

**VITOR LUIZ CASAGRANDE SILVA**

**ESTUDO E ELABORAÇÃO DE UMA  
REDE DE CONTROLE BASEADA NO  
PROTOCOLO CAN PARA APLICAÇÃO  
EM ELETRÔNICA EMBARCADA**

Trabalho de Conclusão de Curso apresentado à  
Escola de Engenharia de São Carlos, da  
Universidade de São Paulo

Curso de Engenharia Elétrica com ênfase em  
Eletrônica

ORIENTADOR: Prof. Dr. Luís Carlos Passarini

São Carlos  
2009

# FOLHA DE APROVAÇÃO

**Autor:** \_\_\_\_\_

**Título:** \_\_\_\_\_

**Trabalho de Conclusão de Curso defendido e aprovado**  
**em** \_\_\_\_/\_\_\_\_/\_\_\_\_,

**com NOTA** \_\_\_\_ (\_\_\_\_, \_\_\_\_), **pela comissão julgadora:**

(Assinatura) \_\_\_\_\_  
(Titulação/nome/instituição)

(Assinatura) \_\_\_\_\_  
(Titulação/nome/instituição)

\_\_\_\_\_  
Coordenador da Comissão de Coordenação do  
Curso de Engenharia Elétrica (CoC-EE)

## **DEDICATÓRIA**

Aos meus pais, Joseluiz e Eliana, por toda a dedicação, carinho, atenção e apoio durante toda a minha graduação.

## **AGRADECIMENTOS**

Ao Leandro Leite, por todo esforço e dedicação ao projeto.

Ao Eng. Me. Rodrigo Sakai e Eng. Dr. Carlos Milhor, por auxiliarem e darem suporte sempre que necessário.

Ao Prof. Dr. Luís Carlos Passarini, pela orientação neste trabalho.

À Equipe EESC – USP de fórmula SAE, por ter proporcionado esta oportunidade e muitas outras.

## RESUMO

Projeto de uma rede baseada no protocolo *Controller Area Network* (CAN), para controle e troca de dados do veículo protótipo de fórmula SAE da equipe EESC - USP. O projeto é composto por três unidades de controle eletrônico - *Electronic Control Unit* (ECU): Interface com a unidade de controle do motor - *Engine Control Module* (ECM), Painel e Aquisição de Dados. Cada ECU, ou nó, tem uma função específica, como obter dados de sensores, prover informações ao piloto ou enviar dados para um ponto remoto por telemetria. O projeto considerou análise de custo como essencial, devido ao orçamento restrito da equipe. Utilizou-se o microcontrolador *Microchip* 18F258 e protocolos recomendados pela norma SAE J1939. Realizaram-se testes e diagnósticos de tráfego de mensagens na rede em uma bancada dedicada, permitindo simular condições reais do veículo. Pôde-se assim constatar o seu funcionamento adequado. Obtiveram-se resultados promissores, possibilitando novas aplicações que ampliem o sistema de controle e troca de informações, sem alterações na topologia da rede. O projeto proporciona aumento de desempenho do veículo através de inovação tecnológica, a qual é um dos objetivos da equipe EESC - USP.

Palavras-Chave: CAN. Fórmula SAE. Sistema embarcado. Rede de controle. Eletrônica automotiva. Controle distribuído. SAE J1939.

## **ABSTRACT**

Project of a network based on Controller Area Network (CAN) protocol, for control and data exchange of the fórmula SAE vehicle prototype from EESC - USP team. The project consists of three electronic control units: Engine Control Module (ECM) Interface, Panel and Data Acquisition. Each ECU has specific functions like data acquisition from sensors, data display to the pilot and data transfer to a remote point, by telemetry. The team has a limited budget thus the project considered cost analyzes as essential. It was used Microchip 18F258 microcontroller and protocols recommended by SAE J1939 standard. It was made tests and message traffic diagnostic in the network using a dedicated bench, that simulated vehicle actual conditions. It was verified the network adequate operation. Promising results was achieved, allowing new applications that expand control system and data exchange, without modifications in the network topology. The project provides vehicle performance improvement using technological innovation, which is one of the goals of EESC – USP team.

Keys Word: CAN. Formula SAE. Automotive electronic. Control Network. Distributed ECUs. SAE J1939.

## LISTA DE FIGURAS

Figura 1. <i>Bits</i> dominantes e recessivos.....	13
Figura 2. Campos de <i>bits</i> do CAN 2.0B .....	13
Figura 3. Programação da interface em LabView .....	22
Figura 4. Interface em LabView .....	22
Figura 5. Esquemático da Interface com ECM.....	29
Figura 6. Circuito impresso do nó Interface com ECM .....	30
Figura 7. Esquemático do nó Painel .....	31
Figura 8. Circuito impresso do nó Painel.....	31
Figura 9. Esquemático do nó Aquisição de Dados.....	32
Figura 10. Circuito impresso do nó Aquisição de Dados .....	33

## LISTA DE TABELAS

Tabela 1 - Divisão do PDUF .....	16
Tabela 2 - Formação do PGN .....	16
Tabela 3 - PGNs utilizados .....	17
Tabela 4 - Definição das mensagens.....	24
Tabela 5 - Definição de NAME e ADRESS .....	28



## LISTA DE ABREVIações E SIGLAS

ACK	<i>Acknowledgment</i>
CAN	<i>Controller Area Network</i>
CSMA/CD with NDA	<i>Carrier Sense Multiple Access / Collision Detection with Non-Destructive Arbitration</i>
DA	<i>Destination Address</i>
DLC	<i>Data Length Code</i>
DP	<i>Data Page</i>
ECM	<i>Engine Control Module</i>
ECU	<i>Electronic Control Unit</i>
EESC	Escola de Engenharia de São Carlos
GE	<i>Group Extension</i>
NACK	<i>Negative-Acknowledgment</i>
PCB	<i>Printed Circuit Board</i>
PDU	<i>Protocol Data Unit</i>
PDUF	<i>PDU Format</i>
PDUS	<i>PDU Specific</i>
PG	<i>Parameter Group</i>
PGN	<i>Parameter Group Number</i>
PRI	Prioridade
R	Reservado
RPM	Rotações Por Minuto
SA	<i>Source Address</i>
SAE	<i>Society of Automotive Engineers</i>
TPS	<i>Throttle Position Sensor</i>
USP	Universidade de São Paulo

# SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>11</b>
1.1 CONTROLLER AREA NETWORK (CAN) .....	12
1.1.1 MENSAGEM CAN.....	14
1.1.2 PGN .....	15
<b>2. OBJETIVOS .....</b>	<b>18</b>
<b>3. REVISÃO BIBLIOGRÁFICA .....</b>	<b>19</b>
<b>4. MATERIAIS E MÉTODOS.....</b>	<b>21</b>
<b>6. RESULTADOS .....</b>	<b>23</b>
6.1 MENSAGENS .....	24
6.1.1 INTERFACE COM ECM .....	24
6.1.2 PAINEL.....	26
6.1.3 AQUISIÇÃO DE DADOS .....	26
6.2. DIMENSIONAMENTO DA REDE.....	27
6.3 PROJETOS DE <i>HARDWARE</i> .....	28
6.3.1 INTERFACE COM ECM .....	29
6.3.2 PAINEL.....	30
6.3.3 AQUISIÇÃO DE DADOS .....	32
6.4 DISCUSSÕES .....	33
<b>7. CONCLUSÕES .....</b>	<b>35</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>36</b>
<b>APÊNDICE A – PRIMEIRO CRONOGRAMA PROPOSTO .....</b>	<b>37</b>
<b>APÊNDICE B – SEGUNDO CRONOGRAMA PROPOSTO .....</b>	<b>38</b>
<b>APÊNDICE C – NÓS FEITOS EM MATRIZES DE CONTATOS.....</b>	<b>39</b>
<b>APÊNDICE D – REDE CAN COM <i>HARDWARES</i> AUXILIARES.....</b>	<b>40</b>
<b>APÊNDICE E – FLUXOGRAMA DA INTERFACE COM ECM.....</b>	<b>41</b>
<b>APÊNDICE F – FLUXOGRAMA DO PAINEL .....</b>	<b>43</b>
<b>APÊNDICE G – FLUXOGRAMA DA AQUISIÇÃO DE DADOS .....</b>	<b>44</b>

## 1. INTRODUÇÃO

Uma rede de controle e troca de dados baseada no protocolo CAN é constituída por nós, que são módulos eletrônicos independentes, distribuídos e conectados por um barramento de comunicação comum. Os módulos desempenham tarefas específicas. Podem possuir um ou mais microcontroladores responsáveis por gerenciar o envio e recebimento de mensagens no barramento. A comunicação através do barramento obedece ao protocolo CAN, o qual possui comunicação serial síncrona e atende todas as normas de segurança e desempenho (Hubert, 2001). O protocolo é fundamentado no conceito *Carrier Sense Multiple Access / Collision Detection with Non-Destructive Arbitration* (CSMA/CD with NDA) no qual as mensagens geradas possuem prioridade determinada e desta forma são enviadas ao barramento preferencialmente, caso duas ou mais mensagens tentem acessá-lo no mesmo instante. Há também detecção de falhas pela rede CAN, podendo ela adaptar-se a esta situação sem parar todo o sistema.

Além da eficiência apresentada, a rede CAN apresenta pouco cabeamento interligando os módulos, com menos risco de maus contatos e facilitando a montagem do sistema. Outra vantagem é a facilidade na ampliação do sistema, caso necessário, sem a necessidade de alterá-lo completamente. Apenas os módulos são reconfigurados, alterando-se a programação dos microcontroladores (Guimarães, 200-?). Tem-se então um sistema que pode ser robusto, versátil e que pode ter um custo relativamente baixo, dependendo da escolha dos componentes. Este tipo de rede atende as necessidades do mercado automobilístico. Atualmente as grandes montadoras automobilísticas têm usado redes CAN em diversos modelos de automóveis.

Consciente das vantagens proporcionadas por este tipo de tecnologia e o seu crescente uso, decidiu-se desenvolver uma rede CAN para o veículo de fórmula SAE da equipe EESC - USP. O desenvolvimento desta rede vai de encontro a um dos objetivos da equipe, a inovação tecnológica visando à melhora do desempenho do veículo.

Este projeto tem grande importância, pois permitirá a ampliação da rede de maneira fácil e rápida, fazendo com que novas tecnologias eletrônicas possam ser inseridas sem grandes complicações, evitando excesso de cabos e otimizando o veículo de Fórmula SAE, de acordo com os interesses da equipe, além de servir de estudo de caso para alunos interessados na tecnologia de redes de controle.

## 1.1 CONTROLLER AREA NETWORK (CAN)

O protocolo de comunicação serial digital CAN promove a interconexão dos dispositivos de controle em automóveis.

