS, como visto anteriormente no capítulo 3, apresenta todos os parâmetros e mensagens para a implementação deste procedimento. A biblioteca SAE J1939 é baseada no padrão SAE J1939, desenvolvido pela SAE. Este padrão faz uso do protocolo CAN e implementa mensagens automotivas, recomendado para aplicação em ônibus e caminhões (SAE J1939, 2007). Como visto anteriormente, foi um dos padrões utilizados para conceber a norma ISOBUS, junto ao padrão LBS. A biblioteca contém a implementação da configuração de uma ECU na rede, em código C.

O segundo passo é a espera de uma mensagem “VT *Status*”. Esta mensagem indica que existe um terminal na rede, e contém dados sobre sua ocupação. Neste passo foram utilizados dados contidos na parte 6 – Terminal Virtual – da norma ISOBUS. O anexo G desta parte apresenta os parâmetros da mensagem “VT *Status*”.

O terceiro passo é a identificação do WS. Este procedimento é realizado por duas mensagens. A primeira, chamada “*Working set Master*”, identifica o mestre do grupo, e contém o número de ECU que formam o conjunto. A segunda, chamada “*Working set member*”, identifica cada membro pelo campo *NAME*. Estes dois PGN estão detalhados no anexo B da parte 7 – Mensagens de implemento – da norma ISOBUS.

O quarto passo é o início do envio cíclico da mensagem “*Working Set Maintenance*”, uma vez que este já foi apresentado ao VT. Esta mensagem garante ao WS o reconhecimento de sua presença pelo VT. Este reconhecimento garante os serviços oferecidos pelo VT. Neste passo, como no segundo passo, são utilizados dados contidos no anexo G da parte 6 – Terminal Virtual – da norma ISOBUS.

O quinto passo é o pedido da mensagem “*Language Command*”, que identifica o idioma, unidades e formatos (data, hora e unidades) utilizados na rede. Este PGN é detalhado no anexo B da parte 7 – Mensagens de implemento – da norma ISOBUS.

O sexto passo é opcional, e se refere a pedido de informações técnicas do VT. Exemplos de informações são: indicação de memória disponível para o envio de um OP, tamanho do monitor e quantidade e tamanho de *soft key*. Neste passo foram utilizados dados contidos no anexo D da parte 6 – Terminal Virtual – da norma ISOBUS.

O sétimo passo, também opcional, se refere ao pedido de informações sobre versão de OP gravada em memória não volátil. Neste passo foram utilizados dados contidos no anexo E da parte 6 – Terminal Virtual – da norma ISOBUS.

O oitavo e último passo deste procedimento é o carregamento do OP no VT. Este procedimento pode ser feito pedindo ao VT para carregar o OP existente em memória não volátil ou enviando o OP através de um protocolo de transporte. Neste passo foram utilizados os seguintes materiais: parte 3 – Camada de Enlace de Dados – e parte 6 – Terminal Virtual – da norma ISOBUS e o *software Mask Generator*. O anexo B da parte 6 da norma ISOBUS apresenta em detalhes todos os objetos definidos e os respectivos atributos. Para cada atributo são definidos tipo (booleano, *byte*, inteiro ou ponto flutuante), tamanho em *bytes*, limite de valores e

descrição. O anexo C da parte 6 apresenta as mensagens para o gerenciamento do envio do OP. O protocolo de transporte utilizado para o envio do OP neste desenvolvimento foi o TP, pois o tamanho do OP não excedeu os 1785 *bytes*. Caso ultrapassasse este valor, o OP deveria ser enviado utilizando-se o ETP. As mensagens e procedimentos utilizados no TP são apresentados na parte 3 da norma ISOBUS. Para o desenvolvimento dos OP, foi utilizado o *software Mask Generator*, da empresa WTK (2007). Apesar da versão gratuita deste *software* não gravar a interface desenvolvida, ela foi útil no estudo de cada objeto e seus parâmetros.

Outros materiais foram utilizados no desenvolvimento da programação, nos testes e na implementação dos estudos de caso. Estes materiais são apresentados a seguir: uma rede CAN, o ambiente de programação MPLAB 7.30 (MICROCHIP, 2007), o ambiente de programação LabView (NI, 2006), o *software* CANoe.ISO11783 (VECTOR, 2007), o terminal GTA (AGCO, 2007), um trator (VALTRA, 2007) e aplicador de calcário (BALDAN, 2007).

A rede CAN é formada por um conjunto de ECU conectados em rede. As ECU utilizadas são ECU genéricas (SOUSA, 2002) que contém um microcontrolador PIC 18F258 (MICROCHIP, 2007), da família PIC18Fxx8 (Figura 20). Esta família possui microcontroladores de 16 *bits* com controlador CAN interno, periféricos como interfaces *Serial Peripheral Interface* (SPI), *Universal Synchronous Asynchronous Receiver Transmitter* (USART) e *Inter-Integrated Circuit* (I2C), comparadores, temporizadores, contadores e conversores analógico-digital. A memória de programa e de dados foi suficiente para desenvolver uma aplicação simples para o protótipo. Também foram utilizados transceptores MAX232 e MCP2551 para comunicação serial da ECU com um computador de mesa e comunicação CAN, respectivamente.

Figura 20 – Circuito eletrônico com interface serial e interface CAN, desenvolvida por Sousa (2002).

Para a programação dos microcontroladores foi utilizado um ambiente de desenvolvimento MPLAB 7.30, com compilador C18 e equipamento para gravação (MICROCHIP, 2007), anteriormente adquiridos no Laboratório de Simulação e Controle.

O programa LABVIEW é uma ferramenta para desenvolvimento de *softwares*. Diversos *softwares* foram desenvolvidos para diferentes aplicações. Uma delas foi um *software* para monitoramento de mensagens em uma rede ISOBUS (GODOY et al., 2007). O segundo *software* relevante foi desenvolvido para gerar o arquivo OP. O terceiro *software* desenvolvido utilizando esta ferramenta foi um WSM. Foram implementados todos os procedimentos para a configuração de um WS com um Terminal Virtual. Este *software* foi o primeiro estudo de caso desenvolvido neste trabalho, que veremos em detalhes posteriormente.

O principal *software* para desenvolvimento de projetos de controladores ISOBUS compatíveis utilizado foi o CANoe.ISO11783, da Vector (2007). CANoe é o nome da plataforma para teste, análise e desenvolvimento de sistemas embarcados. A extensão “ISO11783” é um dos módulos dessa plataforma, ou seja, utiliza o protocolo CAN e é específico para aplicações da norma ISO 11783. Outros módulos desta mesma plataforma são, por exemplo, o J1939, *Media Oriented System*

Transport (MOST), *Local Interconnect Network* (LIN), FlexRay, CANopen e NMEA2000. Neste trabalho foi utilizada uma interface PCMCIA (*Personal Computer Memory Card International Association*) chamada CANSARD, para a interface com o barramento CAN real (Figura 21).

Figura 21 – Cartão PCMCIA para interface do programa CANoe (VECTOR, 2007) com o barramento CAN.

O *software*, além de permitir analisar as mensagens, possui módulos para diversas aplicações, como simular um Terminal Virtual, desenvolver um programa de uma ECU específica, e permite a interação entre o barramento real e o barramento simulado, em tempo real.

O terminal GTA (AGCO, 2007), ilustrado na Figura 22, é um Terminal Virtual ISOBUS compatível e também contém o Controlador de Tarefas, Servidor de Arquivos e um gerenciador de informações de GPS. A união de diversos controladores em um único dispositivo físico auxilia na redução de cabos e conectores dentro da cabine do trator. Além disso, esta redução significa menor custo para as empresas que desenvolvem tais dispositivos. Este Terminal Virtual possui tela *touch screen*. Somente as *soft key* possuem teclas físicas. Há também uma entrada para cartão de memória SD (*Secure Digital*, cartão de memória padrão), para a transferência de dados entre o TC e o FMIS, como visto anteriormente no capítulo 3.2.10. O terminal GTA foi emprestado pela empresa

AGCO (2007). O GTA auxiliou nos testes do segundo protótipo, que será apresentado posteriormente, no Capítulo 5.5.2.

Figura 22 – Terminal GTA (AGCO, 2007).

O trator (modelo Valtra) e o aplicador de calcário (modelo DMP 7500) foram emprestados pela empresa Baldan (2007). O aplicador de calcário possui um tanque reservatório de calcário. O calcário é aplicado através de bocais na parte inferior do tanque, deslocado por eixos helicoidais, que são movidos por um eixo principal. Um motor hidráulico aciona o eixo principal, por meio de correntes. O aplicador de calcário foi utilizado no segundo estudo de caso, nas dependências da empresa Baldan, na cidade de Matão.

Figura 23 – Conjunto trator e aplicador de calcário (BALDAN, 2007).

Além dos materiais relacionados anteriormente, dois eventos contribuíram diretamente para auxiliar o desenvolvimento deste trabalho. O primeiro foi um curso

da empresa Vector (2007) e o segundo foi o Primeiro Workshop ISOBUS Brasil, patrocinado pela FTI (2007) Brasil.

O curso foi ministrado por um especialista na área: Bradford Cox, representante da Vector. O curso teve duração de três dias, e foi realizado nas dependências da EMBRAPA Instrumentação, São Carlos, em dezembro de 2006. O material apresentado em conjunto com a aplicação prática proporcionou um ótimo conhecimento técnico no assunto. O protocolo ISO 11783, a plataforma CANoe e as ferramentas deste programa aplicadas na ISO 11783 foram os temas abordados neste curso.

O Workshop ISOBUS Brasil foi preparado pelos membros da FTI Brasil, em março de 2007, nas dependências da EMBRAPA Instrumentação, São Carlos. Este evento obteve um resultado excelente, com a participação de nomes importantes neste contexto como, por exemplo, William Rudolph, Mike Schmidt, Hans Jürgen Nissen, Peter Fellmeth e Christian Büngener. Estes representantes ofereceram palestras com informações técnicas avançadas (BÜNGENER, FELLMETH, NISSEN, RUDOLPH; SCHMIDT, 2007) e relatos de experiências em *plugfest*, desenvolvimento e testes de aplicações relativas à norma ISOBUS. Ao fim do evento foi realizado um *plugshow*, com demonstrações de produtos ISOBUS compatíveis em bancada (FTI, 2007).

5. RESULTADOS E DISCUSSÕES

Neste capítulo são apresentados os resultados deste trabalho executado sobre os dois estudos de caso desenvolvidos durante a pesquisa. Foram considerados como resultados, as operações e implementações realizadas e atingidas com sucesso. É composto pela implementação de rotinas de programa que obedecem ao método proposto (configuração de um WS) e pela troca de mensagens obtidas nos testes. Esses resultados estão acompanhados de comentários e discussões que foram considerados importantes para a implementação e não discutidos na norma.

5.1 Configuração de uma ECU na rede

A norma ISOBUS determina quatro mensagens específicas para o procedimento de inicialização (Tabela 5). Por regra, as ECU que ainda não possuem endereço não devem enviar qualquer mensagem na rede, exceto as mensagens relativas ao gerenciamento de rede.

A biblioteca SAE J1939 implementa a configuração e disputa de endereço. Esta biblioteca foi utilizada no desenvolvimento deste trabalho, porém foram necessárias algumas adaptações para aumentar a eficácia.

O procedimento implementado nesta biblioteca é baseado na inicialização de uma ECU com endereço não configurável. Quando duas ECU com endereço não configurável disputam o mesmo endereço, a ECU com menor prioridade, ou seja,

com maior valor no campo *NAME*, anuncia a impossibilidade de obter um endereço. Nos testes realizados, não havia uma ferramenta para diagnosticar este evento e corrigi-lo. Neste caso, aquela ECU interrompe a comunicação na rede. Este exemplo, previsto pela norma, é ilustrado nas Figura 24 e Figura 25.

Figura 24 – Disputa de endereço: ECU A entra na rede e ganha disputa (Adaptado de ISO 11783, 2001).

Nos dois exemplos ilustrados nas Figura 24 e Figura 25, a ECU A entra na rede após a ECU B, que já possui um determinado endereço "X". Na Figura 24, a ECU A possui o *NAME* menor, ou seja, maior prioridade. Assim, a ECU B reconhece que perdeu a disputa e envia a mensagem "*Cannot Claim Source Address*", para informar toda a rede que não tem endereço.

