



**UNIVERSIDADE FEDERAL DE OURO PRETO**  
**ESCOLA DE MINAS**  
**COLEGIADO DO CURSO DE ENGENHARIA DE**  
**CONTROLE E AUTOMAÇÃO - CECAU**



**PATRÍCIA GABRIELLY MACEDO MEIRELES**

**SISTEMA DE TELEMETRIA E SUPERVISÓRIO PARA O VEÍCULO BAJA**

**MONOGRAFIA DE GRADUAÇÃO EM ENGENHARIA**  
**DE CONTROLE E AUTOMAÇÃO**

**Ouro Preto, 2017**

PATRÍCIA GABRIELLY MACEDO MEIRELES

**SISTEMA DE TELEMETRIA E SUPERVISÓRIO PARA O VEÍCULO  
BAJA**

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para obtenção do Grau de Engenheiro de Controle e Automação.

Orientador: Prof. Dr. Alan Kardek Rêgo Segundo

Ouro Preto

Escola de Minas – UFOP

Maió/2017

M514s Meireles, Patrícia Gabrielly Macedo.  
Sistema de telemetria e supervisório para o veículo Baja [manuscrito] /  
Patrícia Gabrielly Macedo Meireles. - 2017.

42f.: il.: color; tabs.

Orientador: Prof. Dr. Alan Kardek Rêgo Segundo.

Monografia (Graduação). Universidade Federal de Ouro Preto. Escola de Minas. Departamento de Engenharia de Controle e Automação e Técnicas Fundamentais.

1. Motorsports. 2. Telemetria. 3. Veículos de corridas fora da estrada. I. Rêgo Segundo, Alan Kardek . II. Universidade Federal de Ouro Preto. III. Título.

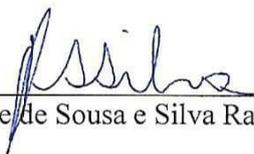
CDU: 681.5

Catálogo: [ficha@sisbin.ufop.br](mailto:ficha@sisbin.ufop.br)

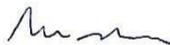
Monografia defendida e aprovada, em 05 de maio de 2017, pela comissão avaliadora constituída pelos professores:



Prof. Dr. Alan Kardek Rêgo Segundo - Orientador



Profa. M. Sc. Regiane de Sousa e Silva Ramalho – Professora Convidada



Prof. Dr. Paulo Marcos de Barros Monteiro – Professor Convidado

*“Todas as vitórias ocultam uma abdicação.”*

*Simone de Beauvoir.*

## **AGRADECIMENTOS**

Gratidão ao Bom Deus por mais esta benção recebida.

Aos meus pais, Marco e Conceição, pelo amor incondicional e por não medirem os esforços para concretizarem mais um sonho meu. À minha irmã, Priscila, companheira de vida, por todo o carinho e paciência. A vitória também é de vocês.

Aos meus grandes amigos do curso por todo o aprendizado compartilhado, em especial à Célia, ao João Paulo, ao Rárisson, ao Rodrigo e ao Itamar.

À UFOP e aos professores do curso de Engenharia de Controle e Automação pelo ensino de qualidade, em especial ao meu orientador Prof. Dr. Alan Kardek pelo apoio e oportunidades.

À equipe Baja da UFOP, em especial ao Marcelo pela amizade.

À Fundação Gorceix por todo o auxílio, em especial ao meu supervisor de estágio o Eng. Rogério Veiga.

A todos que de alguma forma contribuíram positivamente para esta conquista.

## RESUMO

O projeto Baja SAE (Society of Automotive Engineers) da Universidade Federal de Ouro Preto - UFOP, formado por estudantes, possui a finalidade de projetar e construir um veículo Baja - carro do tipo *buggy* para competição *off road* com âmbitos regional, nacional e internacional. O painel de instrumentos do veículo possibilita ao piloto visualizar informações importantes sobre o veículo e a competição, tais como: velocidade, distância percorrida e os níveis de combustível e bateria. Este trabalho tem como objetivo implementar um sistema de telemetria para o veículo, de modo que as informações do painel de instrumentos sejam enviadas via radiofrequência até um sistema supervisor. Com isso, a equipe poderá analisar, diagnosticar ou prevenir problemas de forma antecipada, com a finalidade de melhorar o desempenho durante as competições.

**Palavras-Chave:** Sistema supervisor, telemetria, Baja

## **ABSTRACT**

Baja SAE project of UFOP, formed by students, has the objective of design and build a Baja, which is a car of the kind buggy for national and international off-road competition. The dashboard allows the driver to view important information about the vehicle as speed, distance traveled, fuel levels and battery life. This project aims to implement the telemetry system to the vehicle, which will send the data from the dashboard by radio frequency to a supervisory system. That way, the team will be able to analyze, diagnose or prevent problems forward, intending to improve the performance during the competitions.