Segundo Sakai (2008), as principais características deste protocolo são:

- Rede multimestre: o barramento pode ser utilizado por qualquer nó da rede, desde que esteja livre;
- Taxa de comunicação pode variar até “1 Mbps”;
- Quadros de dados com no máximo “8 bytes”, para trocas de informações concisas;
- Arbitragem para acesso ao barramento sem colisão;
- Campo de *bits* na mensagem para identificação de erros;
- Possibilidade de adição, remoção e mudanças de dispositivos para se obter diferentes configurações do sistema;

Existem três formas de se constituir um barramento CAN, as quais dependem da quantidade de fios utilizada. Existem redes baseadas em um, dois e quatro fios. As redes com dois e quatro fios trabalham com os sinais de dados CAN *High* (CAN\_H) e CAN *Low* (CAN\_L). No caso dos barramentos com quatro fios, além dos sinais de dados, um fio com a alimentação (VCC) e outro com a referência (GND) fazem parte do barramento, levando a alimentação às duas terminações ativas da rede. As redes que utilizam um fio usam-no exclusivamente com linha CAN.

O CAN fundamentado em dois e quatro fios deve ter seus condutores elétricos trançados e não blindados. Os dados enviados através da rede são interpretados pela análise da diferença de potencial entre os fios CAN\_H e CAN\_L. Por isso, o barramento CAN é classificado como par trançado diferencial. Este conceito atenua fortemente os efeitos causados por interferências eletromagnéticas, pois a interferência afeta igualmente os fios, não alterando a diferença de potencial entre eles e conseqüentemente, não afetando o recebimento das mensagens.

No CAN, os dados não são representados por *bits* em nível “0” ou nível “1”. São representados por *bits* Dominantes e *bits* Recessivos, criados em função da condição presente nos fios CAN\_H e CAN\_L (Guimarães, 200-?).

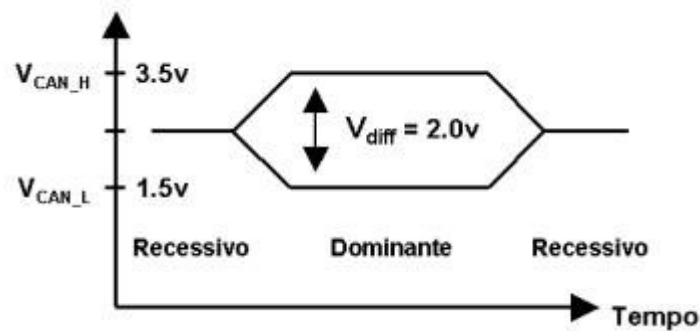


Figura 1. Bits dominantes e recessivos

O protocolo CAN possui camadas físicas e de enlace de dados definidas pelo padrão Open System Interconnection (OSI). As outras cinco camadas restantes ficam abertas para implementações de alto nível de acordo com cada aplicação. (Sakai, 2008)

Existem diversos padrões que são fundamentados no CAN, sendo os principais: NMEA 2000, SAE J1939, DIN 9684, ISO 11783.

As mensagens do protocolo CAN possuem formato fixo e são formadas por campos de *bits*. Com relação ao campo identificador, determina-se o CAN 2.0A, que é o quadro padrão, e 2.0B, que é o quadro estendido sendo que o primeiro possui “11 *bits*” e o segundo “29 *bits*” no campo identificador. Abaixo está uma figura com os campos de *bits*, referente ao CAN 2.0B (Sakai, 2008).

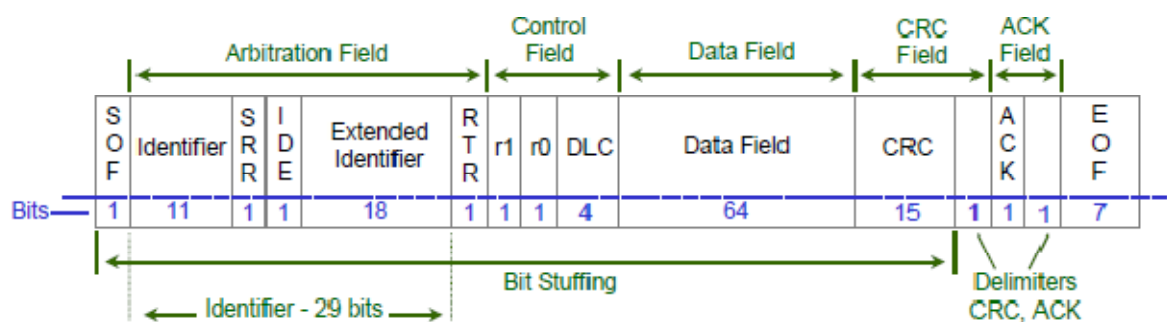


Figura 2. Campos de bits do CAN 2.0B

Segundo Sousa (2002), os campos de bits podem ser definidos como:

- *SOF (Start of Frame) Field* (Campo de Início de Quadro): É composto por um bit dominante que indica início de quadro;
- *Arbitration Field* (Campo de Arbitragem): Contém o identificador e um bit denominado RTR (*Remote Transmission Request*). Se o RTR for 0, o quadro é do tipo quadro de dados, caso seja 1, indica que o quadro é do tipo quadro remoto;

- *Control Field* (Campo de Controle): É composto por 6 bits de controle. O primeiro é o IDE (*Identifier Extended Bit*), responsável por sinalizar se o quadro é padrão, caso o bit seja dominante, ou estendido, caso o bit seja recessivo. O bit seguinte, r0, é reservado para novas aplicações em futuras versões do CAN. Os quatro últimos bits deste campo formam um conjunto denominado DLC (*Data Length Code*), que indica o número de bytes no campo de dados;
- *Data Field* (Campo de Dados): É composto por “8” bytes (64 bits) com o *bit* mais significativo do primeiro byte de dados sendo transmitido primeiro. Os bytes podem representar diversas informações, uma informação ou partes de uma informação;
- *CRC (Cyclic Redundance Check) Field* (Campo de Verificação de Redundância Cíclica): É composto por “15” bits, utilizados para implementação do código de detecção de erros e um bit recessivo delimitador deste campo. O código é calculado de acordo com um polinômio específico definido para o CAN, e utiliza os “4” primeiros campos do quadro de dados (SOF, Arbitration Field, Control Field e Data Field) para o cálculo. Este código introduz uma redundância suficiente para permitir que receptores deduzam que houve um erro durante a transmissão;
- *ACK (Acknowledge) Field* (Campo de Confirmação): consiste em dois bits, um bit denominado *ACK*, recessivo, que é sobrescrito por bits dominantes transmitidos de um outro nó que recebe a mensagem com sucesso, confirmando o recebimento. O segundo bit é recessivo delimitador;
- *EOF (End of Frame) Field* (Campo de Fim de Quadro): É composto por “7” bits recessivos, indicando o fim do quadro.

Neste projeto utilizou-se o padrão SAE J1939, o qual se baseia no CAN 2.0B que tem aplicação principalmente no setor automotivo, especialmente em ônibus e caminhões.

### 1.1.1 MENSAGEM CAN

De acordo com o CAN 2.0 B, as mensagens possuem um quadro de *bits* com diversos campos definidos. Estes campos possuem um ou mais *bits*, e servem para identificação, *Data Length Code* (DLC), dados, controle de transmissão ou detecção de

erros no envio. Para os usuários, os campos relevantes são o campo identificador de 29 *bits*, o campo DLC de “4 *bits*” e o campo de dados de “0” a “8 bytes”.

Considerando-se o padrão SAE J1939, o campo identificador pode ser dividido em seis novos campos:

- Prioridade: Formado por “3” *bits*, que definem a prioridade da mensagem;
- Reservado (R): Formado por “1” bit, que é sempre enviado como “0”;
- *Data Page* (DP): Formado por “1” bit;
- *Protocol Data Unit Format* (PDUF): Formado por “8” *bits*;
- *Protocol Data Unit Specific* (PDUS): Formado por “8” *bits*;
- Endereço Origem: Formado por “8” *bits*;

A prioridade varia com a importância da mensagem, e pode decidir qual mensagem será enviada primeiro ao barramento, quando duas mensagens são enviadas ao mesmo tempo por nós diferentes. O endereço origem é característica do nó que está enviando a mensagem, e conterá sempre o endereço utilizado por ele, na rede, naquele momento. Os campos R, DP, PDUF e PDUS, em conjunto, definem o campo *Parameter Group Number* (PGN), e identificam a mensagem que está sendo enviada. Assim, os outros nós que recebem uma mensagem podem decidir se utilizam os seus dados ou não, dependendo de cada aplicação.

Muitos PGNs já foram definidos pela norma, sendo que alguns deles são reservados para uso proprietário. Para a aplicação na equipe, usaram-se PGNs proprietários.

### 1.1.2 PGN

O PGN é formado pelos campos Reservado, *Data Page*, PDUF e PDUS. O campo PDUF tem o tamanho de “1 byte”. Seus possíveis valores são de “0” a “255”. Duas formas são possíveis para enviar uma mensagem. Uma delas é com destino específico (DA), ou seja, a mensagem é direcionada a somente um nó. Outra forma é com destino global, ou seja, a mensagem é enviada para todos os nós, que devem decidir se analisam os dados contidos ou não. Por esse motivo, o campo PDUF foi dividido em dois campos de valores, como na tabela abaixo:

Tabela 1 - Divisão do PDUF

Formato	Campo PDUF	Campo PDUS
PDU1	0 – 239	Endereço destino (DA)
PDU2	240 – 255	Extensão de grupo

O formato PDU1 define mensagens com destino específico, e contém no campo PDUF valores entre “0” e “239”. Neste caso, o campo PDUS contém o endereço destino (DA), e não é considerado parte do PGN.

O formato PDU2 define mensagens com destino global, e o campo PDUS complementa o PGN, e é definido por *Group Extension* (GE).

Desta maneira, os PGNs podem ser formados por estes campos da seguinte maneira:

Tabela 2 - Formação do PGN

R	DP	PDUF	PDUS	PGN
0	0	0	DA	0
0	0	1	DA	256
0	0	....	DA	...
0	0	238	DA	60928
0	0	239	DA	61184
0	0	240	0	61440
0	0	240	1	61441
0	0	...	...	...
0	0	255	254	65534
0	0	255	255	65535
0	1	0	DA	65536
0	...	...	...	....

Nota-se que o campo *Reservado* é sempre “0”. O campo *Data Page* será usado como “1” quando as possibilidades para utilizá-lo como “0” esgotarem-se.

Os PGNs reservados para uso proprietário são aqueles com PDUF igual a “255” para mensagens globais e PDUF igual a “239” para mensagens com destino específico. Ou seja, utilizaram-se os seguintes PGNs:



Tabela 3 - PGNs utilizados

R	DP	PDUF	PDUS	PGN
0	0	239	DA	61184
0	0	255	0	65280
0	0	255	1	65281
0	0	255	...	...
0	0	255	255	65535

Também se utilizaram algumas mensagens para o gerenciamento de rede como *Address Claimed*, mensagem para pedir informações *Request PGN*, mensagem de confirmação - *Acknowledgment (ACK)* ou não confirmação - *Negative Acknowledgment (NACK)*, etc.