1. Conseqüentemente, o PGN não considera o segundo *byte*. O PGN equivale a $EA00_{16}$ e o segundo *byte* é interpretado como o endereço destino. Como foi utilizado o endereço global ($FF_{16} = 255$), esta mensagem foi enviada para todas as ECU na rede. O PGN $EA00$ é a mensagem “*Request PGN*”. Este PGN é utilizado para pedir alguma mensagem. A mensagem solicitada é definida nos três *bytes* no campo de dados. O PGN contido no campo de dados, $00EE00_{16}$, é a mensagem “*Address Claim*”.

Tabela 7 – Pedido “educado” para configuração de endereços.

Tempo	PGN	Nome	Origem	Destino	DLC	Dados
38.9634	18EAFFFEX	<i>RequestPGN</i>	nulo	todos	3	Req. PGN: EE00
38.9642	CEEFF80X	<i>AddressClaimed</i>	80	todos	8	88 00 00 00 00 00 00 80
38.9666	18EEFF26X	<i>AddressClaimed</i>	26	todos	8	1b 00 c0 0c 00 1d 00 a0
38.9729	18EEFFF8X	<i>AddressClaimed</i>	f8	todos	8	00 00 c0 0c 00 3d 04 a0
40.2792	18EEFFF7X	<i>AddressClaimed</i>	f7	todos	8	14 00 C0 0C 00 82 00 A0

Na Tabela 7, a coluna PGN contém todos os dados do campo identificador da mensagem, e não apenas o PGN. O nome do PGN é apresentado na coluna seguinte. A coluna “Origem” refere-se ao endereço que enviou a mensagem e a coluna “Destino” ao endereço destino. A coluna “DLC” (*Data Length Code*) informa a quantidade de bytes no campo de dados, apresentado na coluna seguinte. Como pode ser verificado, três ECU se manifestaram enviando a mensagem solicitada (*AddressClaimed*) em menos de 0,01s. Após um determinado tempo, a ECU que pediu pelos endereços verificou que o endereço pretendido, $F7_{16}$, não estava sendo utilizado e por isso fez o pedido deste endereço na rede, sem disputa. Inicialmente, esta ECU utilizou o endereço nulo ($FE_{16} = 254$), pois não possuía nenhum endereço ainda. Este procedimento é vantajoso em relação a uma disputa de endereços, pois evita uma possível reconfiguração de endereço de diversas ECU na rede. O controlador que executou este procedimento foi o Controlador de Tarefas do GTA (endereço $F7_{16}$), e as outras ECU que estavam na rede eram o Terminal Virtual

(endereço 26_{16}), Servidor de Arquivos (endereço $F8_{16}$), ambos integrados no GTA, e outra ECU (proprietária) para testes, desenvolvida com a biblioteca SAE J1939.

5.2 Comunicação VT – WS

Os demais procedimentos de configuração de um WS são pedidos de informações, mensagens de estado, entre outros verificados anteriormente na Figura 18.

Figura 26 – Diagrama com sequência de mensagens de configuração de um WS com um VT.

A Figura 26 apresenta um diagrama que ilustra uma seqüência de mensagens entre o VT e o WSM. Esta série de mensagens contém os passos dois (2) a sete (7) da metodologia proposta anteriormente no capítulo 4 – Materiais e Métodos.

A seqüência de mensagens ilustrada na Figura 26 pode ser vista na Tabela 8. Esta tabela de mensagens é parcial e contém edições nos campos “Nome” e “Dados”, e foi gerada no *software* CANoe.ISO11783. A tabela integral e original é apresentada no apêndice B.

Após o pedido para carregar uma determinada versão, o VT responde se a possui ou não. No exemplo mostrado na Tabela 8, o OP foi apagado propositalmente do VT anteriormente. Assim, o VT responde que desconhece a versão. O cabeçalho de cada coluna é semelhante ao da Tabela 7.

Tabela 8 – Troca de mensagens entre WSM e VT, no procedimento de configuração do WS.

Tempo	PGN	Mensagem	origem	Dest	DLC	Dados
35.4877	1CE6FF26X	<i>VTtoECU – VT Status Message</i>	26	todos	8	FE FF FF FF FF FF 00 00
35.7666	1CFE0D80X	<i>WorkingSetMaster</i>	80	--	8	01 FF FF FF FF FF FF FF
35.7676	1CEA2680X	<i>RequestPGN</i>	80	26	3	Req. PGN: Fe0f (LanguageCommand)
35.7683	1CE72680X	<i>ECUtoVT – WS Maintenance Message</i>	80	26	8	FF FF FF FF FF FF FF FF
35.7689	1CE72680X	<i>ECUtoVT – Get Hardware</i>	80	26	8	C7 FF FF FF FF FF FF FF
35.7767	1CFE0F26X	<i>LanguageCommand</i>	26	--	8	65 6E 50 03 5A 56 FF FF
35.9000	1CE68026X	<i>VTtoECU – Get Hardware Response</i>	26	80	8	C7 FF 02 09 E0 01 E0 01
35.9009	1CE72680X	<i>ECUtoVT – Load Version</i>	80	26	8	D1 45 4E 41 4C 54 41 31
35.9076	1CE68026X	<i>VTtoECU – Load Version Response</i>	26	80	8	D1 FF FF FF FF FF 02 FF
35.9085	1CE72680X	<i>ECUtoVT – Get Memory</i>	80	26	8	C0 FF 5D 04 00 00 FF FF
35.9213	1CE68026X	<i>VTtoECU – Get Memory Response</i>	26	80	8	C0 02 00 FF FF FF FF FF

Neste exemplo, o WSM deverá enviar o OP. Opcionalmente, o WSM pode pedir informações sobre memória disponível antes de enviar o OP, como ilustra a Figura 27.

Figura 27 – Diagrama com sequência de mensagens para envio do OP.

A transferência do OP, o oitavo e último passo da sequência de configuração de um WS, foi separada em duas partes. Primeiro foi necessário desenvolver um *software* para construir um OP para posteriormente desenvolver o código para realizar o protocolo de transporte. Estes dois processos serão apresentados nos capítulos seguintes.

5.3 Construindo um OP

Como visto anteriormente no capítulo 4 – Materiais e Métodos, o anexo B na parte 6 da norma ISOBUS apresenta em detalhes todos os objetos definidos e os respectivos atributos. Apesar de existir um *software* no mercado que cria arquivos OP, chamado *Mask Generator* (WTK, 2007), apenas a versão demonstrativa deste *software* estava disponível (Figura 28).

Figura 28 – Tela parcial do programa *Mask Generator* (WTK, 2007)

Todavia, esta versão foi útil para entender a relação dos objetos entre si, o posicionamento e tamanho dos objetos (medidos em *pixels*), e para testar os arquivos de OP criados em um *software* desenvolvido neste trabalho. Apesar da versão demonstrativa não salvar arquivos OP, o *software Mask Generator* abre arquivos com a extensão apropriada. Desta maneira, ele foi utilizado para “calibração” do *software* desenvolvido neste trabalho, ou seja, os arquivos gerados neste *software* foram testados naquele.

O *software* LabView foi utilizado para desenvolver outro *software* (Figura 29) para criar OP. Este *software* não possui interface gráfica favorável, ou seja, são apenas entradas de dados (objetos e seus atributos). Os objetos são concatenados

em um conjunto de *bytes*, como mostrado anteriormente na Figura 16, e ao finalizar é gerado um arquivo padronizado segundo a norma ISOBUS.

Figura 29 – Tela parcial do programa desenvolvido para gerar o arquivo OP.

Dentre os trinta e um (31) objetos definidos na parte 6 da norma ISOBUS, o objeto mais trabalhoso para a implementação foi o *Picture graphic bitmap* (figura/imagem). O motivo foi a diferença de encapsulamento no formato de figura utilizado pela norma, sendo este diferente do formato de *bitmap* ordinário com extensão *bmp*. Em arquivos “.bmp”, o total de *bits* em cada linha é ajustado para um número múltiplo de 32 *bits*, imediatamente maior ou igual (BMP, 2006). Por exemplo, uma figura com 65 pixels de largura, com resolução de 16 cores (4 *bits* por pixel) possui 260 (65 *pixels* vezes 4 *bits*) *bits* em cada linha. Ao encapsular, as linhas ficam com 288 *bits* (9 vezes 32 *bits*). A norma ISOBUS utiliza um encapsulamento múltiplo de 8 *bits*, para reduzir o tamanho do OP (ISO 11783, 2004). Neste caso, o exemplo

anterior teria apenas 264 *bits* por linha (33 vezes 8 *bits*). Com a diferença de alguns *bits* por linha, a imagem fica distorcida, ou ligeiramente inclinada, como na Figura 30. A diferença de encapsulamento foi resolvida no *software* desenvolvido neste trabalho.

Figura 30 – Figuras ilustrando um implemento agrícola. Figura no formato “.bmp” (a) e a mesma figura interpretada segundo a norma ISOBUS.

O tamanho de OP pode variar de algumas centenas de *bytes*, por exemplo, 500 *bytes* até mais de 1Mb (1.000.000 *bytes*). O tamanho do OP não é diretamente relacionado com a complexidade do mesmo. É recomendável a utilização de OP reduzido por vários motivos. Primeiro, ele deve estar contido em memória não volátil de uma ECU. Segundo, o VT deve possuir memória suficiente para recebê-lo. Terceiro, este deve ser enviado através de um TP, ou ETP, nos quais cada pacote transfere apenas sete *bytes* por vez, causando um tráfego intenso de mensagens no barramento durante a transmissão.

O fator significativo no aumento do tamanho do OP é a utilização de desenhos e figuras, que estão no formato “*bitmap*”. Em geral, figuras facilitam a interação com o operador, que identifica um comando ou ferramenta mais rapidamente que textos. Para lidar com esta situação, deve-se reduzir o tamanho das figuras, otimizando a área útil do desenho, exemplificado nas Figura 31 (a) e (b).

Figura 31 – Quatro imagens com diferentes tamanhos e resoluções. Em (a), tamanho normal e resolução de 256 cores. Em (b), tamanho reduzido e resolução de 256 cores. Em (c), tamanho reduzido e resolução de 16 cores. Em (d), tamanho reduzido e monocromático.

Outra forma de diminuir o tamanho do OP é utilizar figuras com resolução baixa. As resoluções determinadas pela norma são: monocromático, 16 cores e 256 cores. A Figura 31 (c) e (d) mostram a diferença da mesma imagem com resoluções diferentes.

A Tabela 9 mostra em números a diferença de tamanho em *bytes* entre as quatro imagens mostrada na Figura 31. Como pode ser verificada, a redução da Figura 31 (a) para a Figura 31 (d) é superior a dez vezes em tamanho de *bytes*, sem comprometer a compreensão do operador ao selecionar este implemento no VT. O tamanho dos outros tipos de objetos varia em torno de dez (10) a quarenta (40) *bytes*. Comparando com uma figura *bitmap* este número é consideravelmente menor.

Tabela 9 – Comparação do tamanho (*bytes*) de figuras com diferentes tamanhos e resoluções.

Figura 31	dimensão	área	resolução	tamanho (<i>bytes</i>)
(a)	216 x 64	13824 <i>pixels</i>	256 cores	13824
(b)	187 x 48	8976 <i>pixels</i>	256 cores	8976
(c)	187 x 48	8976 <i>pixels</i>	16 cores	4488
(d)	187 x 48	8976 <i>pixels</i>	monocromático	1122

Os OP criados no desenvolvimento deste trabalho serão apresentados posteriormente, nos capítulos 5.5.1 e 5.5.2.

5.4 Transmissão do OP utilizando o TP

A norma ISOBUS define três protocolos de transporte: TP, ETP e BAM (*Broadcast Announce Message*, ou mensagem com destino global), . Eles têm como objetivo gerenciar a transmissão de mensagens com tamanho superior a 8 *bytes* a destinos global ou específico. Em um protocolo de transporte, a mensagem original é dividida em pacotes de 7 *bytes*, os quais são numerados e enviados como mensagens simples segundo os comandos do gerenciamento da conexão. Ao chegar ao destino, os pacotes são remontados na sequência correta, e então a mensagem original pode ser interpretada. As principais características dos três protocolos de transporte são mostradas na Tabela 10.