## 2. OBJETIVOS

O projeto teve como objetivo o estudo e desenvolvimento de uma rede CAN composta de três nós, montada em bancada, para possibilitar uma aplicação futura no protótipo Solid Edge EESC - USP de fórmula SAE. O projeto focou a estrutura da rede, a programação dos microcontroladores e o projeto de *hardware* dos módulos eletrônicos.

### 3. REVISÃO BIBLIOGRÁFICA

Na década de oitenta houve grande aumento de componentes eletrônicos e telemetria em veículos. Isto fez crescer a quantidade e a espessura dos cabos. Reconfigurações na rede e manutenção nos nós tomavam muito tempo e mostravam-se como fontes de erros.

Para solucionar estes problemas a empresa Robert Bosch propôs, em 1980, a rede CAN, que permite a interconexão de componentes de controle em veículos. Esta rede apresenta diversas vantagens e tornou-se uma opção bastante utilizada para a montagem de uma rede de dados veicular (Barbosa, 2003).

Como principais vantagens pode-se citar: utilização de cabos com número reduzido de fios, facilidade de instalação, facilidade de manutenção e flexibilidade de expansão. Como desvantagem cita-se o fato do aumento da complexidade do sistema causado pela implementação de protocolos de rede (Sakai, 2008).

Outras vantagens são a proteção contra interferências eletromagnéticas, mecanismos para detecção de erros e método de arbitragem para acesso ao barramento. Este tipo de rede oferece a possibilidade de se trabalhar com faixas de transmissão que variam de poucos “Kbps” a “1 Mbps”.

A rede CAN é padronizada mundialmente pela resolução ISO 11898, ou ISO 11519 criada pela *International Society of Organization*. Os diferentes tipos de redes CAN e aplicabilidade de cada uma em diferentes veículos são regulamentados pela *Society of Automotive Engineers* (Barbosa, 2003).

As normas ISO11898 e ISO11519 definem as camadas Física (1) e de Dados (2), de acordo com a taxa de transmissão, sendo a primeira para altas taxas de transmissão e a segunda para baixas taxas.

As camadas de Rede (3), Transporte (4), Sessão (5), Apresentação (6) e Aplicação (7) podem ser especificadas por diversos padrões como NMEA 2000, SAE J1939, DIN 9684, ISO 11783.

Sousa (2002) sistematizou o CAN e os padrões desenvolvidos baseados no protocolo CAN, como, por exemplo, o padrão SAE J1939. Em seu trabalho, gerou documento referencial prático para uso do CAN em aplicações agrícolas. Desenvolveu um circuito eletrônico de interface entre os padrões RS232 e o CAN. Embora voltado para aplicações agrícolas, o trabalho contribui para qualquer estudo inicial sobre o protocolo CAN.

Neste projeto adotou-se o padrão SAE J1939. Este padrão de comunicação é utilizado em aplicações automotivas. Baseia-se no tipo de quadro CAN 2.0B, que tem campo identificador de “29 bits”, o que implica na possibilidade de se ter mais de “500”

milhões de mensagens diferentes. O tipo de quadro CAN 2.0A apresenta “11” *bits* no campo identificador, permitindo “2048” mensagens (Sousa, 2002).

Definem-se através da J1939 todos os níveis necessários a um protocolo para que o mesmo possa ser efetivamente utilizado em aplicações completas.

A norma SAE J1939 é utilizada em empresas como Dearborn, Motorola, Eaton, Caterpillar, Detroit Diesel, Bosch, TACOM e Alphabet Stoneridge, entre outras, visando aplicações em caminhões e ônibus (Guimarães; Saraiva, 2003).

Devido à crescente aplicação da rede CAN, novas tecnologias tem sido disponibilizadas e a quantidade de fabricantes de produtos voltados para este setor tem crescido. Como consequência o desenvolvimento de produtos com base nesta tecnologia tem seu custo reduzido.

## 4. MATERIAIS E MÉTODOS

A princípio, o projeto da rede previa três nós: Interface com *Engine Control Module* (ECM), *Shifter*, que é o sistema responsável pelas trocas de marcha, e Aquisição de Dados. No decorrer do projeto a equipe EESC - USP demonstrou-se satisfeita com seu desenvolvimento e cogitou a hipótese de implementá-lo no veículo. Como a rede era um protótipo ela poderia apresentar falhas e estas não poderiam comprometer o desempenho de todo o veículo.

De posse desta premissa decidiu-se por modificar o escopo do projeto, substituindo o nó *Shifter*, pois uma falha neste nó poderia implicar na paralisação das trocas de marcha, pelo nó Painel, o qual não traria conseqüências graves em caso de falha.

Após pesquisar e analisar componentes e protocolos Sousa (2002) criou um protótipo de ECU bem sucedido, com o intuito de estabelecer uma interface RS232/CAN, que permite enviar e receber informações do barramento CAN através de um PC. Este projeto utilizou a ECU proposta por Sousa como modelo, com algumas modificações.

Para a execução do projeto utilizou-se, com relação ao *hardware*: três microcontroladores PIC 18F258, três transceptores CAN MCP 2551, conectores do tipo DB9, componentes eletrônicos como resistores, capacitores, reguladores de tensão, diodos e fotoacopladores, transistores BC 547 e osciladores de 10 MHz, *leds* e display de sete segmentos. Utilizou-se um gravador de PIC para gravar a programação nos microcontroladores e cabos para se fazer o barramento CAN. Nos circuitos auxiliares foram utilizados placas universais, botões, potenciômetros, *leds* e resistores.

No desenvolvimento do *hardware* foram utilizados os *softwares* *Altium Designer*, que permitiu o desenho esquemático do circuito impresso e a geração do *Printed Circuit Board* (PCB) para posterior uso no *software* *CircuitCam*, que gera o arquivo utilizado na fresa do departamento de elétrica da EESC, para confecção das placas.

Para verificar o funcionamento da rede fez-se necessário o uso de um *hardware* auxiliar, semelhante aos nós da rede, chamado de *Sniffer*. O *Sniffer* converte as mensagens do barramento CAN para serial, e vice-versa. Isso permite que as mensagens sejam lidas através de uma interface feita em LabView, mostrada nas figuras um e dois, e permite também que mensagens possam ser enviadas do computador para o barramento.

Fez-se a programação da rede com o *software* *MPLAB*, fornecido pela própria fabricante dos microcontroladores, utilizando linguagem C. O compilador utilizado foi o C18.

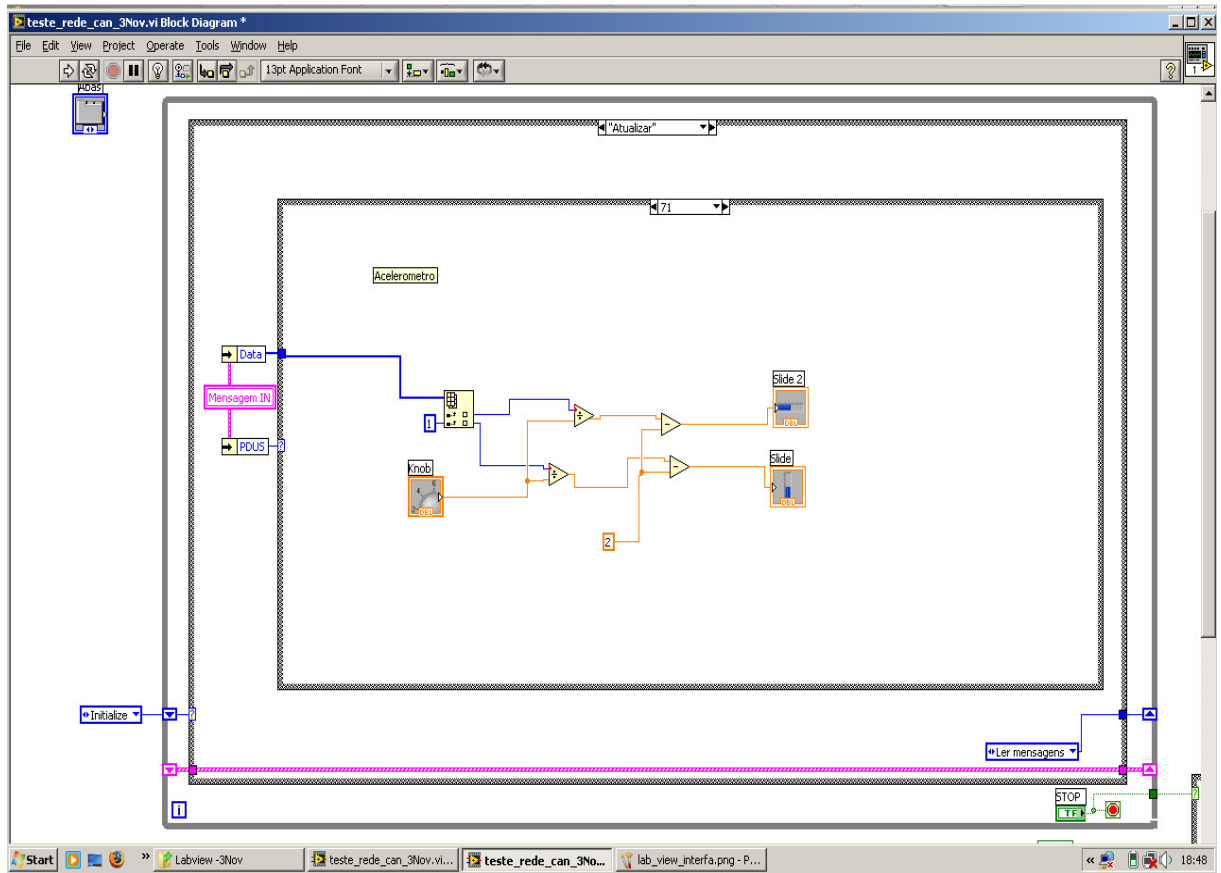


Figura 3. Programação da interface em LabView

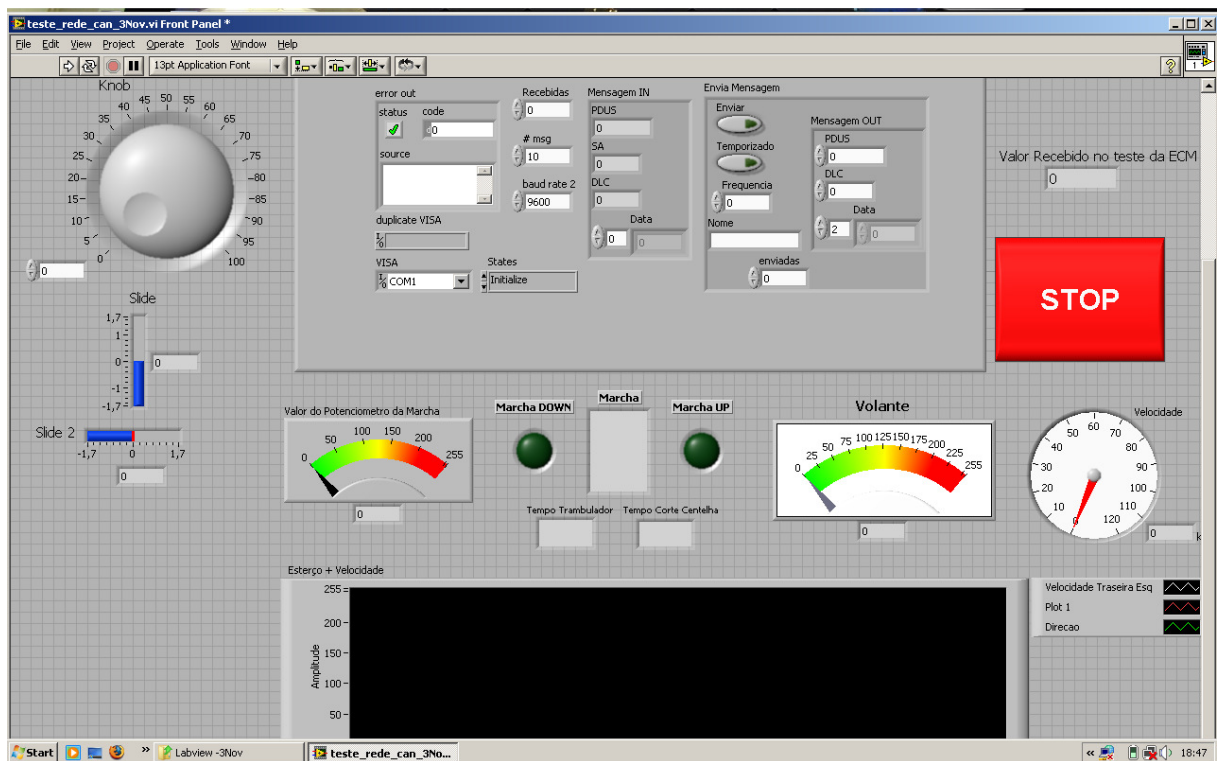


Figura 4. Interface em LabView

## 6. RESULTADOS

Iniciou-se o projeto com o estudo do funcionamento de uma rede CAN. Estudaram-se as normas que definem a camada de usuário da rede. Dessas, foi definida qual seria mais adequada aos interesses do projeto e da equipe EESC - USP de fórmula SAE. Definiram-se, então, quais mensagens seriam colocadas no barramento. O próximo passo foi definir qual seria o microcontrolador utilizado, sendo escolhido o *Microchip PIC 18F258*, devido a sua facilidade de uso e a sua utilização por Sousa (2002) em aplicações agrícolas, apresentando bons resultados.

Após um estudo prévio da folha de dados do microcontrolador e componentes do *hardware*, pôde-se desenvolver o *hardware*, composto de nós e barramento.

No cronograma inicial, *apêndice a*, toda a programação da rede seria construída antes da implementação do *hardware*. Não havia, no entanto, meios para testá-la e confirmar seu funcionamento. Como consequência, construiu-se o *hardware* primeiro, para, posteriormente, dar continuidade a programação. Fez-se, então, um segundo cronograma, para suprir as necessidades que surgiram, *apêndice b*.

Construíram-se três nós em matrizes de contato, *apêndice c*, o que permitiu a execução de alguns testes simples de troca de mensagens. No entanto a matriz de contato apresentou, com frequência, maus contatos. Isto forçou a construção dos protótipos em circuito impresso, tornando o *hardware* confiável. Além disso, fizeram-se circuitos auxiliares para simular condições reais do veículo.

Por meio de sensores indutivos posicionados em frente a uma ventoinha de PC, com um pequeno metal fixado em uma de suas hélices, simulou-se o sensor de velocidade. Potenciômetros simularam a posição do volante e a marcha. Botões simularam o limiar de temperatura da água e baixa pressão do óleo lubrificante. Utilizando o *hyperterminal* do PC simulou-se a comunicação serial com a ECM, *apêndice d*.

Descartados os problemas de maus contatos no *hardware*, desenvolveu-se a programação, testando-a e otimizando-a de forma rápida. Nota-se, como consequência, uma otimização do cronograma, já que se montou, validou e testou o protótipo paralelamente à programação da rede.

Com o amadurecimento do projeto surgiram novas idéias. Construíram-se novos protótipos, visando obter o melhor desempenho e maior confiabilidade. Isso fez decair o risco da rede CAN prejudicar o desempenho do veículo de fórmula SAE, quando implementada.

Descreve-se a seguir todo o procedimento realizado para a construção de *software* e *hardware* de cada nó.

## 6.1 MENSAGENS

Definiram-se as mensagens em função de cada nó da rede. Os nós são chamados de: *Interface com ECM, Painel e Aquisição de Dados*. A seguir é apresentada uma planilha com a definição das mensagens de acordo com cada nó.

Tabela 4 - Definição das mensagens

Módulo Relacionado	Mensagem	#define	msg.Specific	Data[0]	Data[1]	Data[2]
Shifter	Corte de Centelha	MSG_CORTE_CENTELHA	30			
	Trocar Marcha	MSG_MUDANCA_MARCHA	31	0 -> Recua, 1 -> Avança		
	Modo Automático	MSG_MODALITO_AUTOMATICO	32	0 -> Desativa, 1 -> Ativa		
	Marcha Atual	MSG_MARCHA	33	Marcha Atual	Valor do Potenciometro	
	Configuração do Shifter	MSG_CONFIG_SHIFTER	39	Tempo Trambulador	Tempo Corte de Centelha	Multiplicador do tempo do tramb
Interface ECM		MSG_RPM	40	RPM Low	RPM High	
		MSG_DATA_ECM	42	TPS	Temp. Ar	Temp. Agua
		MSG_TPS_MIN_MAX	43	TPS Minimo	TPS Maximo	
		MSG_VBAT				
		MSG_CONFIG_INTECM	49	Valor de teste da comunicacao serial		
Aquisição 1		MSG_VELOCIDADE	66	Velocidade Atual		Valor do Potenc. De esterço
		MSG_DIRECAO	195	Valor do Potenc. De esterço		
		MSG_BEACON				
		MSG_BRAKE	69			
PAINEL		MSG_STOP	91			
		MSG_CONFIG_PAINEL	99			

### 6.1.1 INTERFACE COM ECM

Este nó faz a interface com o módulo de controle do motor, ECM, já que este não possui uma interface CAN. Torna-se possível, então, a comunicação entre a ECM e o barramento CAN, ou seja, eles passam a trocar mensagens. Inicialmente obter-se-iam os sinais dos sensores dividindo os cabos dos sensores e analisando os sinais obtidos, por meio do microcontrolador. Considerou-se este, um procedimento arriscado, pois poderia comprometer o funcionamento do veículo, que tem seu funcionamento baseado nestes sensores. A ECM apresenta uma interface serial, a qual se comunica somente pelo *software* da fabricante *Performance Electronics*. Após contato via email e telefone com a empresa, situada em Ohio, EUA, obteve-se parte do código utilizado para efetuar a comunicação. Uma vez conhecida como era feita a comunicação serial, e as equações utilizadas para



cada sinal, obteve-se os valores indicados por cada sensor, sem oferecer riscos ao funcionamento do veículo. Por este motivo a Interface com ECM tem a função de receber as informações de interesse vindas da ECM, pela porta serial, transformá-las para o padrão CAN e enviá-las ao barramento. Este módulo não recebe mensagens vindas do barramento.

Devido ao fato de só enviar mensagens, pôde-se simplificar o *hardware*, pois só há necessidade de estabelecer contato com a porta serial da ECM, que contém apenas três fios, e com o barramento CAN. Eliminou-se todo o recebimento de sinais advindos de sensores.

O fluxograma referente à programação deste nó encontra-se no *apêndice e*. Listou-se a seguir as mensagens que a Interface com ECM envia.

#### ➤ **Mensagens de Saída**

- ***RPM***

Rotações por minuto do motor

- ***Temperatura do Ar***

Obtida através do sinal vindo do sensor de temperatura do ar no coletor de admissão.

- ***Temperatura do Motor***

Obtida através do sinal vindo do sensor de temperatura do fluido refrigerante do motor.

- ***THROTTLE POSITION SENSOR (TPS)***

Obtido através do sensor de posição da borboleta.

- ***TPS\_MIN***

É o valor mínimo que o TPS pode atingir. Este parâmetro é exigido para referenciar-se o sensor.

- ***TPS\_MAX***

É o valor máximo que o TPS pode atingir. Este parâmetro é exigido para referenciar-se o sensor.

### 6.1.2 PAINEL

Este nó substituiu o nó *Shifter* e é responsável por promover a interface entre o veículo e o piloto. Ele recebe e apresenta ao piloto, a marcha em que o carro se encontra, a faixa de rotação do motor, superaquecimento, e baixa pressão do óleo. O fluxograma referente à programação deste nó encontra-se no *apêndice f*. As mensagens que este nó trata são descritas a seguir:

➤ **Mensagens de Entrada**

- ***Marcha atual***
- ***Pressão do óleo lubrificante***
- ***Superaquecimento***
- ***RPM***

### 6.1.3 AQUISIÇÃO DE DADOS

Este nó faz a aquisição de dados, por meio de sensores de velocidade, e posição do volante, e transmite as mensagens listadas a seguir. O fluxograma referente à programação deste nó encontra-se no *apêndice g*.

➤ **Mensagens de Saída**

- ***Rotação Dianteira Direita***

Obtida através do sinal vindo de um sensor indutivo ligado a roda dianteira direita, que permite saber a velocidade de rotação da roda direita e, por conseguinte, a velocidade do carro (baseada nessa roda).

- ***Rotação Dianteira Esquerda***

Obtida através do sinal vindo de um sensor indutivo ligado a roda dianteira esquerda que permite saber a velocidade de rotação da roda esquerda e, por conseguinte, a velocidade do carro (baseada nessa roda).

- ***Posição Volante***

Obtida através de um sinal vindo de um potenciômetro, que permite saber o ângulo em que o volante está inclinado.

- ***Interruptor Freio***

Obtida através de um interruptor que indica acionamento do freio, permite saber quando o freio é acionado.

## 6.2. DIMENSIONAMENTO DA REDE

Após a revisão bibliográfica deu-se início ao dimensionamento da rede. Levou-se em consideração qual seria a norma e o microcontrolador utilizados, e posteriormente estudaram-se os componentes necessários para cada nó.

Para atender as necessidades de uma equipe de fórmula SAE a implementação da rede deveria ser fácil, ter baixo custo, rápida manutenção e possibilidade de construção somente por membros da equipe. O projeto levou também em consideração a possibilidade de obter-se suporte técnico rapidamente.