Tabela 10 – Características dos Protocolos de transporte.

Protocolo de transporte	Quantidade de dados enviados (<i>bytes</i>)	Destino
TP	9 a 1.785	específico
ETP	1.786 a 117.440.512	específico
BAM	9 a 1.785	global

Neste trabalho foi utilizado o TP, pela quantidade baixa de dados a ser transmitido e pelo tipo de destino.

São reservados dois PGN para a utilização do TP. O primeiro, *Transport Protocol – Connection Management* (TP.CM), é utilizado para gerenciar a conexão, por exemplo, iniciar, pedir por pacotes, validar ou abortar uma transferência. O segundo, *Transport Protocol – Data Transfer* (TP.DT), é utilizado para transferir os pacotes de dados.

Alguns PGN utilizam um *byte* de controle, ou *control byte*, (geralmente o primeiro *byte* do campo de dados) para multiplexar diferentes funções no mesmo PGN. O TP.CM é um deles, e suas diferentes funções são mostradas na Tabela 11.

Tabela 11 – Funções do PGN TP.CM (ISO 11783, 1998)

PGN	Byte de controle D0	Função	Dados D1-D7
DP = 0 PDUF = 236 PDUS = DA PGN = 60416 (00EC00 ₁₆)	16	<i>Request to Send</i> (RTS, pedido de envio) (TP.CM_RTS)	# de <i>bytes</i> , # de pacotes, PGN associado.
	17	<i>Clear to Send</i> (CTS, disponível para enviar) (TP.CM_CTS)	# pacotes a serem enviados, # primeiro pacote, PGN associado.
	19	<i>End of Message Acknowledgement</i> (EOM, confirmação do fim da mensagem) (TP.CM_EndofMsgACK)	# de <i>bytes</i> , # de pacotes, PGN associado.
	255	Abort (TP.Conn_Abort, cancelar)	PGN associado
	32	Protocolo BAM (TP.CM_BAM)	# de <i>bytes</i> , # de pacotes, PGN associado.

A desvantagem da utilização de um *byte* de controle é a diminuição do campo de dados, de 8 para 7 *bytes*, pois um deles é utilizado para o controle das funções. Porém, a vantagem é mais significativa, uma vez que é possível utilizar até 256 funções diferentes associados a um único PGN. Como pode ser observado na Tabela 11, cada função utiliza os sete *bytes* de dados de uma maneira diferente, conveniente aos requisitos relativos à função. Ou seja, ao identificar um PGN que possui um *byte* de controle, uma ECU necessita identificar este *byte* para interpretar os dados restantes.

O TP.DT não possui um *byte* de controle. Por sua vez, o primeiro *byte* (D0) representa o número do pacote que está sendo enviado. Deste modo, restam 7 *bytes* no campo de dados, que representam o pacote enviado.

A Figura 32 apresenta um fluxograma que ilustra uma seqüência de mensagens para a realização de uma transferência de dados utilizando o TP.

Figura 32 – Diagrama com seqüência de mensagens do protocolo de transporte (TP).

Na Tabela 12 um exemplo deste procedimento pode ser verificado. Analogamente a Tabela 8, esta tabela é parcial e contém edições. A tabela integral e original é apresentada no anexo B. O cabeçalho de cada coluna é semelhante ao da Tabela 7.

Tabela 12 – Troca de mensagens entre WSM e VT, no procedimento de envio do OP.

Tempo	PGN	Nome	Origem	Dest	DLC	Dados
35.9223	18EC2680X	<i>TP_CM – Request to Send</i>	80	26	8	RTS PGN: E700 Size: 1118 Packets: 160
35.9397	18EC8026X	<i>TP_CM – Clear to Send</i>	26	80	8	CTS PGN: E700 Next: 1 Packets: 16
35.9404	18EB2680X	<i>TP_DT</i>	80	26	8	SEQ.: 1 11 00 00 00 01 01 E8
35.9411	18EB2680X	<i>TP_DT</i>	80	26	8	SEQ.: 2 03 01 00 01 F9 2A 05
35.9416	18EB2680X	<i>TP_DT</i>	80	26	8	SEQ.: 3 00 14 00 45 4E E8 03
35.9423	18EB2680X	<i>TP_DT</i>	80	26	8	SEQ.: 4 01 01 A0 0F 03 00 B2
35.9428	18EB2680X	<i>TP_DT</i>	80	26	8	SEQ.: 5 36 0A 00 06 00 B2 36
35.9435	18EB2680X	<i>TP_DT</i>	80	26	8	SEQ.: 6 0A 00 50 00 04 2B 0A
35.9441	18EB2680X	<i>TP_DT</i>	80	26	8	SEQ.: 7 00 12 00 E9 03 01 01
35.9447	18EB2680X	<i>TP_DT</i>	80	26	8	SEQ.: 8 A1 0F 03 00 B8 0B 19
35.9453	18EB2680X	<i>TP_DT</i>	80	26	8	SEQ.: 9 00 0A 00 B9 0B 19 00
35.9459	18EB2680X	<i>TP_DT</i>	80	26	8	SEQ.: 10 44 00 BD 0B 19 00 C8
35.9465	18EB2680X	<i>TP_DT</i>	80	26	8	SEQ.: 11 00 EA 03 01 01 A2 0F
35.9472	18EB2680X	<i>TP_DT</i>	80	26	8	SEQ.: 12 01 00 F8 2A 00 00 00
35.9477	18EB2680X	<i>TP_DT</i>	80	26	8	SEQ.: 13 00 B8 0B 03 B4 00 3C
35.9484	18EB2680X	<i>TP_DT</i>	80	26	8	SEQ.: 14 00 00 04 00 FD 2A 0C
35.9490	18EB2680X	<i>TP_DT</i>	80	26	8	SEQ.: 15 00 06 00 E0 2E 12 00
35.9496	18EB2680X	<i>TP_DT</i>	80	26	8	SEQ.: 16 1E 00 B0 36 00 00 00
36.1340	18EC8026X	<i>TP_CM – Clear to Send</i>	26	80	8	CTS PGN: e700 Next: 17 Packets: 16
36.1349	18EB2680X	<i>TP_DT</i>	80	26	8	SEQ.: 17 00 FE 2A 6E 00 23 00
36.1356	18EB2680X	<i>TP_DT</i>	80	26	8	SEQ.: 18 B9 0B 03 B4 00 3C 00
36.1362	18EB2680X	<i>TP_DT</i>	80	26	8	SEQ.: 19 00 05 00 B0 36 00 00
---	---	---	---	---	---	---
36.3070	18EB2680X	<i>TP_DT</i>	80	26	8	SEQ.: 158 00 E9 03 FF FF FF 03
36.3076	18EB2680X	<i>TP_DT</i>	80	26	8	SEQ.: 159 00 1C 08 00 AD 00 00
36.3083	18EB2680X	<i>TP_DT</i>	80	26	8	SEQ.: 160 EA 03 FF FF FF FF FF
36.3176	18EC8026X	<i>TP_CM – End of Message ACK</i>	26	80	8	EoM ACK PGN: e700 Size: 1118 Packets: 160

Como pode ser observado na Tabela 12, o pedido foi realizado pelo endereço de origem 80₁₆ para o endereço 26₁₆ (VT). Na primeira mensagem (RTS) estão contidos os dados do TP, ou seja, o número de *bytes*, igual a 1118 *bytes*, o número de pacotes, igual a 160 pacotes, e o PGN E700, equivalente a mensagem “*ECUtoVT*”. Após 17,1 [ms], a segunda mensagem (CTS) enviada pelo VT permite o envio de apenas 16 pacotes, iniciando no pacote “1”. Imediatamente, os 16 pacotes são enviados na sequência. Após processá-los, o VT permite mais 16 pacotes,

agora iniciando no pacote “17”, e assim por diante. Ao receber o último pacote (número 160), o VT finaliza a conexão enviando a mensagem “*EndofMsgACK*”.

Os valores de intervalos (*time-out*) e o detalhamento de cada PGN podem ser verificados na parte 3 – Camada de enlace de dados. O BAM utiliza o mesmo PGN que o TP, mas os procedimentos e *time-out* são diferentes. O ETP é detalhado no anexo K, na parte 6 – Terminal Virtual da norma ISOBUS.

5.5 Estudos de caso

A implementação do padrão ISOBUS foi dividida em duas fases. A primeira foi a implementação de um protótipo rápido, no qual as funções específicas do WSM foram desenvolvidas em LabView. A aplicação do protótipo foi o controle de um Módulo de Guiagem e Propulsão (MGP), componente do Robô Agrícola Móvel (RAM). O RAM (PORTO; SOUSA; INAMASU, 2003) é um robô desenvolvido em um projeto no Laboratório de Simulação e Controle, EESC/USP, mostrado na Figura 33 (a).

Figura 33 – Robô Móvel Agrícola (a) e Aplicador de Calcário (b)

A segunda fase foi um projeto mais detalhado, com a implementação de rotinas em C para otimizar as funções, concentrando toda a programação integrada em um microcontrolador para otimizar o desempenho final. Neste caso, o controlador para estudo de caso foi um aplicador de calcário, mostrado na Figura 33 (b).

5.5.1 Implementação de um controlador ISOBUS compatível para o controle de um Módulo de Guiagem e Propulsão (MGP)

O objetivo do desenvolvimento do protótipo rápido foi criar um WS capaz de se comunicar com um Terminal Virtual de mercado. Esta comunicação reúne tanto a configuração do WS com o VT quanto a capacidade de interação do usuário com o respectivo WS através do VT.

O MGP foi utilizado no desenvolvimento do protótipo rápido. O MGP possui dois motores, um para direção e outro para girar a roda. Não houve critérios de seleção para a escolha deste módulo. A vantagem foi utilizar o controlador do MGP com comunicação CAN sem alterar sua programação, pois este foi designado como *working set member*. O objetivo principal foi implementar a configuração do WSM com o VT.

Os materiais utilizados neste desenvolvimento foram um Terminal Virtual comercial, um hardware programável com interface CAN, um controlador com aplicação na área agrícola com interface CAN (MGP) e um barramento CAN. Os *softwares* foram: Labview, MPLab , CANoe e *Mask Generator*. Outros materiais foram as normas ISO11783 e SAE J1939. Na época do desenvolvimento do primeiro

protótipo, de setembro de 2006 a fevereiro de 2007, não foi possível adquirir um Terminal Virtual de mercado. Estes terminais não estavam disponíveis no Brasil. No lugar do Terminal Virtual, foi utilizado o *software* CANoe.ISO11783, conectado ao barramento CAN e que simulou um Terminal Virtual. A Figura 34 ilustra o barramento montado para o desenvolvimento do protótipo rápido.

Figura 34 – Diagrama ilustrando a montagem do barramento para o protótipo rápido.

O arquivo OP foi criado em um programa desenvolvido em Labview, como visto no capítulo 5.3 – Construindo um OP. O tamanho do arquivo foi de 1480 *bytes*, o que possibilitou a utilização do TP para o envio ao VT. A máscara principal deste OP é mostrada na Figura 35.

Figura 35 – Máscara de dados do OP criado para o protótipo rápido.

A comunicação do MGP com o barramento foi direta, uma vez que este módulo possui interface CAN. O *software* CANoe.ISO11783, como já citado neste capítulo, substituiu o Terminal Virtual no barramento, realizando os procedimentos de configuração com o protótipo.

A interface serial CAN, desenvolvida por Sousa (2002), foi integrada no módulo que representou o WSM. As mensagens recebidas pelo CAN eram enviadas para um PC via serial, e processadas em um programa desenvolvido em Labview. As mensagens enviadas pelo mesmo programa via serial foram enviadas no barramento CAN, para a comunicação com as outras ECU na rede.

A interface serial – CAN não teve apenas o papel de receber e repassar mensagens em diferentes protocolos. O microcontrolador foi responsável pelo procedimento de configuração de endereço.

A programação do WSM, em LabView, foi organizada em dezessete (17) funções, listadas na Tabela 13.