Analisado esses fatores, concluiu-se que a rede CAN deveria ser implementada com o microcontrolador *Microchip* PIC 18F258. Este PIC possui interface CAN e tem sido bastante utilizado para este fim. A escolha do microcontrolador da *Microchip* baseou-se principalmente no fato da fabricante fornecer gratuitamente o código em linguagem C recomendado pela norma SAE J1939 para a implementação da rede.

A norma SAE J1939 é utilizada principalmente em caminhões e ônibus, além de máquinas agrícolas e supre as necessidades da equipe. Utiliza-se protocolo CAN baseado nesta norma em pesquisas na própria universidade, o que facilita a obtenção de suporte técnico em uma situação crítica.

Esta norma utiliza quadro CAN 2.0B, no qual as mensagens têm identificador de vinte e nove *bits*, o que implica na possibilidade de se ter quinhentos e trinta e sete milhões de tipos de mensagens, aproximadamente. Este valor é muito maior do que o necessário para este projeto. A velocidade de transmissão de dados é de “250 Kbps”, sendo esta a velocidade estabelecida pela SAE J1939. Cada nó tem um *NAME* e *ADDRESS* fixos, para

que não haja riscos de conflitos quando a rede é ligada, o que aconteceu no início do projeto. Descobriu-se que as unidades não conseguiam se comunicar quando o *Sniffer* não estava no barramento. A explicação para isto é que todos os nós utilizavam o mesmo arquivo “Formula.h”. Elas inicializavam exatamente com o mesmo *NAME* e *ADRESS* e não conseguiam se comunicar.

Para solucionar o problema, criou-se um arquivo de projeto do *MPLAB*, com extensão “.mcp”, para cada nó, em pastas diferentes e com definições diferentes.

Assim, dentre as definições do arquivo “Formula.h”, existem as linhas “#define Fórmula\_STARTING\_ADDRESS <Valor>” e “#define Fórmula\_IDENTITY\_NUMBER <Valor>”, que foram declaradas, em cada nó, com os valores apresentados abaixo.

Esta solução não afeta o desempenho da rede, mas tornar-se-ia complexa se muitos nós fossem inseridos e necessitassem deste tipo de ajuste.

Tabela 5 - Definição de NAME e ADRESS

<b>Aquisição 1</b>
#define Fórmula_STARTING_ADDRESS 130 #define Fórmula_IDENTITY_NUMBER 50
<b>Shifter</b>
#define Fórmula_STARTING_ADDRESS 131 #define Fórmula_IDENTITY_NUMBER 51
<b>Interface com ECM</b>
#define Fórmula_STARTING_ADDRESS 132 #define Fórmula_IDENTITY_NUMBER 52
<b>Painel</b>
#define Fórmula_STARTING_ADDRESS 139 #define Fórmula_IDENTITY_NUMBER 59

### 6.3 PROJETOS DE *HARDWARE*

Esta etapa seria feita somente no oitavo mês do projeto, mas foi antecipada devida falta de confiabilidade da matriz de contato. Em contrapartida, a programação dos nós sofreu um pequeno atraso, de acordo com o cronograma. Esta alteração fez-se necessária para se ter a certeza que não havia falhas de *hardware*. Isto otimizou o processo e permitiu a realização da programação em paralelo com testes e validação da rede.

O *hardware* de cada nó é constituído basicamente de um PIC 18F258, um transceptor CAN MCP 2551, um regulador de tensão L7805A, um oscilador de 10MHz, capacitores e resistores, além de quatro conectores DB9.

Uma vez que cada nó da rede recebe informações diferentes, eles têm configurações de *hardware* diferentes. Utilizou-se o *software Altium Designer* para construir o layout do *hardware* dos nós e gerar o PCB, que é a organização dos componentes na placa de circuito impresso e distribuição das trilhas. Por meio do *software CircuitCam* pode-se obter o arquivo utilizado pela fresa para fazer os circuitos impressos, originados dos PCBs.

### 6.3.1 INTERFACE COM ECM

Este nó tem por função receber os dados de sensores através da comunicação serial com a ECM e enviá-los ao barramento. Exerce um papel importante, pois permite o envio desses dados via telemetria da equipe, para levantamento do desempenho do veículo. O desenvolvimento deste *hardware* deu-se primeiramente com a definição dos componentes utilizados, desenhando-se assim o esquemático. Através do esquemático pode-se gerar o PCB, para que então fosse gerado o arquivo necessário para a confecção do circuito impresso. Abaixo estão as figuras relativas a cada um dos processos.

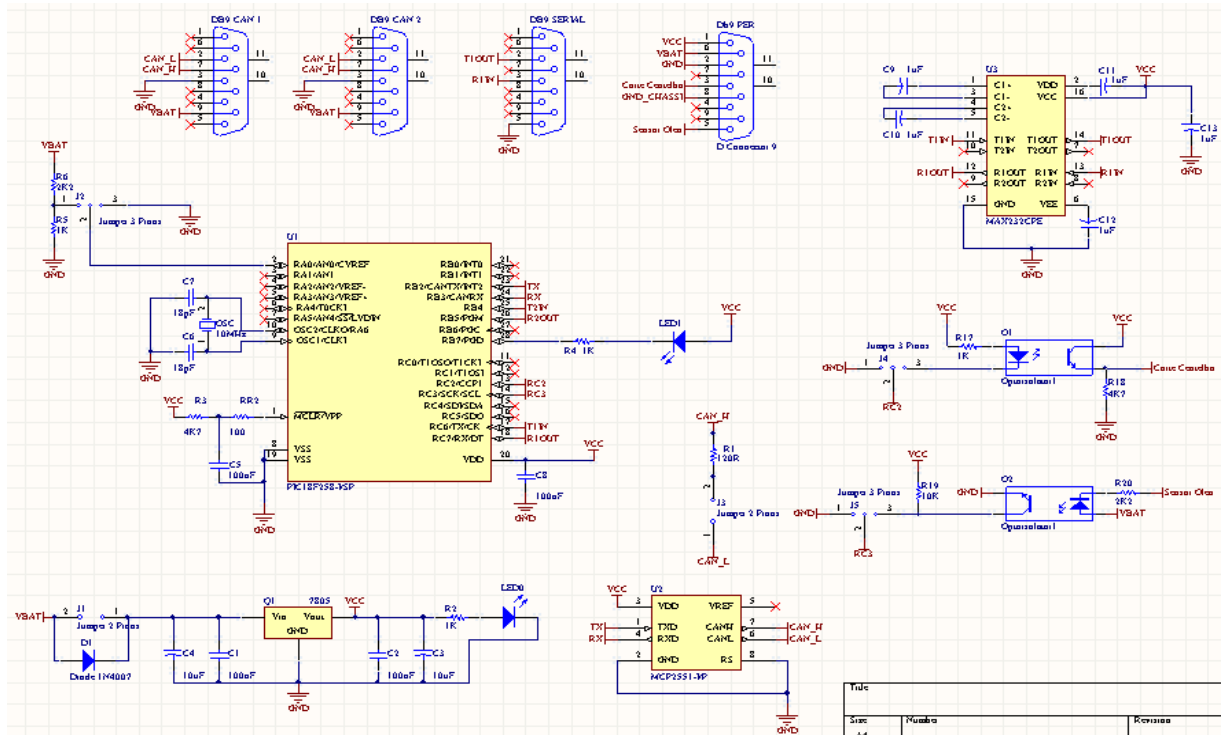


Figura 5. Esquemático da Interface com ECM

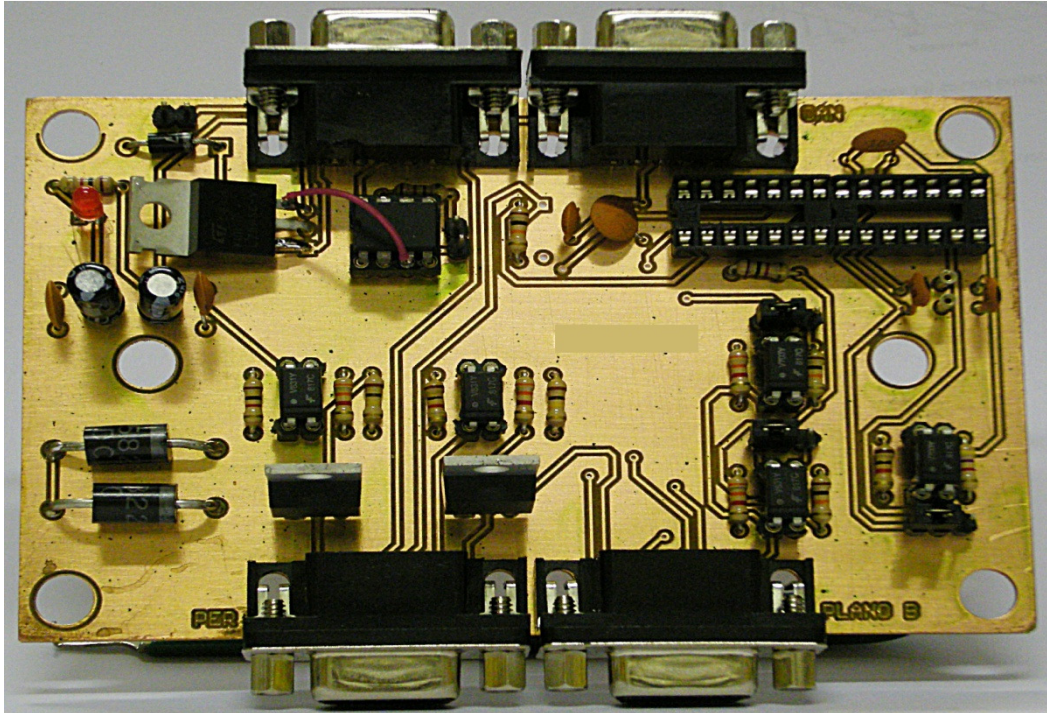


Figura 6. Circuito impresso do nó Interface com ECM

### 6.3.2 PAINEL

Este nó é responsável por fazer a interface entre o veículo e o piloto, apresentando as informações necessárias para um bom desempenho do piloto e do veículo. Apresenta a faixa de RPM do motor, superaquecimento e pressão do óleo, além da marcha que o veículo se encontra. Este nó somente recebe dados vindos do barramento com a função de apresentá-los ao piloto. Fez-se primeiramente o esquemático com todos os componentes necessários, para, então, gerar-se o PCB, que por sua vez foi utilizado para gerar o arquivo necessário para a confecção do circuito impresso. Abaixo estão as figuras relativas a cada um dos processos descritos.