Tabela 13 – Funções desenvolvidas para WSM.

LOGO	FUNÇÃO	DESCRIÇÃO
Recebe mensagem CAN	Verifica dados que chegam à porta serial. Em caso de nova mensagem, armazena todos os parâmetros em um cluster.
Envia mensagem CAN	Recebe parâmetros (campos identificador e dados) e envia pela porta serial, caso a mesma estiver livre (<i>time_out</i> serial).
Time_out Serial	Período de 12 ms para que uma mensagem seja enviada. Após este tempo, uma nova mensagem pode ser enviada.
Manutenção	Configura parâmetros para a mensagem <i>Working Set Maintenance</i> , que deve ser enviada uma vez por segundo.
TP	Gerencia todo o protocolo de transporte, neste caso, somente para enviar o OP.
Memória	Configura parâmetros para verificar se VT possui memória suficiente para receber o OP.
Fim do OP.	Configura parâmetros para informar o fim do OP.
Pacotes	Verifica a mensagem recebida TP.CM_CTS e configura parâmetros para o envio de pacotes.
Inicia TP.CM	Configura parâmetros para iniciar um TP, ou seja, a mensagem TP.CM_RTS.
TP.CM	Configura parâmetros para qualquer mensagem TP.CM.
Gerencia TP.DT	Gerencia a montagem dos pacotes para posterior envio.
TP.DT	Configura parâmetros para enviar pacotes.
Mudar valor	Configura parâmetros para mudar valor de variável no OP.
Mudar atributo	Configura parâmetros para mudar atributo de variável no OP.
Controle do MGP	Configura parâmetros para enviar uma mensagem de controle para o MGP.
Aumenta velocidade	Configura parâmetros para aumentar a velocidade do MGP.
Diminui velocidade	Configura parâmetros para diminuir a velocidade do MGP.

No início do programa, são atribuídos valores iniciais para as variáveis e valores da taxa de comunicação da porta serial. Após a configuração inicial, o programa executa ciclicamente o procedimento a seguir:

- Primeiro passo: verificação da chegada de novas mensagens pela interface CAN-USART, identificação de parâmetros e configuração de indicadores, se for necessária realizada alguma ação em resposta;

- Segundo passo: gerenciamento da mensagem “WS Maintenance”;

- Terceiro passo: gerenciamento do envio do OP;

- Quarto passo: atualizar dados no OP;

- Quinto Passo: enviar comando para o MGP;

Todos os passos são realizados em seqüência. Quando uma função configura parâmetros para enviar uma mensagem e a função “*time_out serial*” não permite o envio, ou seja, o intervalo de 12 ms não transcorreu após o envio da mensagem anterior, aquela função é chamada a cada iteração, até que o *time_out* ocorra e a mensagem possa ser enviada.

A cada mensagem enviada, o *time_out* é iniciado para o intervalo de 12 ms, ou seja, a eficiência da comunicação é muito baixa, considerando que uma mensagem enviada por um controlador CAN decorre em torno de 1 ms (GODOY, 2007). Este fato, devido à falta de uma comunicação direta com o CAN, tornou inviável a continuidade da implementação de mais rotinas para um WSM em Labview.

Este protótipo foi apresentado no primeiro Workshop ISOBUS Brasil (Figura 36). Considerando o protótipo rápido uma forma de experimentar a comunicação WSM – VT e outros procedimentos de uma rede ISOBUS, os objetivos foram alcançados com sucesso.

Figura 36 – Protótipo rápido montado em bancada no primeiro workshop ISOBUS Brasil (FTI, 2007). PC com programação do WSM (1), Interface Serial-CAN (2), controlador do MGP (3), MGP (4).

Na Figura 36 é apresentada a montagem parcial do protótipo rápido. O barramento CAN é composto por diversos módulos, nos quais um deles (item 2, Figura 36) é utilizado para a interface com o PC (item 1, Figura 36) que contém a programação do WSM. O controlador do MGP (item 3, Figura 36) está conectado ao barramento CAN e ao MGP (item 2, Figura 36). O PC com o *software* CANoe não aparece nesta figura.

A experiência do primeiro protótipo influenciou diretamente na realização do segundo protótipo, apresentado no capítulo seguinte.

5.5.2 Implementação de um controlador ISOBUS compatível para o controle de um aplicador de calcário

O segundo protótipo foi um projeto em parceria com empresas privadas, financiado pela FINEP. (FINEP n.º 3498/04. “Distribuidor de Insumo Localizado” Edital/Contrato MCT-RBT/FINEP/EMBRAPA CTAGRO 01/2004).

O objetivo deste protótipo foi criar um WS capaz de controlar uma aplicação agrícola no campo, neste caso em particular, controlar uma aplicação de calcário.

Os materiais necessários para este desenvolvimento foram: um trator com Terminal Virtual comercial e com a ECU do Trator, um controlador com aplicação na área agrícola com interface CAN e um barramento CAN. Os *softwares* foram: *Mask Generator*, e plataformas de desenvolvimento MPLab e CANoe. Também foram utilizados os padrões ISO 11783 e SAE J1939. Na época do desenvolvimento do segundo protótipo, com início em março de 2007, não existiam tratores ISOBUS compatíveis no Brasil. Portanto, a solução foi adaptar um trator comum com um Terminal Virtual e uma ECU do Trator. A empresa AGCO disponibilizou um Terminal Virtual, modelo GTA, para a EESC/USP, para fins de desenvolvimento relacionados ao padrão ISOBUS. Este Terminal Virtual, com entrada para GPS, foi utilizado na adaptação do trator. Adaptando a falta de uma ECU do Trator para os testes, necessária para aquisição da velocidade do trator, foi utilizada uma ECU genérica com interface CAN e simular a ECU do Trator, disponibilizando o dado da velocidade, obtido de um sensor de velocidade em relação ao solo, Radar III (DICKY-JOHN, 2007). A ATB Baldan (2007) disponibilizou o aplicador de calcário DMP-7500. Na máquina foi instalado um motor hidráulico controlado por uma válvula

solenóide. A válvula, por sua vez foi controlada por um circuito PWM (*Pulse Width Modulation*, Modulação por Largura de Pulso, neste caso utilizado para variar a potência do motor), realimentado por um sensor indutivo, que informa a rotação do eixo acionado pelo mesmo motor. A eletrônica e rotina de controle da válvula foi desenvolvida pela Enalta (2007) descrita com maior detalhe a seguir.

A Figura 37 ilustra o barramento montado para o desenvolvimento do segundo protótipo.

Figura 37 – Diagrama ilustrando a montagem do barramento para o segundo protótipo.

A interface gráfica para representar o controlador de calcário foi desenvolvida da mesma maneira que o primeiro protótipo. O OP ficou com 1117 *bytes*, e utilizou-se o TP para seu envio. Na tela de dados e controle (Figura 38 (b)) é possível indicar

a taxa de aplicação do calcário desejada, e verificar a taxa real de aplicação durante a execução da tarefa.

Figura 38 – Máscara de dados do OP desenvolvido para o controlador de calcário. Em (a), tela inicial. Em (b), tela de dados e controle.

O terminal GTA possui outras funcionalidades, tais como Controlador de Tarefas e GPS. Porém, estes controladores não foram utilizados nestes testes.

O aplicador de calcário foi desenvolvido pela empresa ATB Baldan. O módulo eletrônico que controla a taxa de aplicação, desenvolvido pela empresa Enalta, é disponibilizado com o implemento agrícola, sem compatibilidade ISOBUS. Este protótipo (Figura 39, item 1) substituiu o módulo eletrônico original nos testes. Ele interage com a válvula (Figura 39, item 2) reguladora de fluxo, controlada por um sinal PWM. A válvula aciona um motor hidráulico (Figura 39, item 3). O motor gira o eixo principal (Figura 39, item 4) através de uma corrente. A rotação do eixo principal é verificada por um sensor indutivo (Figura 39, item 5), que realimenta o sistema que controla a aplicação de calcário.

Figura 39 – Montagem do controlador de calcário. (1) ECU do WSM, (2) válvula com controle de fluxo em função do sinal PWM, (3) motor hidráulico, (4) eixo principal, (5) sensor indutivo para leitura da rotação do eixo principal, (6) Radar III.

O *software* CANoe foi utilizado apenas para registrar as mensagens do barramento durante o desenvolvimento e testes.

A velocidade do trator foi obtida utilizando-se o Radar III (Figura 40). O sensor foi acoplado no implemento agrícola (Figura 39, item 6) pois ocorreram problemas de leitura quando instalado no trator, por motivo de vibração da estrutura. Através da ECU genérica (SOUSA, 2002), foi possível simular a ECU do Trator para disponibilizar a velocidade do trator para o implemento agrícola.

Figura 40 – Sensor Radar III (DICKEY-JOHN, 2007).

A principal diferença entre o primeiro e o segundo protótipo foi a linguagem de programação. A programação deste foi desenvolvida no *software* MPLab, em linguagem C. O algoritmo desenvolvido é baseado na seqüência de procedimentos de configuração de um WS, cuja metodologia foi apresentada no capítulo 4 e os resultados de cada processo nos capítulos 5.1 a 5.4. Além do procedimento de configuração de um WS, foi implementado o controle de aplicação do calcário. Este controle atuou em um motor hidráulico, com um sinal PWM. O controle foi do tipo proporcional e em função da velocidade do trator e da taxa de aplicação nominal que o operador escolheu no VT, durante a aplicação.

Este protótipo foi testado em campo, nas dependências da empresa ATB Baldan, em Matão, em agosto de 2007. O controlador de calcário apresentou funcionamento adequado quanto à inicialização do WS, a comunicação com o Terminal Virtual e a aplicação agrícola. O controlador de calcário ISOBUS demonstrou uma aplicação real da norma ISOBUS em uma máquina agrícola tracionada por trator.

O gráfico da Figura 41 apresenta grandezas medidas durante a aplicação de calcário realizada pelo protótipo em questão. Os dados foram retirados das trocas de mensagens durante a aplicação, enquanto as ECU se comunicavam entre si, para realizar a tarefa, e posteriormente tratadas e plotadas neste gráfico.

Figura 41 – Dados registrados no teste do segundo protótipo.

A duração desta amostra foi de aproximadamente 55 segundos. A velocidade do conjunto trator implemento foi de aproximadamente 9 km/h (linha laranja, Figura 41). Cada amostra foi dividida em três taxas nominais de aplicação, 1000 (mil), 1500 (mil e quinhentos) e 2000 (dois mil) kg/ha (linha azul claro, Figura 41), escolhida pelo operador. A empresa Enalta disponibilizou a relação entre velocidade, taxa nominal e rotação do eixo do aplicador de calcário. Desta maneira, o controlador de calcário (WSM) verifica a velocidade (proveniente do simulador de TECU), e a taxa nominal (escolhida no VT) e calcula a rotação nominal, em RPM, naquele instante (linha vermelha, Figura 41). O sensor da rotação localizado no aplicador de calcário realimenta o sistema, com a informação da rotação real do eixo de aplicação (linha verde, Figura 41). A diferença entre as rotações real e nominal são entradas para um controlador proporcional simples, que produz um valor de PWM (linha lilás, Figura 41), que é enviado ao motor hidráulico do aplicador. O valor do PWM possui

resolução de 10 *bits* (valores de 0 a 1023). Os valores de rotação nominal, rotação real e PWM são internos ao controlador de calcário, e foram enviados como mensagens proprietárias na rede somente para análise de dados posterior. O valor da taxa de aplicação real foi enviado ao VT para que o operador pudesse verificá-lo (linha azul escuro, Figura 41).

Como percebido na Figura 41, a curva de cor azul claro indica a mudança de taxa de aplicação pelo operador no VT ao longo do tempo. No instante de cada mudança, o controlador corrige a taxa de aplicação real aumentando o PWM, verificado na curva de cor lilás, ou seja, aumentando a potência do motor hidráulico que controla a aplicação.

6. CONCLUSÃO

A implementação dos dois protótipos ISOBUS corrobora com o potencial de aplicação esperado deste padrão na área agrícola. A experiência acumulada neste trabalho pode contribuir no sentido de auxiliar a promoção e apoiar projetos de controladores ISOBUS compatíveis no Brasil.

O segundo protótipo montado no aplicador de calcário realizou a configuração com o sistema do trator adaptado com uma rede ISOBUS. Este controlador de calcário revelou-se um equipamento ISOBUS compatível. Da mesma forma, este controlador poderia ser testado com outros tratores ISOBUS. Esta possibilidade verifica a padronização do implemento agrícola, que pode ser utilizado em conjunto com diferentes modelos de tratores ISOBUS.

No teste, o controlador de calcário transmitiu via rede os dados do OP (interface gráfica) para a representação daquele no Terminal Virtual existente no trator. A utilização do Terminal Virtual existente no trator dispensa a necessidade de uma IHM proprietária, simplifica o projeto de hardware de um implemento agrícola e portanto reduz custo.

No teste, o controlador de calcário também dispensou um dispositivo para a medição da velocidade. Este parâmetro e muitos outros, como posicionamento global, posição do engate, PTO, entre outros que são especificados na norma e fornecidos pelo trator, verificam a conveniência do sistema, simplificando ainda mais o projeto de um implemento agrícola.

O controlador de calcário demonstrou os benefícios da implementação da norma ISOBUS. A padronização de conectores confirma a possibilidade de conexão

de dispositivos de diferentes fabricantes. A padronização da comunicação permitiu a utilização de um barramento único, possibilitou a configuração para diversos tipos de aplicação agrícola e reduziu custo com chicotes elétricos. No entanto, a maior demonstração foi a possibilidade de redução de dispositivos do fabricante de implementos agrícolas, tais como uma IHM na cabine do trator e dispositivos de leitura de parâmetros do trator, como posição, velocidade, parâmetros de engate e de PTO.

Porém, há fatores que devem ser levados em consideração para a implementação da norma ISOBUS. O custo de implementação inicial é relativamente alto para empresas de pequeno e médio porte. Custos com aquisição do padrão, com ferramentas de desenvolvimento, com testes de protótipos e com certificação devem ser considerados (SCHMIDT, 2007). Outro fator importante é o custo no investimento de profissionais, a médio e longo prazo, para a especialização neste assunto, visto que a norma ISOBUS é extensa e ainda está incompleta.

Apesar do alto custo inicial, as empresas devem considerar a tendência global, na possibilidade da demanda do mercado nacional e internacional por produtos com a certificação do padrão.

A implementação destes protótipos também demonstrou que a utilização de outros equipamentos em conformidade com a norma, de ferramentas que simulam o comportamento de tais equipamentos e de locais adequados para testes (bancada ou campo experimental) são instrumentos importantes no desenvolvimento de projetos no tema. O curso da empresa Vector também foi essencial para o desenvolvimento dos protótipos dentro do prazo exigido pelo presente trabalho. Apenas a utilização dos quatorze documentos da norma ISOBUS seria inviável para

o desenvolvimento deste projeto. Para iniciar um trabalho de implementação, dificilmente pode-se prescindir das ferramentas e apoio.

A troca de experiência realizada no Workshop ISOBUS Brasil auxiliou diretamente no desenvolvimento deste trabalho. O contato com os profissionais que estão diretamente ligados com o desenvolvimento da norma, tais como William Rudolph, Mike Schmidt, Hans Jürgen Nissen, Peter Fellmeth e Christian Büngener, e suas respectivas palestras nos trouxeram uma grande riqueza de detalhes, importante para a compreensão do estado e evolução da técnica. Isso demonstra que eventos como oficinas e *plugfest* patrocinados pelos grupos NAIITF, IGI e FTI Brasil são indispensáveis para a obtenção de informações e conseqüentemente para o desenvolvimento de projetos de equipamentos ISOBUS compatíveis.

A necessidade da realização do curso da Vector e o evento Workshop ISOBUS Brasil, que trouxeram diversos representantes de empresas internacionais citados neste trabalho, demonstrou que não há profissionais qualificados em relação a este assunto no Brasil. Pode-se concluir que o tema traz oportunidades para o Brasil equiparar a técnica e formar profissionais em comunicação de eletrônica embarcada.

Finalmente, a implementação do controlador de calcário e a verificação dos pontos positivos e negativos citados acima demonstra o potencial da aplicação deste padrão em máquinas e implementos agrícolas.

7. TRABALHOS FUTUROS

O protocolo ainda está em desenvolvimento. A implementação está no início. O processo de diagnóstico ainda está em fase de desenvolvimento e há necessidade de estudos que levem em conta as diversas características das máquinas agrícolas. No Brasil, não há ainda instituição capaz de realizar testes de compatibilidade. Portanto há oportunidades e necessidades para:

- Desenvolver metodologias de controle ativo em sistemas distribuídos como em engate de três pontos ou do controle da tomada de potência durante operações em campo;
- Estudar e implementar procedimentos que possam ser adotados por laboratórios brasileiros;
- Desenvolver ferramentas de apoio à implementação do ISOBUS, como simuladores, editores e gerenciadores de informações de operação ou tarefa agrícola;
- Estudar e desenvolver implementações em diferentes máquinas agrícolas como semeadoras e adubadoras.

A metodologia apresentada neste trabalho pode ser aplicada a outros dispositivos do trator além do Terminal Virtual. Um exemplo é a configuração de um implemento agrícola com o Controlador de Tarefas, que coordena a realização de uma determinada tarefa no campo, no envio de mensagens padronizadas ao WS e no armazenamento das informações obtidas neste processo.

REFERÊNCIAS

- AGCO. (2007). AGCO Corporation. Disponível em: <<http://www.agcocorp.com>>. Acesso em: Dez. 2006.
- AGRITECHNICA. (2007). Apresenta informações sobre a feira Agritechnica. Disponível em <<http://www.agritechnica.com/830.0.html>>. Acesso em: Nov. 2007.
- AKYILDIZ, I. F.; WANG, X.; WANG, W. (2005). Wireless mesh networks: a survey. **Computer Networks**. v. 47, Issue 4, p. 445-487, Mar. 2005.
- ALVES, M.; TOVAR, E.; VASQUES, F.; HAMMER, G.; ROTHER, K. (2002). Real-time communications over hybrid wired/wireless PROFIBUS-based networks. In: REAL-TIME SYSTEMS EUROMICRO CONFERENCE, 14, 2002. **Proceedings...** Viena, Austria: 2002, pp. 142-150.
- AMC. (2007). Agricultural Machinery Conference. Disponível em < <http://www.amc-online.org/index.html>>. Acesso em: Nov. 2007.
- AUERNHAMMER, H.; ROTHMUND, M. (2004). Automated Process Data Acquisition within standardized Communication Systems and its Practical Applications. In: OLYMPICS OF AGRICULTURAL ENGINEERING, CIGR INTERNATIONAL CONFERENCE, 2004, Beijing. Vol. 2, p III – 93.
- BALDAN. (2007). Agri-Tillage do Brasil (ATB). Disponível em: <<http://www.baldan.com.br>>. Acesso em: Nov. 2007.
- BENNEWEIS, R.K. Status of the ISO 11783 Serial control and communications data network standard. **ASAE Meeting Presentation**, Tampa Convention Center, Tampa, Florida, Jul. 2005.
- BMP. (2006). Apresenta informações sobre o formato *bitmap*. Disponível em: <<http://www.fileformat.info/format/bmp/egff.htm>>. Acesso em: Dez. 2006.
- BOSCH, R., GMBH. (1991). **CAN Specification, Version 2.0**. Postfach 30 02 40, D-70442 Stuttgart, 1991. 73 p.
- BÜNGENER, C. (2007). **Development of ISOBUS Applications based on the open-source library ISOAgLib**. Documento referente a apresentação no workshop ISOBUS Brasil. Disponível em: <<http://isobus.org.br/workshop2007/prog.php>>. Acesso em: Jul. 2007.
- CANBUS. (2007). Apresenta informações sobre o protocolo CAN. Disponível em <<http://www.canbus.com.br/>>. Acesso em: Nov. 2007.
- CANOPEN. (2007). Apresenta informações sobre o protocolo CANopen. Disponível em <<http://www.can-cia.org/index.php?id=171>>. Acesso em: Nov. 2007.
- DEVICENET. (2007). Apresenta informações sobre o protocolo DeviceNet. Disponível em <<http://www.odva.org/>>. Acesso em: Nov. 2007.

DICKEY-JOHN. (2007). Apresenta informações sobre o sensor de velocidade em relação ao solo Radar III. Disponível em <<http://www.dickey-john.com/>>. Acesso em: Set. 2007.

ENALTA. (2007). Enalta Inovações Tecnológicas. Disponível em: <<http://www.enalta.com.br>>. Acesso em Nov. 2007.

FELLMETH, P. (2003). **CAN-based tractor** – agricultural implement communication ISO 11783. CAN Newsletter September 2003. Disponível em: <<http://www.can-cia.org/newsletter/>>. Acesso em Fev. 2007.

_____. (2007). **Simulation, Test and Development of ISO11783 Systems**. Documento referente a apresentação no workshop ISOBUS Brasil. Disponível em: <<http://isobus.org.br/workshop2007/prog.php>>. Acesso em: Jul. 2007.

FLEXRAY. (2007). Apresenta informações sobre o protocolo FlexRay. Disponível em <<http://www.flexray.com/>>. Acesso em: Nov. 2007.

FTI. (2007). Força Tarefa ISOBUS Brasil. Disponível em <<http://www.isobus.org.br/>>. Acesso em: Nov. 2007.

GODOY, E. P. (2007). **Desenvolvimento de uma ferramenta de análise de desempenho de redes CAN (Controller Area Network) para aplicações em sistemas agrícolas**. Dissertação de Mestrado, Escola de Engenharia de São Carlos, Universidade de São Paulo. São Carlos, 2007.

GODOY, E. P.; SAKAI, R. M. R.; SOUSA, R. V.; PORTO, A. J. V.; INAMASU, R. Y. (2007). Development of an Analysis and Test Tool of ISO 11783 Networks for Agricultural Machinery. In: CONGRESSO BRASILEIRO DE ENGENHARIA AGRÍCOLA, 36, 2007, Bonito. **Anais**. Bonito: CONBEA, 2007.

GUIMARÃES, A. A. (2003). **Análise da norma ISO11783 e sua implementação no barramento do implemento de um monitor de semeadora**. Dissertação de Mestrado, Escola Politécnica, Universidade de São Paulo. São Paulo, 2003.

IGI. (2007). Implementation Group ISOBUS. Disponível em <<http://www.isobus.com/>>. Acesso em: Nov. 2007.

ISO. (2007). International Standardization Organization. Disponível em: <<http://www.iso.org>>. Acesso em: Abr. 2007.

ISO 11783. (1998). **ISO 11783** – Part 3: Data Link Layer. Disponível em: <<http://www.iso.org>>. Acesso em: Set. 2006.

_____. (2001). **ISO 11783** – Part 5: Network Management. Disponível em: <<http://www.iso.org>>. Acesso em: Set. 2006.

_____. (2004). **ISO 11783 – Part 6: Virtual Terminal**. Disponível em: <<http://www.iso.org>>. Acesso em: Set. 2006.

_____. (2006). **Tractors and machinery for agriculture and forestry – Serial control and communication data network**. Disponível em: <<http://www.iso.org>>. Acesso em: Set. 2006.

_____. (2007a). **ISO 11783 – Part 1: General Standard for Mobile Data Communication**. Disponível em: <<http://www.iso.org>>. Acesso em: Out. 2007.

_____. (2007b). **ISOBUS Data Dictionary**. Disponível em: <[http://www.isobus.net/isobus E/](http://www.isobus.net/isobus_E/)>. Acesso em: jul. 2007.

_____. (2007c). **ISO/DIS 11783 – Part 12: Diagnostics services**. Disponível em: <<http://www.iso.org>>. Acesso em: Out. 2007.

ISOAGLIB. (2007). Apresenta tutoriais e exemplos de aplicações com a norma ISOBUS. Disponível em: <<http://www.isoaglib.org/>>. Acesso em: Mar. 2007.

ISOBUS. (2007). Apresenta informações sobre o desenvolvimento do padrão ISOBUS. Disponível em <[http://www.isobus.net/isobus E/](http://www.isobus.net/isobus_E/)>. Acesso em: Jul. 2007.