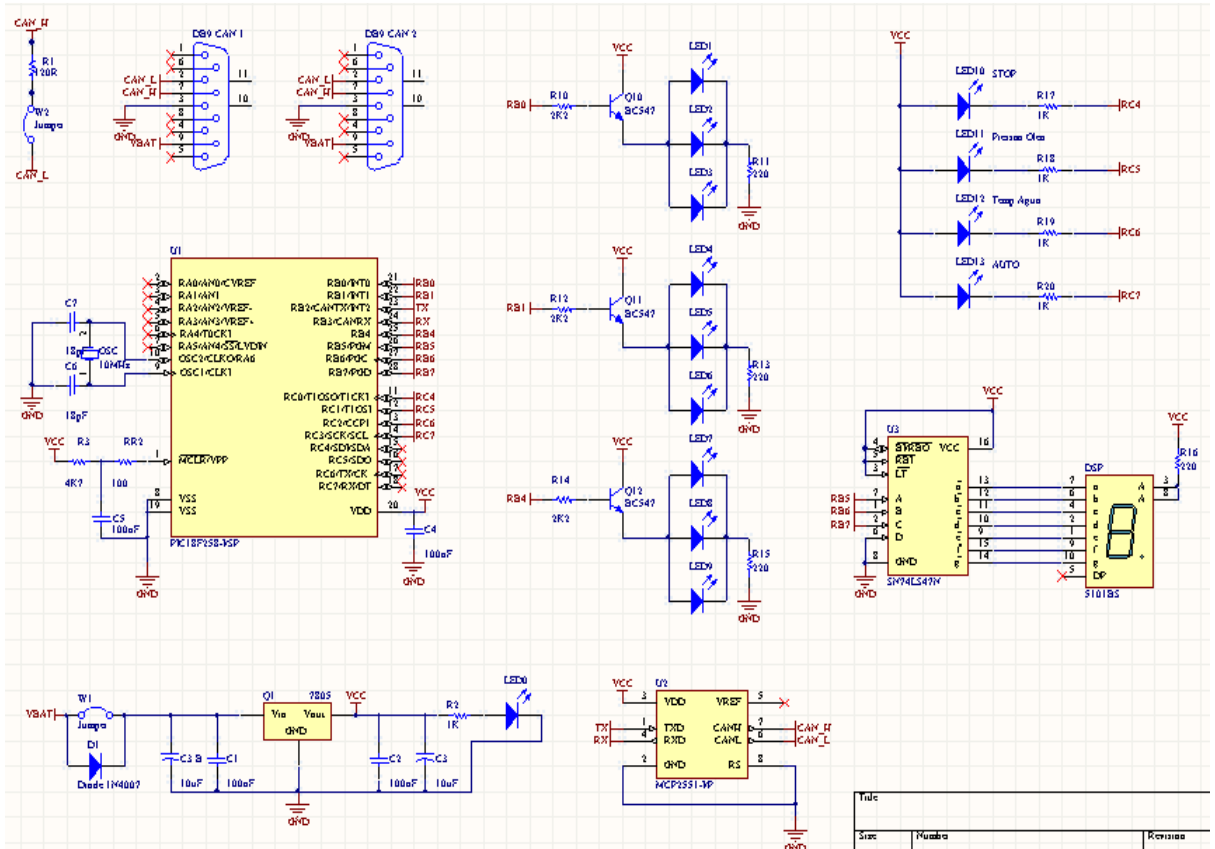


Figura 7. Esquemático do nó Painel

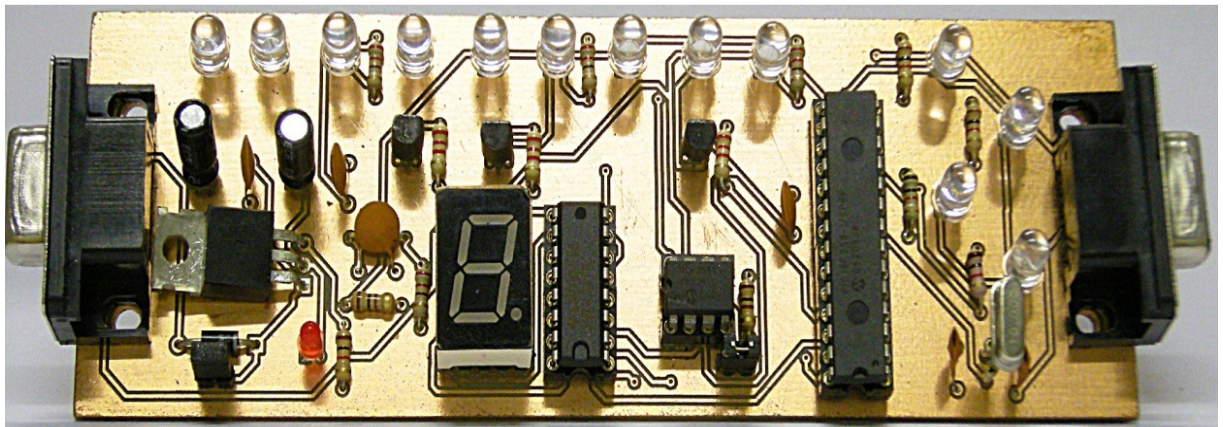


Figura 8. Circuito impresso do nó Painel



### 6.3.3 AQUISIÇÃO DE DADOS

Este nó recebe sinais, como velocidade e posição do volante, vindos de sensores indutivos e potenciômetros. Os sinais foram isolados através de fotoacopladores. Utilizaram-se quatro conectores DB9. Dois deles ficam responsáveis apenas pelo barramento CAN, proporcionando modularidade, já que novos nós podem ser inseridos rapidamente no barramento, sem nenhuma modificação do *hardware*. Os outros dois conectores são responsáveis por receber os sinais. Fez-se um *hardware* auxiliar com potenciômetro e sensores indutivos, além de um botão para simular os sinais recebidos por este nó. As imagens da construção deste nó são mostradas a seguir:

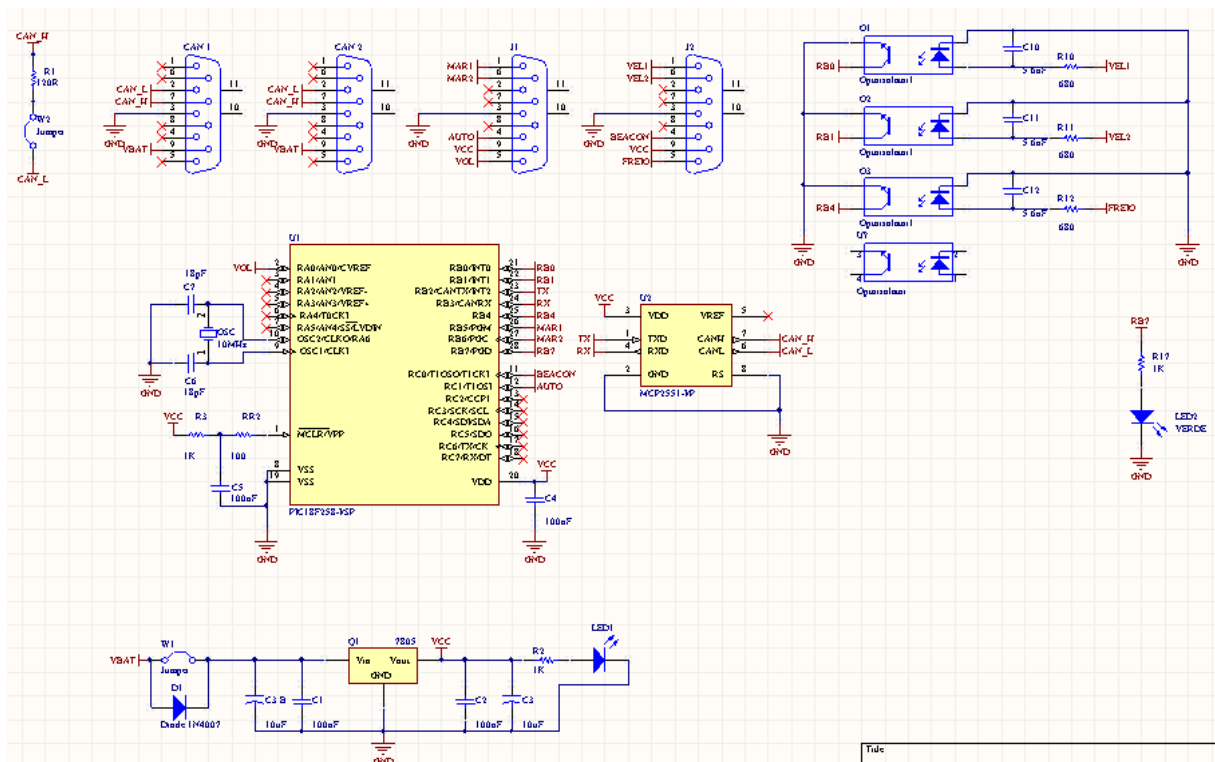


Figura 9. Esquema do nó Aquisição de Dados



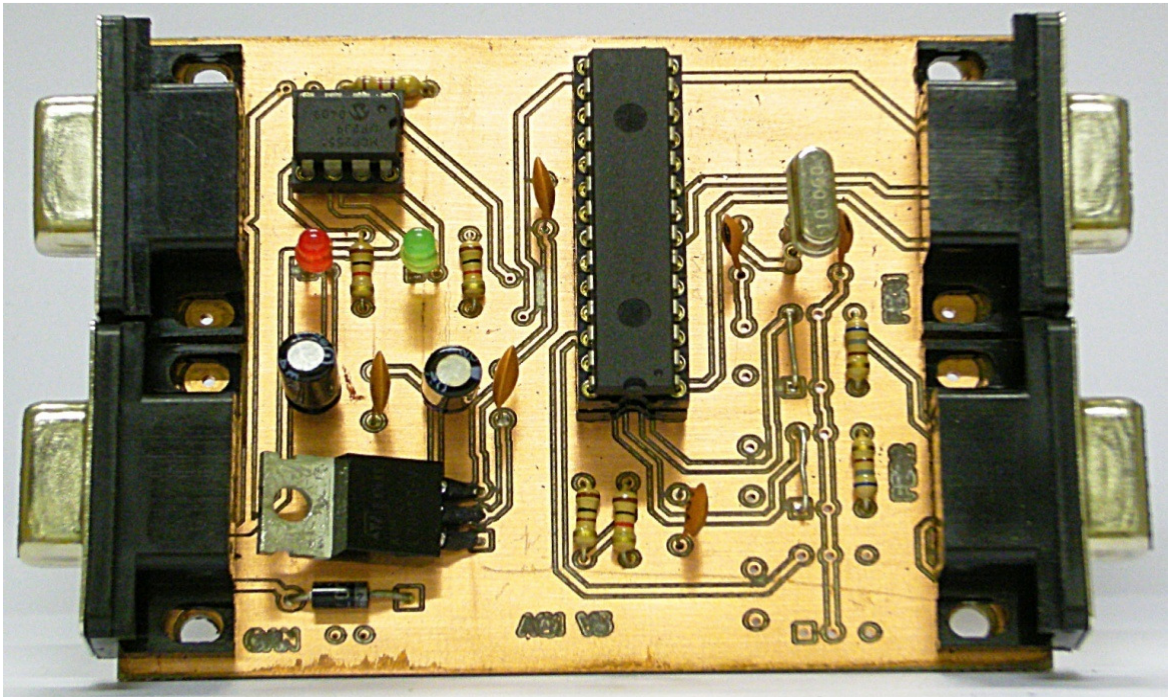


Figura 10. Circuito impresso do nó Aquisição de Dados

## 6.4 DISCUSSÕES

Finalizou-se o projeto conforme o segundo cronograma proposto. Houve adaptações para atender as necessidades da equipe de fórmula SAE, no entanto não houve prejuízos no desenvolvimento do projeto. O nó *Shifter* já está em fase avançada e poderá ser concluído futuramente. No seu lugar desenvolveu-se o nó Painel. Desenvolveu-se a etapa Montagem de Protótipo com antecedência, passando por otimizações, e fez-se a etapa de programação em paralelo com os testes e validação da rede. Isto permitiu testes e otimizações simultâneos.