LANDI. (2004). **Uma Proposta para Adoção de Dispositivos Computacionais Portáteis para Implementação do Terminal Virtual e do Controlador de Tarefas da Norma ISO 11783 em Redes Embarcadas em Máquinas Agrícolas**. Dissertação de Mestrado, Escola Politécnica, Universidade de São Paulo. São Paulo, 2004.

LANDTECHNIK-VEREINIGUNG. (2002). LBS: the mobile agricultural BUS. LBS Documentation, version 2.0 – 08. Frankfurt, 1998. Disponível em: <<http://isotc.issso.ch>>. Acesso em: mar. 2007.

MICROCHIP. (2007). Fornece programa MPLAB e biblioteca SAE J1939. Disponível em <<http://www.microchip.com/>>. Acesso em: Jul. 2007.

MODBUS. (2007). Apresenta informações sobre o protocolo Modbus de comunicação industrial. Disponível em <<http://www.modicon.com/techpubs/toc7.html>>. Acesso em: Nov. 2007.

MUNACK, A., SPECKMANN, H. Communication Technology Is the Backbone of Precision Agriculture. **Agricultural Engineering International: the CIGR Journal of Scientific Research and Development**. Vol. III. Mai. 2001.

NAIITF. (2007). North American ISOBUS Implementation Task Force. Disponível em: <<http://naiitf.aem.org>>. Acesso em: Nov. 2007.

NI. (2006). National Instruments. Disponível em: <<http://www.ni.com/>>. Acesso em Out. 2006.

NISSEN H. J.; (2007a). **ISOBUS Functional Overview**. Documento referente a apresentação no workshop ISOBUS Brasil. Disponível em: <<http://isobus.org.br/workshop2007/prog.php>>. Acesso em: Jul. 2007.

_____. (2007b). **Organização e Atividades da Força Tarefa Européia – IGI**. Documento referente a apresentação no workshop ISOBUS Brasil. Disponível em: <<http://isobus.org.br/workshop2007/prog.php>>. Acesso em: Jul. 2007.

OKSANEN, T.; ÖHMAN, M.; MIETTINEN, M.; VISALA, A. (2004). Open Configurable Control system for Precision Farming. In: ASAE CONFERENCE ON AUTOMATION TECHNOLOGY FOR OFF-ROAD EQUIPMENT, 2004, Kyoto. **Proceedings**. Kyoto: *ASAE Publication Number 701P1004*, Out, 2004, p. 184-191.

_____. (2005). ISO 11783 – Standard and its Implementation. In: IFAC WORLD CONGRESS, 16, 2005, Praha. **Proceedings**. 7-8 Jul, 2005, p. 6.

PLUGFEST. (2007). Apresenta informações sobre os eventos *plugfests* anteriormente realizados e que serão realizados. Disponível em <<http://www.aem.org/Technical/NAITF/Plugfests/index.asp>>. Acesso em Nov. 2007.

PORTO, A. J. V.; SOUSA, R. V.; INAMASU, R. Y. (2003). Robô Agrícola Móvel (RAM): uma revisão das pesquisas recentes sobre sistema de navegação autônoma de robôs e veículos agrícolas. In: CONGRESSO BRASILEIRO DA SOCIEDADE BRASILEIRA DE INFORMÁTICA APLICADA À AGROPECUÁRIA E AGROINDÚSTRIA – SBIAGRO, 4, 2003, Porto Seguro. **Anais**. SBIAGRO. 1 CD-ROM.

POWELL. (2007). Conectores padronizados pela norma ISOBUS. Disponível em <<http://www.powell.com/agriculture.php>>. Acesso em: Dez. 2006.

PROFIBUS. (2007). Apresenta informações sobre o protocolo Profibus de comunicação industrial. Disponível em <<http://www.profibus.com/pb/>>. Acesso em: Nov. 2007.

RUDOLPH W. W. (2007). **Atividades atuais e futuras da NAITF**. Documento referente a apresentação no workshop ISOBUS Brasil. Disponível em: <<http://isobus.org.br/workshop2007/prog.php>>. Acesso em: Jul. 2007.

SAE J1939. (2007). Introduction to SAE J1939. Disponível em <<http://www.kvaser.com/can/hlps/>>. Acesso em: Nov. 2007.

SAKAI, R. M. R.; PEREIRA, R. R. D.; SOUSA, R. V.; INAMASU, R. Y.; PORTO, A. J. V. (2007a). Revisão do Padrão Isobus para Comunicação do Implemento Agrícola com Terminal Virtual e Controlador de Tarefas. In: CONGRESSO BRASILEIRO DA SOCIEDADE BRASILEIRA DE INFORMÁTICA APLICADA À AGROPECUÁRIA E AGROINDÚSTRIA – SBIAGRO, 6, 2007, São Pedro. **Anais**. SBIAGRO. 1 CD-ROM.

SAKAI, R. M. R.; CAVANI, F. A.; SOUSA, R. V.; INAMASU, R. Y.; PORTO, A. J. V. (2007b). Softwares para Desenvolvimento de Aplicações ISOBUS. In: CONGRESSO BRASILEIRO DA SOCIEDADE BRASILEIRA DE INFORMÁTICA APLICADA À

AGROPECUÁRIA E AGROINDÚSTRIA – SBIAGRO, 6, 2007, São Pedro. **Anais. SBIAGRO.** 1 CD-ROM.

SARAIVA, A. M.; CUGNASCA. C. E. (2006). Redes de Comunicação Serial em Máquinas Agrícolas: uma Revisão. **Revista Brasileira de Agroinformática**, v. 8, n. 1, p. 17-35, 2006.

SCHMIDT, M. (2007). **Getting Started on ISOBUS** – Selecting Processors, Software, Development Tools and Suppliers. Documento referente a apresentação no workshop ISOBUS Brasil. Disponível em: <<http://isobus.org.br/workshop2007/prog.php>>. Acesso em: Jul. 2007.

SILVA, K. M. R. (2003). **Agrican:** Simulador de redes embarcadas no protocolo ISO 11783 para ambiente WEB. Dissertação de Mestrado, Escola Politécnica, Universidade de São Paulo. São Paulo, 2003.

SOUSA, R. V. (2002). **CAN (Controller Área Network):** Uma abordagem para automação e controle na área agrícola. Dissertação (Mestrado em Engenharia Mecânica). Escola de Engenharia de São Carlos, Universidade de São Paulo. São Carlos, 2002.

STONE, M. L.; MCKEE, K. D.; FORMWALT, C. W.; BENNEWEIS, R. K. (1999). ISO 11783: An Electronic Communication Protocol for Agricultural Equipment. **ASAE Publication**, Agricultural Equipment Technology Conference, Louisville, Kentucky, n. 913C1798, p. 1-17, Fev. 1999.

STRAUSS, C. (2001). **Implementação e avaliação de uma rede experimental baseada em CAN para aplicações agrícolas.** Dissertação de Mestrado, Escola Politécnica, Universidade de São Paulo. São Paulo, 2001.

TANENBAUM, A. S. (1997). **Redes de computadores.** 3.ed. Rio de Janeiro. Editora Campus.

VALTRA. (2007). Valmet Tratores. Disponível em: <<http://www.valtra.com.br>>. Acesso em: Nov. 2007.

VECTOR. (2007). Fornece o ambiente de desenvolvimento CANoe. Disponível em: <<http://www.vector-worldwide.com/>>. Acesso em Set. 2006.

XML. (2001). World Wide Web Consortium – X3C. Apresenta informações sobre o formato XML. Disponível em: <<http://www.w3.org/2001/XMLSchema>>. Acesso em: Nov. 2007.

WTK. (2007). Apresenta uma versão demonstrativa do software *Mask Generator*. Disponível em: <<http://www.wtk-elektronik.de/eindex.htm>>. Acesso em Out. 2006.

APÊNDICES

APÊNDICE A – Implementação de função em linguagem C

Este anexo apresenta a implementação de uma função (em linguagem C) na biblioteca SAE J1939, para recalcular um novo valor de endereço de uma ECU que perde a disputa por um endereço.

Função:

```

BOOL CA_RecalculateAddress ( unsigned char *CommandedAddress )
{
    if (!ISO11783_Flags.WaitingForAddressClaimContention) // Alguma ECU entrou
                                                         // com mesmo SA, e
                                                         // maior prioridade.
        *CommandedAddress = 128; // Iniciar procura por endereco disponivel
    else *CommandedAddress = *CommandedAddress +1; // Esta ECU ainda não
                                                         // encontrou SA disponivel,
                                                         // logo incrementa e tenta
                                                         // novamente.

    if (*CommandedAddress == 239) //Até agora, do 128 a 238 estão reservados.
        return 0;
    return 1;
}

```

Esta função incrementa o valor do endereço pretendido na rede (variável *CommandedAddress*). Se a ECU já possui este endereço e perdeu a disputa para outra ECU que acabou de entrar na rede, a busca se inicia pelo endereço 128, que é o primeiro valor de endereço não preferencial. Ao retornar o valor “1”, o programa chama uma função para pedir o novo endereço. Se a ECU perde o endereço novamente, o programa retorna a esta função para incrementar o valor, e assim por diante. Caso todos os endereços não preferenciais possíveis (128 a 238 inclusive) não puderem ser utilizados, a função retorna “0”, e o programa chama outra função que envia a mensagem “*Cannot Claim for Source Address*”. Neste caso, a ECU interrompe a comunicação na rede.

APÊNDICE B – Mensagens em um barramento ISOBUS

As mensagens na tabela seguinte foram registradas em um teste realizado com o protótipo (segundo estudo de caso deste trabalho). No barramento estão conectados o WSM (protótipo) e o terminal GTA, que possui as ECU Terminal Virtual, Controlador de Tarefas e Servidor de Arquivos. Neste pequeno trecho, com duração de alguns segundos, ocorre a configuração do WS (protótipo) com o Terminal Virtual.

Esta tabela indica o tempo (“Tempo”, primeira coluna) em que a mensagem foi enviada no barramento, o identificador (“PGN”, segunda coluna) da mensagem, o nome do PGN (“Nome”, terceira coluna) contido no identificador, o endereço de origem (“Src”, quarta coluna) da mensagem, o endereço destino (“Dest”, quinta coluna) da mensagem, a quantidade de bytes (“DLC”, sexta coluna) no campo de dados e os dados (“Dados” sétima coluna) contidos na mensagem.

Tempo	PGN	Nome	Src	Dest	DLC	Dados
33.046.121	1cabfff8x	FSServer	f8	all	8	00 00 00 ff ff ff ff ff
34.216.188	18eeff26x	AddressClaimed	26	all	8	1b 00 c0 0c 00 1d 00 a0
34.480.171	1cfe0f26x	LanguageCommand	26	--	8	65 6e 50 03 5a 56 ff ff
34.480.876	1ce6ff26x	VTtoECU	26	all	8	fe ff ff ff ff ff 00 00
35.082.895	1cabfff8x	FSServer	f8	all	8	00 00 00 ff ff ff ff ff
35.476.269	ceeff80x	AddressClaimed	80	all	8	88 00 00 00 00 00 00 80
35.487.749	1ce6ff26x	VTtoECU	26	all	8	fe ff ff ff ff ff 00 00
35.766.650	1cfe0d80x	WorkingSetMaster	80	--	8	01 ff ff ff ff ff ff ff
35.767.310	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
35.767.694	1cea2680x	RequestPGN	80	26	3	Req. PGN: fe0f
35.768.390	1ce72680x	ECUtoVT	80	26	8	ff ff ff ff ff ff ff ff
35.768.975	1ce72680x	ECUtoVT	80	26	8	c7 ff ff ff ff ff ff ff
35.769.651	1ce72680x	ECUtoVT	80	26	8	a8 08 52 ff 00 00 00 00
35.776.799	1cfe0f26x	LanguageCommand	26	--	8	65 6e 50 03 5a 56 ff ff
35.816.596	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
35.866.632	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
35.900.099	1ce68026x	VTtoECU	26	80	8	c7 ff 02 09 e0 01 e0 01
35.900.956	1ce72680x	ECUtoVT	80	26	8	d1 45 4e 41 4c 54 41 31
35.907.668	1ce68026x	VTtoECU	26	80	8	d1 ff ff ff ff 02 ff ff
35.908.552	1ce72680x	ECUtoVT	80	26	8	c0 ff 5d 04 00 00 ff ff
35.916.692	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
35.921.385	1ce68026x	VTtoECU	26	80	8	c0 02 00 ff ff ff ff ff
35.922.329	18ec2680x	TP_CM	80	26	8	RTS PGN: e700 Size: 1118 Packets: 160

35.939.477	18ec8026x	TP_CM	26	80	8	CTS PGN: e700 Next: 1 Packets: 16
35.940.421	18eb2680x	TP_DT	80	26	8	Seq.: 1 11 00 00 00 01 01 e8
35.941.101	18eb2680x	TP_DT	80	26	8	Seq.: 2 03 01 00 01 f9 2a 05
35.941.657	18eb2680x	TP_DT	80	26	8	Seq.: 3 00 14 00 45 4e e8 03
35.942.322	18eb2680x	TP_DT	80	26	8	Seq.: 4 01 01 a0 0f 03 00 b2
35.942.878	18eb2680x	TP_DT	80	26	8	Seq.: 5 36 0a 00 06 00 b2 36
35.943.550	18eb2680x	TP_DT	80	26	8	Seq.: 6 0a 00 50 00 04 2b 0a
35.944.110	18eb2680x	TP_DT	80	26	8	Seq.: 7 00 12 00 e9 03 01 01
35.944.771	18eb2680x	TP_DT	80	26	8	Seq.: 8 a1 0f 03 00 b8 0b 19
35.945.323	18eb2680x	TP_DT	80	26	8	Seq.: 9 00 0a 00 b9 0b 19 00
35.945.983	18eb2680x	TP_DT	80	26	8	Seq.: 10 44 00 bd 0b 19 00 c8
35.946.547	18eb2680x	TP_DT	80	26	8	Seq.: 11 00 ea 03 01 01 a2 0f
35.947.228	18eb2680x	TP_DT	80	26	8	Seq.: 12 01 00 f8 2a 00 00 00
35.947.788	18eb2680x	TP_DT	80	26	8	Seq.: 13 00 b8 0b 03 b4 00 3c
35.948.464	18eb2680x	TP_DT	80	26	8	Seq.: 14 00 00 04 00 fd 2a 0c
35.949.020	18eb2680x	TP_DT	80	26	8	Seq.: 15 00 06 00 e0 2e 12 00
35.949.697	18eb2680x	TP_DT	80	26	8	Seq.: 16 1e 00 b0 36 00 00 00
35.966.729	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
36.016.810	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
36.066.802	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
36.116.839	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
36.134.015	18ec8026x	TP_CM	26	80	8	CTS PGN: e700 Next: 17 Packets: 16
36.134.947	18eb2680x	TP_DT	80	26	8	Seq.: 17 00 fe 2a 6e 00 23 00
36.135.635	18eb2680x	TP_DT	80	26	8	Seq.: 18 b9 0b 03 b4 00 3c 00
36.136.203	18eb2680x	TP_DT	80	26	8	Seq.: 19 00 05 00 b0 36 00 00
36.136.880	18eb2680x	TP_DT	80	26	8	Seq.: 20 00 00 00 2b 0c 00 06
36.137.440	18eb2680x	TP_DT	80	26	8	Seq.: 21 00 ba 0b 14 00 20 00
36.138.104	18eb2680x	TP_DT	80	26	8	Seq.: 22 bb 0b 14 00 20 00 bc
36.138.661	18eb2680x	TP_DT	80	26	8	Seq.: 23 0b 14 00 20 00 ba 0b
36.139.341	18eb2680x	TP_DT	80	26	8	Seq.: 24 03 78 00 0f 00 01 01
36.139.913	18eb2680x	TP_DT	80	26	8	Seq.: 25 00 01 2b 00 00 00 00
36.140.581	18eb2680x	TP_DT	80	26	8	Seq.: 26 bb 0b 03 78 00 0f 00
36.141.146	18eb2680x	TP_DT	80	26	8	Seq.: 27 01 01 00 02 2b 00 00
36.141.818	18eb2680x	TP_DT	80	26	8	Seq.: 28 00 00 bc 0b 03 78 00
36.142.378	18eb2680x	TP_DT	80	26	8	Seq.: 29 0f 00 00 01 00 03 2b
36.143.054	18eb2680x	TP_DT	80	26	8	Seq.: 30 00 00 00 00 bd 0b 03
36.143.627	18eb2680x	TP_DT	80	26	8	Seq.: 31 c8 00 c8 00 00 04 00
36.144.307	18eb2680x	TP_DT	80	26	8	Seq.: 32 e1 2e 00 00 00 00 e2
36.157.823	18ec8026x	TP_CM	26	80	8	CTS PGN: e700 Next: 33 Packets: 16
36.158.759	18eb2680x	TP_DT	80	26	8	Seq.: 33 2e 00 00 1e 00 07 2b
36.159.427	18eb2680x	TP_DT	80	26	8	Seq.: 34 5a 00 05 00 08 2b 5a
36.159.987	18eb2680x	TP_DT	80	26	8	Seq.: 35 00 23 00 a0 0f 04 01
36.160.640	18eb2680x	TP_DT	80	26	8	Seq.: 36 02 00 89 13 8a 13 a1
36.161.196	18eb2680x	TP_DT	80	26	8	Seq.: 37 0f 04 01 05 00 88 13
36.161.844	18eb2680x	TP_DT	80	26	8	Seq.: 38 8b 13 8c 13 8d 13 8e
36.162.404	18eb2680x	TP_DT	80	26	8	Seq.: 39 13 a2 0f 04 01 01 00
36.163.069	18eb2680x	TP_DT	80	26	8	Seq.: 40 88 13 88 13 05 0b 01
36.163.629	18eb2680x	TP_DT	80	26	8	Seq.: 41 01 01 f9 2a 0a 00 14
36.164.289	18eb2680x	TP_DT	80	26	8	Seq.: 42 00 19 01 89 13 05 0a
36.164.849	18eb2680x	TP_DT	80	26	8	Seq.: 43 02 01 01 fa 2a 0a 00

36.165.518	18eb2680x	TP_DT	80	26	8	Seq.: 44 14 00 19 02 8a 13 05
36.166.078	18eb2680x	TP_DT	80	26	8	Seq.: 45 01 03 01 01 fb 2a 0a
36.166.746	18eb2680x	TP_DT	80	26	8	Seq.: 46 00 14 00 19 03 8b 13
36.167.306	18eb2680x	TP_DT	80	26	8	Seq.: 47 05 07 04 01 00 fc 2a
36.167.979	18eb2680x	TP_DT	80	26	8	Seq.: 48 06 00 0c 00 8c 13 05
36.168.535	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
36.179.063	18ec8026x	TP_CM	26	80	8	CTS PGN: e700 Next: 49 Packets: 16
36.179.999	18eb2680x	TP_DT	80	26	8	Seq.: 49 07 05 01 00 05 2b 1e
36.180.687	18eb2680x	TP_DT	80	26	8	Seq.: 50 00 0f 00 8d 13 05 07
36.181.252	18eb2680x	TP_DT	80	26	8	Seq.: 51 06 01 00 06 2b 1e 00
36.181.912	18eb2680x	TP_DT	80	26	8	Seq.: 52 0f 00 8e 13 05 0c 07
36.182.472	18eb2680x	TP_DT	80	26	8	Seq.: 53 01 00 ff 2a 08 00 12
36.183.152	18eb2680x	TP_DT	80	26	8	Seq.: 54 00 f8 2a 0b e0 01 e0
36.183.713	18eb2680x	TP_DT	80	26	8	Seq.: 55 01 01 da 59 01 ff ff
36.184.377	18eb2680x	TP_DT	80	26	8	Seq.: 56 01 76 00 44 65 73 65
36.184.913	18eb2680x	TP_DT	80	26	8	Seq.: 57 6e 76 6f 6c 76 69 6d
36.185.557	18eb2680x	TP_DT	80	26	8	Seq.: 58 65 6e 74 6f 0a 65 6d
36.186.106	18eb2680x	TP_DT	80	26	8	Seq.: 59 20 70 61 72 63 65 72
36.186.770	18eb2680x	TP_DT	80	26	8	Seq.: 60 69 61 3a 0a 0a 42 61
36.187.318	18eb2680x	TP_DT	80	26	8	Seq.: 61 6c 64 61 6e 0a 0a 45
36.187.978	18eb2680x	TP_DT	80	26	8	Seq.: 62 6e 61 6c 74 61 0a 0a
36.188.527	18eb2680x	TP_DT	80	26	8	Seq.: 63 46 49 4e 45 50 0a 0a
36.189.175	18eb2680x	TP_DT	80	26	8	Seq.: 64 45 6d 62 72 61 70 61
36.197.531	18ec8026x	TP_CM	26	80	8	CTS PGN: e700 Next: 65 Packets: 16
36.198.455	18eb2680x	TP_DT	80	26	8	Seq.: 65 0a 49 6e 73 74 72 75
36.199.143	18eb2680x	TP_DT	80	26	8	Seq.: 66 6d 65 6e 74 61 e7 e3
36.199.684	18eb2680x	TP_DT	80	26	8	Seq.: 67 6f 0a 41 67 72 6f 70
36.200.324	18eb2680x	TP_DT	80	26	8	Seq.: 68 65 63 75 e1 72 69 61
36.200.864	18eb2680x	TP_DT	80	26	8	Seq.: 69 0a 0a 4c 61 62 2e 20
36.201.508	18eb2680x	TP_DT	80	26	8	Seq.: 70 64 65 20 53 69 6d 75
36.202.049	18eb2680x	TP_DT	80	26	8	Seq.: 71 6c 61 e7 e3 6f 0a 45
36.202.697	18eb2680x	TP_DT	80	26	8	Seq.: 72 45 53 43 20 2d 20 55
36.203.245	18eb2680x	TP_DT	80	26	8	Seq.: 73 53 50 00 f9 2a 0b 3c
36.203.914	18eb2680x	TP_DT	80	26	8	Seq.: 74 00 10 00 01 da 59 01
36.204.474	18eb2680x	TP_DT	80	26	8	Seq.: 75 ff ff 01 04 00 43 61
36.205.134	18eb2680x	TP_DT	80	26	8	Seq.: 76 6c 63 00 fa 2a 0b 3c
36.205.694	18eb2680x	TP_DT	80	26	8	Seq.: 77 00 10 00 01 da 59 01
36.206.367	18eb2680x	TP_DT	80	26	8	Seq.: 78 ff ff 01 05 00 41 70
36.206.919	18eb2680x	TP_DT	80	26	8	Seq.: 79 6c 69 63 00 fb 2a 0b
36.207.583	18eb2680x	TP_DT	80	26	8	Seq.: 80 3c 00 10 00 01 da 59
36.217.059	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
36.217.748	18ec8026x	TP_CM	26	80	8	CTS PGN: e700 Next: 81 Packets: 16
36.218.680	18eb2680x	TP_DT	80	26	8	Seq.: 81 01 ff ff 01 04 00 49
36.219.344	18eb2680x	TP_DT	80	26	8	Seq.: 82 6e 66 6f 00 fc 2a 0b
36.219.896	18eb2680x	TP_DT	80	26	8	Seq.: 83 50 00 28 00 01 da 59
36.220.577	18eb2680x	TP_DT	80	26	8	Seq.: 84 01 ff ff 00 0c 00 41
36.221.113	18eb2680x	TP_DT	80	26	8	Seq.: 85 75 74 6f 2f 0a 4d 61
36.221.765	18eb2680x	TP_DT	80	26	8	Seq.: 86 6e 75 61 6c 00 fd 2a
36.222.317	18eb2680x	TP_DT	80	26	8	Seq.: 87 0b 96 00 0f 00 01 da
36.222.986	18eb2680x	TP_DT	80	26	8	Seq.: 88 59 01 ff ff 00 0b 00