A alteração no cronograma se deu devido à falta de confiabilidade nas matrizes de contatos. Ocorreram maus contatos e estes poderiam ser confundidos com uma possível falha na programação. Além deste fator, gastava-se muito tempo na detecção destas falhas. Com a confecção dos protótipos em circuito impresso os problemas de *hardware* foram solucionados e pôde-se então trabalhar com a programação, executando testes com *hardwares* auxiliares e verificando as mensagens do barramento.

Montou-se a rede em bancada e utilizando o *Sniffer* realizou-se a comunicação com o PC, apresentando os dados na interface feita em *LabView*. Visualizaram-se assim todas as mensagens que passavam pelo barramento, detectando falhas no recebimento de mensagens. Geraram-se as mensagens a partir dos sinais vindos dos *hardwares* auxiliares, que simularam a troca de marcha, o acionamento do freio e velocidade.

Durante a programação dos nós notou-se um problema. A rede não funcionava sem o *Sniffer*, o que não poderia ocorrer, já que a rede é modular. Constatou-se que os nós estavam utilizando o mesmo arquivo "*Fórmula.h*" e conseqüentemente tinham o mesmo *NAME* e *ADRESS*, o que travava a rede assim que ligada. Solucionou-se este problema gerando arquivos diferentes para cada nó, e atribuindo *NAME* e *ADRESS* fixos para cada um.

Como a rede funcionou de maneira adequada decidiu-se implementá-la no veículo, o que seria um passo a frente do que o cronograma previa. Havia incerteza quanto à robustez da rede, devido ao fato de ser um protótipo, diante de vibração e temperatura. No entanto, obtiveram-se resultados promissores com a rede no veículo, já que esta funcionou satisfatoriamente e a rede mostrou-se adequada para esta aplicação.

Construiu-se, então, uma rede de controle de baixo custo, sendo testada e validada pela própria equipe, através de *hardwares* auxiliares simples e que apresentou robustez suficiente para aplicação em um veículo de fórmula SAE.

Apresentou-se este projeto no Simpósio Internacional de Iniciação Científica da Universidade de São Paulo (SIICUSP), no ano de dois mil e oito, por ter sido tema de iniciação científica do aluno.

## 7. CONCLUSÕES

Mesmo com os novos desafios propostos e diferentes necessidades da equipe de fórmula SAE, concluiu-se o projeto com êxito. Para tanto, fizeram-se necessárias mudanças no escopo e no cronograma do projeto. Seguiu-se o segundo cronograma proposto e realizaram-se todas as etapas no prazo estabelecido, confirmando, assim, o sucesso do projeto.

Desafios surgiram durante o projeto, e puderam ser resolvidos de maneira rápida devido à preocupação com a escolha do microcontrolador, a norma e a facilidade de se encontrar suporte. Um fator determinante para o sucesso foi o comprometimento da equipe envolvida, buscando soluções para cada problema encontrado.

A rede CAN implementada no veículo faz com que cabos de sensores sejam encurtados, pois se coloca os nós em locais próximos aos sensores de interesse, e as informações destes sensores são transmitidas pela rede. A inserção de novas tecnologias eletrônicas baseadas nos sensores foi facilitada, bastando reprogramar os nós. A transmissão dos dados da rede por telemetria será importante para a equipe, que passará a ter informações detalhadas do desempenho do veículo, permitindo análises e otimizações do veículo.

Apesar de todas as vantagens que a rede proporciona, e a preocupação com o custo do projeto, a rede CAN pode apresentar custo final maior do que o sistema já existente no veículo. No entanto, quando o número de sensores e tecnologias eletrônicas do veículo aumentar, a rede CAN passará a ser a opção mais viável e de menor custo.

A complexidade da parte eletrônica do veículo, quanto ao *hardware*, diminui, pois os módulos eletrônicos anteriores não tinham um padrão de microcontroladores e componentes e eram projetados de acordo com cada projetista da equipe. Os nós da rede CAN possuem padrões e semelhanças de *hardware*, o que facilita sua manutenção no veículo e seu entendimento pela equipe. Quanto ao *software*, a complexidade aumentou, pois se utiliza um protocolo de comunicação com diversas tecnologias para tornar a rede CAN robusta e cada nó possui uma programação diferente, referente às suas funções.

Para a equipe o uso da rede CAN é de grande interesse, pois as vantagens que ela proporciona são grandes e podem auxiliar no aumento de desempenho e conhecimento do comportamento do veículo nas mais variadas situações.

Obtiveram-se resultados bastante promissores e novos nós devem ser desenvolvidos, para implementações futuras no veículo, visando sempre à inovação tecnológica e, conseqüentemente, melhor desempenho, tanto da rede quanto do veículo.

## REFERÊNCIAS BIBLIOGRÁFICAS

BARBOSA, L. R. G. **Rede CAN**. 2003. 14p. Escola de Engenharia da UFMG, Universidade Federal de Minas Gerais, Belo Horizonte, 2003.

GUIMARÃES, A. A. [200-?]. CAN BUS complete. Disponível em:  
<[http://www.pcs.usp.br/~laa/Grupos/EEM/CAN\\_Bus\\_Parte\\_1.html](http://www.pcs.usp.br/~laa/Grupos/EEM/CAN_Bus_Parte_1.html)>  
<[http://www.pcs.usp.br/~laa/Grupos/EEM/CAN\\_Bus\\_Parte\\_2.html](http://www.pcs.usp.br/~laa/Grupos/EEM/CAN_Bus_Parte_2.html)>  
<[http://www.pcs.usp.br/~laa/Grupos/EEM/CAN\\_Bus\\_Parte\\_3.html](http://www.pcs.usp.br/~laa/Grupos/EEM/CAN_Bus_Parte_3.html)>  
Acesso em: 10 Jan. 2008.

GUIMARÃES, A. A.; SARAIVA, A. M. **O Protocolo Internacional CAN Bus na Comunicação de Dados de Sistemas Agrícolas: Presente e Futuro**. 4p. 2003.

HUBERT, M. K. **Protocolo CAN como solução para aplicações distribuídas, baseada entre objetos para pc's e microcontroladores**. 2001. 58p. Projeto de Diplomação- Instituto de Física e Matemática, Universidade Federal de Pelotas, Pelotas, 2001.

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.

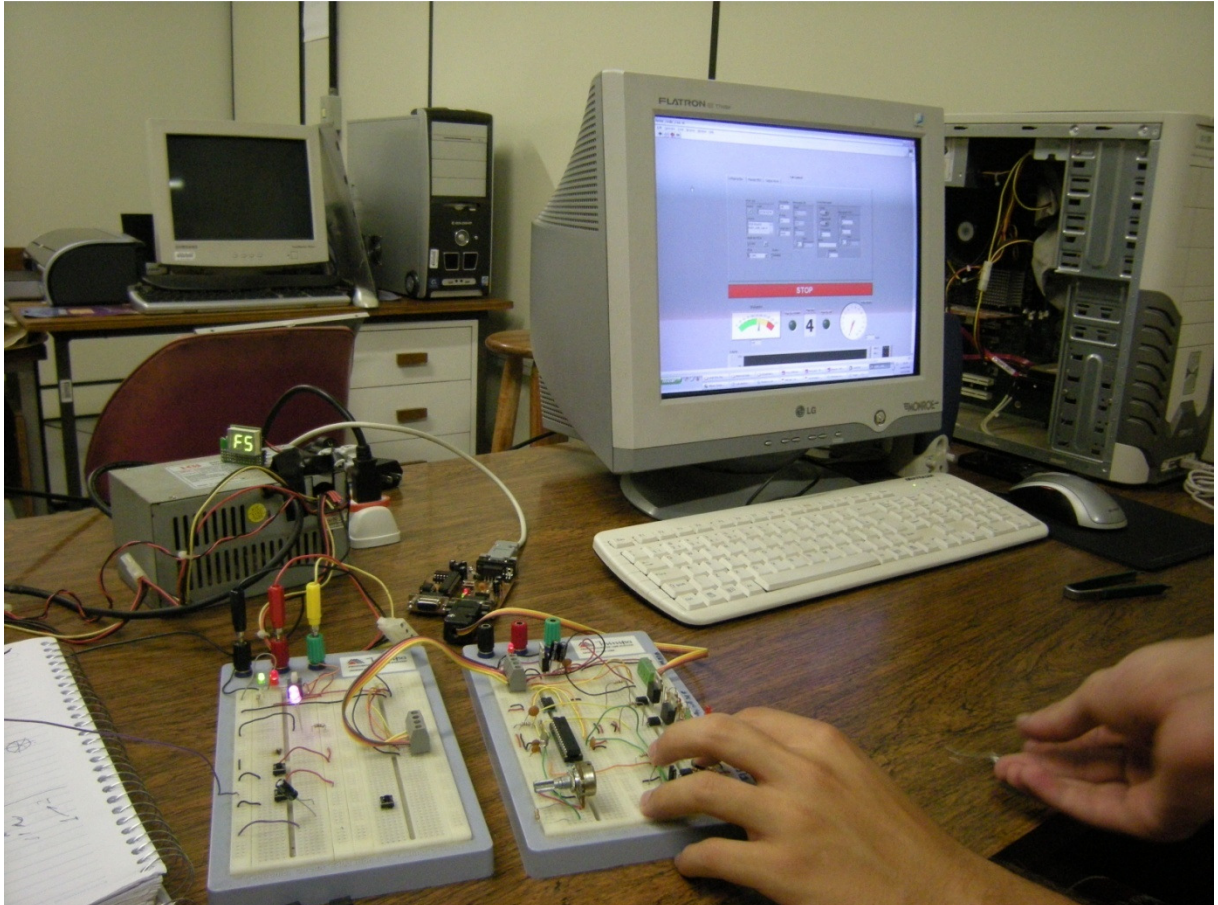
SOCIETY OF AUTOMOTIVE ENGINEERS. **SAE J1939**: Recommended practice for a serial control and communications vehicle network. 2005.

SOUSA, R. V. **CAN (Controller Area Network): uma abordagem para automação e controle na área agrícola**. São Carlos, 2002. 84p. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2002.