36.223.522	18eb2680x	TP_DT	80	26	8	Seq.: 89 54 61 78 61 20 61 70
36.224.170	18eb2680x	TP_DT	80	26	8	Seq.: 90 6c 69 63 2e 00 fe 2a
36.224.722	18eb2680x	TP_DT	80	26	8	Seq.: 91 0b 3c 00 0f 00 01 da
36.225.391	18eb2680x	TP_DT	80	26	8	Seq.: 92 59 01 ff ff 00 05 00
36.225.931	18eb2680x	TP_DT	80	26	8	Seq.: 93 6b 67 2f 68 61 00 ff
36.226.595	18eb2680x	TP_DT	80	26	8	Seq.: 94 2a 0b 50 00 28 00 01
36.227.155	18eb2680x	TP_DT	80	26	8	Seq.: 95 d9 59 01 ff ff 00 04
36.227.824	18eb2680x	TP_DT	80	26	8	Seq.: 96 00 53 54 4f 50 00 00
36.238.072	18ec8026x	TP_CM	26	80	8	CTS PGN: e700 Next: 97 Packets: 16
36.238.992	18eb2680x	TP_DT	80	26	8	Seq.: 97 2b 0b 96 00 0f 00 01
36.239.684	18eb2680x	TP_DT	80	26	8	Seq.: 98 da 59 01 ff ff 00 0b
36.240.229	18eb2680x	TP_DT	80	26	8	Seq.: 99 00 4d 6f 64 6f 20 41
36.240.889	18eb2680x	TP_DT	80	26	8	Seq.: 100 70 6c 69 63 2e 00 01
36.241.449	18eb2680x	TP_DT	80	26	8	Seq.: 101 2b 0b 78 00 0f 00 01
36.242.113	18eb2680x	TP_DT	80	26	8	Seq.: 102 da 59 00 ff ff 00 0a
36.242.654	18eb2680x	TP_DT	80	26	8	Seq.: 103 00 41 75 74 6f 6d e1
36.243.306	18eb2680x	TP_DT	80	26	8	Seq.: 104 74 69 63 6f 00 02 2b
36.243.870	18eb2680x	TP_DT	80	26	8	Seq.: 105 0b 78 00 0f 00 01 da
36.244.542	18eb2680x	TP_DT	80	26	8	Seq.: 106 59 00 ff ff 01 06 00
36.245.083	18eb2680x	TP_DT	80	26	8	Seq.: 107 4d 61 6e 75 61 6c 00
36.245.751	18eb2680x	TP_DT	80	26	8	Seq.: 108 03 2b 0b 78 00 0f 00
36.246.307	18eb2680x	TP_DT	80	26	8	Seq.: 109 01 da 59 00 ff ff 01
36.246.955	18eb2680x	TP_DT	80	26	8	Seq.: 110 09 00 44 65 73 6c 69
36.247.500	18eb2680x	TP_DT	80	26	8	Seq.: 111 67 61 64 6f 00 04 2b
36.248.164	18eb2680x	TP_DT	80	26	8	Seq.: 112 0b b4 00 32 00 01 d9
36.257.564	18ec8026x	TP_CM	26	80	8	CTS PGN: e700 Next: 113 Packets: 16
36.258.508	18eb2680x	TP_DT	80	26	8	Seq.: 113 59 00 ff ff 00 17 00
36.259.164	18eb2680x	TP_DT	80	26	8	Seq.: 114 43 6f 6e 74 72 6f 6c
36.259.705	18eb2680x	TP_DT	80	26	8	Seq.: 115 61 64 6f 72 0a 64 65
36.260.361	18eb2680x	TP_DT	80	26	8	Seq.: 116 20 43 61 6c 63 e1 72
36.260.901	18eb2680x	TP_DT	80	26	8	Seq.: 117 69 6f 00 05 2b 0b 14
36.261.569	18eb2680x	TP_DT	80	26	8	Seq.: 118 00 14 00 01 d9 59 01
36.262.134	18eb2680x	TP_DT	80	26	8	Seq.: 119 ff ff 01 01 00 2b 00
36.262.802	18eb2680x	TP_DT	80	26	8	Seq.: 120 06 2b 0b 14 00 14 00
36.263.362	18eb2680x	TP_DT	80	26	8	Seq.: 121 01 d9 59 01 ff ff 01
36.264.022	18eb2680x	TP_DT	80	26	8	Seq.: 122 01 00 2d 00 07 2b 0b
36.264.575	18eb2680x	TP_DT	80	26	8	Seq.: 123 3c 00 0f 00 01 da 59
36.265.263	18eb2680x	TP_DT	80	26	8	Seq.: 124 01 ff ff 02 04 00 6b
36.265.815	18eb2680x	TP_DT	80	26	8	Seq.: 125 6d 2f 68 00 08 2b 0b
36.266.479	18eb2680x	TP_DT	80	26	8	Seq.: 126 3c 00 0f 00 01 da 59
36.267.044	18eb2680x	TP_DT	80	26	8	Seq.: 127 01 ff ff 02 03 00 52
36.267.708	18eb2680x	TP_DT	80	26	8	Seq.: 128 50 4d 00 e0 2e 0c 50
36.268.264	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
36.278.816	18ec8026x	TP_CM	26	80	8	CTS PGN: e700 Next: 129 Packets: 16
36.279.748	18eb2680x	TP_DT	80	26	8	Seq.: 129 00 14 00 01 d9 59 01
36.280.445	18eb2680x	TP_DT	80	26	8	Seq.: 130 08 52 00 00 00 00 00
36.281.021	18eb2680x	TP_DT	80	26	8	Seq.: 131 00 00 00 00 00 80 3f
36.281.685	18eb2680x	TP_DT	80	26	8	Seq.: 132 00 00 02 00 e1 2e 0c
36.282.241	18eb2680x	TP_DT	80	26	8	Seq.: 133 50 00 14 00 01 d9 59
36.282.914	18eb2680x	TP_DT	80	26	8	Seq.: 134 01 09 52 00 00 00 00

36.283.470	18eb2680x	TP_DT	80	26	8	Seq.: 135	00 00 00 00 cd cc cc
36.284.138	18eb2680x	TP_DT	80	26	8	Seq.: 136	3d 01 00 02 00 e2 2e
36.284.694	18eb2680x	TP_DT	80	26	8	Seq.: 137	0c 50 00 14 00 01 d9
36.285.363	18eb2680x	TP_DT	80	26	8	Seq.: 138	59 01 0a 52 00 00 00
36.285.943	18eb2680x	TP_DT	80	26	8	Seq.: 139	00 00 00 00 00 00 00
36.286.619	18eb2680x	TP_DT	80	26	8	Seq.: 140	80 3f 00 00 02 00 b0
36.287.167	18eb2680x	TP_DT	80	26	8	Seq.: 141	36 0e c0 5d b4 00 3c
36.287.836	18eb2680x	TP_DT	80	26	8	Seq.: 142	00 00 ff ff 00 b1 36
36.288.384	18eb2680x	TP_DT	80	26	8	Seq.: 143	0e c0 5d 4e 00 32 00
36.289.052	18eb2680x	TP_DT	80	26	8	Seq.: 144	00 ff ff 00 b2 36 0e
36.298.088	18ec8026x	TP_CM	26	80	8	CTS PGN: e700 Next: 145 Packets: 16	
36.299.012	18eb2680x	TP_DT	80	26	8	Seq.: 145	c0 5d dc 00 06 00 00
36.299.685	18eb2680x	TP_DT	80	26	8	Seq.: 146	a8 61 00 08 52 15 00
36.300.241	18eb2680x	TP_DT	80	26	8	Seq.: 147	00 00 00 09 52 15 00
36.300.909	18eb2680x	TP_DT	80	26	8	Seq.: 148	00 00 00 0a 52 15 00
36.301.461	18eb2680x	TP_DT	80	26	8	Seq.: 149	00 00 00 d8 59 17 00
36.302.126	18eb2680x	TP_DT	80	26	8	Seq.: 150	02 00 00 00 d9 59 17
36.302.686	18eb2680x	TP_DT	80	26	8	Seq.: 151	00 05 00 00 00 da 59
36.303.358	18eb2680x	TP_DT	80	26	8	Seq.: 152	17 00 03 00 00 00 c0
36.303.926	18eb2680x	TP_DT	80	26	8	Seq.: 153	5d 18 00 02 ff ff 00
36.304.587	18eb2680x	TP_DT	80	26	8	Seq.: 154	a8 61 19 02 00 ff ff
36.305.151	18eb2680x	TP_DT	80	26	8	Seq.: 155	00 01 00 1c 08 00 ad
36.305.827	18eb2680x	TP_DT	80	26	8	Seq.: 156	00 00 e8 03 ff ff ff
36.306.395	18eb2680x	TP_DT	80	26	8	Seq.: 157	02 00 1c 08 00 ad 00
36.307.068	18eb2680x	TP_DT	80	26	8	Seq.: 158	00 e9 03 ff ff ff 03
36.307.644	18eb2680x	TP_DT	80	26	8	Seq.: 159	00 1c 08 00 ad 00 00
36.308.324	18eb2680x	TP_DT	80	26	8	Seq.: 160	ea 03 ff ff ff ff ff
36.317.133	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00	
36.317.685	18ec8026x	TP_CM	26	80	8	EoM ACK PGN: e700 Size: 1118 Packets: 160	
36.318.533	18efff80x	ProprietaryA	80	all	7	01 26 02 00 08 26 a0	
36.319.229	1ce72680x	ECUtoVT	80	26	8	12 ff ff ff ff ff ff ff	
36.319.798	1ce72680x	ECUtoVT	80	26	8	a8 08 52 ff 00 00 00 00	
36.320.482	1ce72680x	ECUtoVT	80	26	8	a8 0a 52 ff 00 00 00 00	
36.367.175	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00	
36.417.207	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00	
36.467.268	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00	
36.517.345	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00	
36.518.021	1ce72680x	ECUtoVT	80	26	8	a8 08 52 ff 00 00 00 00	
36.567.361	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00	
36.617.386	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00	
36.667.423	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00	
36.717.479	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00	
36.767.560	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00	
36.768.256	1ce72680x	ECUtoVT	80	26	8	ff ff ff ff ff ff ff ff	
36.768.825	1ce72680x	ECUtoVT	80	26	8	a8 08 52 ff 00 00 00 00	
36.817.561	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00	
36.867.586	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00	
36.917.655	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00	
36.967.683	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00	
37.017.760	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00	
37.018.436	1ce72680x	ECUtoVT	80	26	8	a8 08 52 ff 00 00 00 00	

37.067.773	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
37.102.732	1ce6ff26x	VTtoECU	26	all	8	fe 80 e8 03 a0 0f 00 ff
37.104.365	1ce68026x	VTtoECU	26	80	8	a8 08 52 00 00 00 00 00
37.106.185	1ce68026x	VTtoECU	26	80	8	a8 08 52 00 00 00 00 00
37.107.781	1ce68026x	VTtoECU	26	80	8	a8 08 52 00 00 00 00 00
37.117.798	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
37.167.260	1cabfff8x	FSServer	f8	all	8	00 00 00 ff ff ff ff ff
37.167.868	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
37.217.897	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
37.267.974	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
37.268.650	1ce72680x	ECUtoVT	80	26	8	a8 08 52 ff 00 00 00 00
37.317.991	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
37.368.015	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
37.418.084	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
37.468.109	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
37.518.189	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
37.518.866	1ce72680x	ECUtoVT	80	26	8	a8 08 52 ff 00 00 00 00
37.568.202	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
37.618.227	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
37.668.295	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
37.718.324	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
37.768.401	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
37.769.097	1ce72680x	ECUtoVT	80	26	8	ff ff ff ff ff ff ff ff
37.769.665	1ce72680x	ECUtoVT	80	26	8	a8 08 52 ff 00 00 00 00
37.818.402	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
37.868.463	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
37.918.499	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
37.968.524	1cefff80x	ProprietaryA	80	all	7	00 00 00 00 00 00 00
37.989.052	1ce68026x	VTtoECU	26	80	8	12 00 ff ff ff ff 00 ff
37.989.936	1ce72680x	ECUtoVT	80	26	8	d0 45 4e 41 4c 54 41 30
37.991.428	1ce68026x	VTtoECU	26	80	8	a8 08 52 00 00 00 00 00
37.993.897	1ce68026x	VTtoECU	26	80	8	a8 08 52 00 00 00 00 00