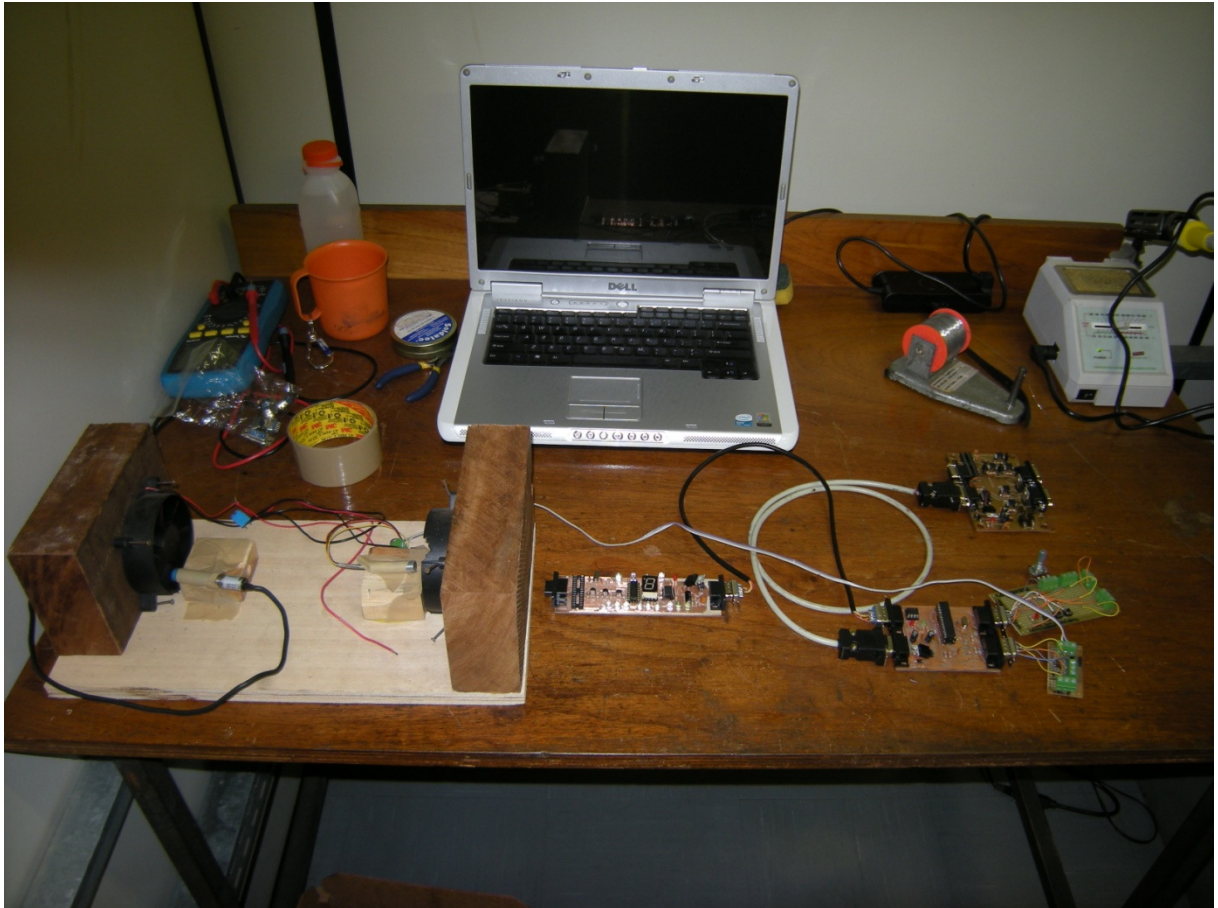


## APÊNDICE C – NÓS FEITOS EM MATRIZES DE CONTATOS

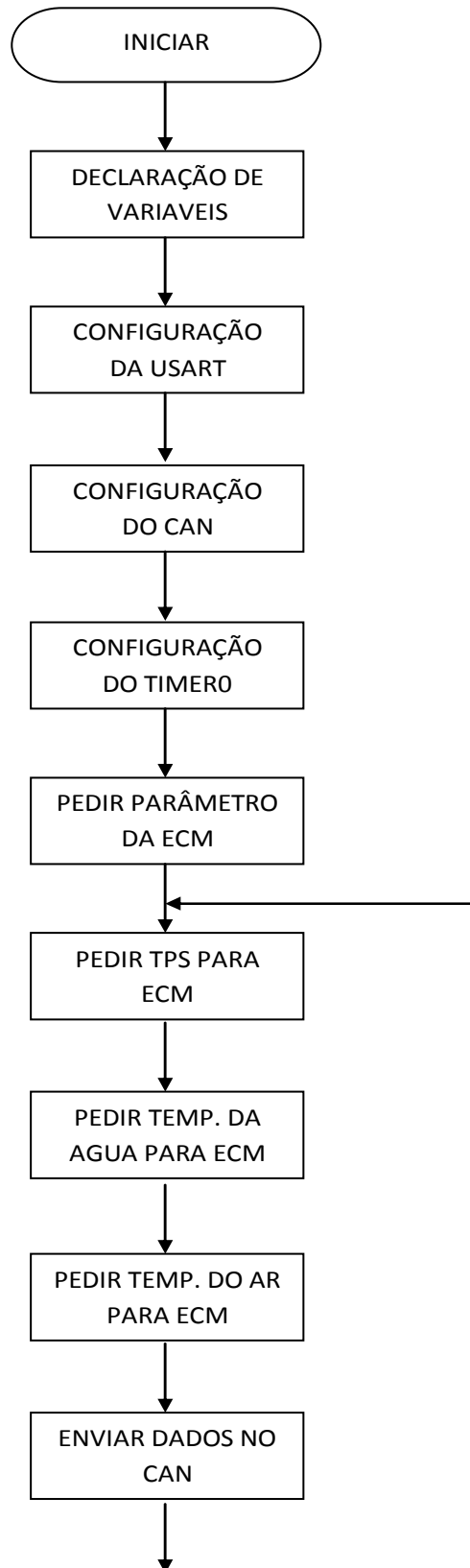


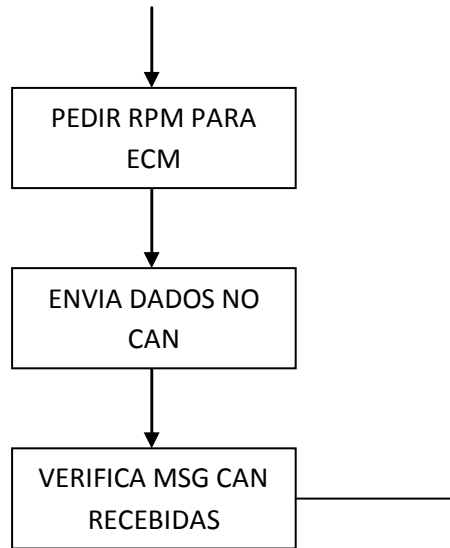


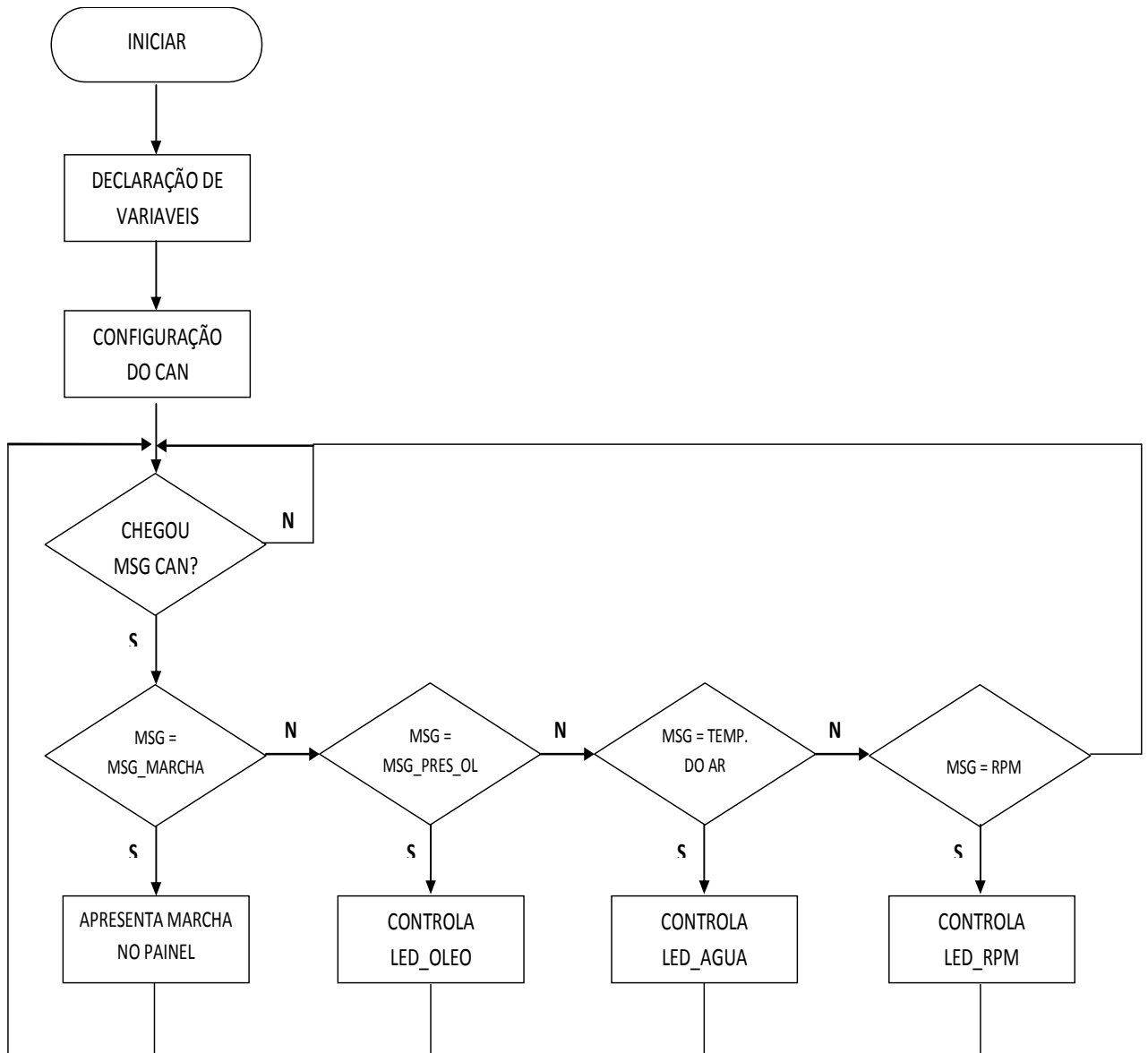
## APÊNDICE D – REDE CAN COM *HARDWARES* AUXILIARES





**APÊNDICE E – FLUXOGRAMA DA INTERFACE COM ECM**



**APÊNDICE F – FLUXOGRAMA DO PAINEL**

**APÊNDICE G – FLUXOGRAMA DA AQUISIÇÃO DE DADOS**