**Keywords:** Supervisory system, telemetry system, Baja.

## LISTA DE ABREVIACÕES E SIGLAS

A/D - Conversor Analógico/Digital

D/A - Conversor Digital/Analógico

PC - *Inter-Intergrad Circuit* -

I/O - *Input/Output* - Entrada/Saída

LSB - *Less Significant Bit* - Bit Menos Significativo

MSB - *Most Significant Bit* - Bit Mais Significativo

SAE - *Society of Automotive Engineers* - Sociedade de Engenheiros Automotivos

SCADA - *Supervisory Control and Data Acquisition* - Aquisição de Dados e Controle Supervisório

SCI- *Serial Communications Interface* – Interface de Comunicação Serial

SCL - *Serial Clock*

SDA - *Serial Data*

UFOP - Universidade Federal de Ouro Preto

USART - *Universal Synchronous Asynchronous Receiver Transmitter* - Transmissor/Receptor Universal Síncrono e Assíncrono

## **LISTA DE TABELAS**

Tabela 2.1- Descrição dos pinos do PIC16F877A .....	18
---	----

## LISTA DE FIGURAS

Figura 2-1 Diagrama de pinos do PIC16F877A .....	18
Figura 2-2- Exemplo de barramento I <sup>2</sup> C .....	20
Figura 2-3- Diagrama de sinais do protocolo I <sup>2</sup> C .....	21
Figura 2-4- Topologia de redes 802.15.4 .....	25
Figura 2-5- Tipos de antenas dos módulos XBee .....	25
Figura 2-6 – Diagrama de pinos do XBee®/XBee-PRO® RF .....	26
Figura 3-1- Esquema de ligação do painel de instrumentos .....	27
Figura 3-2 - Esquema de ligação do XBee transmissor e do receptor.....	28
Figura 3-3 – Código para o envio de dados via I <sup>2</sup> C.....	29
Figura 3-4 - Código para o recebimento de dados via I <sup>2</sup> C .....	30
Figura 3-5 - Código para o envio de dados via RS232.....	31
Figura 3-6 - Esquema de ligação do XBee com o microcontrolador .....	32
Figura 3-7 - Tela do supervisor desenvolvido para o Inconfidentes Baja.....	34
Figura 3-8 - Simulação do projeto no software Isis 8.2 Professional.....	34
Figura 3-9 - Simulação dos sensores no Isis 8.2 Professional.....	35
Figura 3-10 - Simulação da comunicação I <sup>2</sup> C no Isis 8.2 Professional.....	35
Figura 3-11 - Simulação do XBee transmissor no Isis 8.2 Professional .....	36
Figura 3-12 - Projeto geral em modo <i>run</i> .....	36
Figura 3-13 - Emulador de porta serial.....	37
Figura 3-14 -Layout da placa de circuito Mestre .....	38

Figura 3-15 - Layout da placa de circuito Escravo.....	39
Figura 4-1 - Projeto geral em modo run no software ISIS .....	40
Figura 4-2 - Sistema em funcionamento mostrando 81% da carga da bateria através dos displays .....	41
Figura 4-3 - Sistema em funcionamento mostrando a velocidade a 27 km/h através dos displays .....	41
Figura 4-4 - Sistema supervisorio em funcionamento.....	42

## SUMÁRIO

1	INTRODUÇÃO .....	14
1.1	Objetivo geral.....	15
1.2	Objetivos específicos .....	15
1.3	Justificativa do trabalho .....	15
1.4	Estrutura do trabalho .....	15
2	SISTEMA SUPERVISÓRIO E DE TELEMETRIA .....	16
2.1	Definição.....	16
2.2	Microcontrolador PIC16F877A .....	16
2.3	Comunicação Serial – I <sup>2</sup> C .....	19
2.3.1	Comunicação Serial – USART .....	21
2.3.2	Modo assíncrono.....	22
2.3.3	Modo Síncrono .....	23
2.4	Sistema de Supervisório.....	23
2.5	Sistema de Telemetria.....	24
3	METODOLOGIA .....	27
3.1	Comunicação I <sup>2</sup> C.....	28
3.2	Supervisório .....	32
3.3	Simulação do Projeto .....	34
3.4	Placas de circuito impresso do projeto.....	37
4	RESULTADOS E DISCUSSÃO .....	40
5	CONCLUSÃO .....	43
6	REFERÊNCIAS BIBLIOGRÁFICAS.....	44

## 1 INTRODUÇÃO

Para otimizar o desempenho de um carro de corridas, uma gama de informações sobre o veículo deve ser medida e analisada. Em um carro de Fórmula Profissional, por exemplo, centenas de sensores são incorporados, o que o torna um dos objetos mais fortemente instrumentados do mundo (WALDO, 2005). Os dados dos sensores são constantemente obtidos e transmitidos, via radiofrequência, ao box, para que a equipe possa analisá-los em tempo real.

O programa Baja SAE BRASIL é um desafio lançado aos estudantes de Engenharia que oferece a chance de aplicar na prática os conhecimentos adquiridos em sala de aula. Eles são envolvidos em um caso real de desenvolvimento de um veículo *off road*, desde sua concepção, projeto detalhado, construção e testes. Os alunos que participam do Baja SAE BRASIL devem formar equipes que representarão a Instituição de Ensino Superior à qual estão ligados. Essas equipes são desafiadas anualmente a participar da competição, que reúne os estudantes e promove a avaliação comparativa dos projetos. No Brasil a competição nacional recebe o nome de Competição Baja SAE BRASIL e as competições regionais são nomeadas como Etapa Sul, Sudeste e Nordeste. Os vencedores da etapa nacional ganham o direito de competir na prova internacional nos Estados Unidos (SAE BRASIL, 2017).

No final de 2011, levados pela proposta e pelo desafio da Competição Baja, estudantes de Engenharia da Universidade Federal de Ouro Preto fundaram a equipe Baja UFOP, hoje denominada Inconfidentes Baja. A equipe é dividida em vários setores, e um deles é a subequipe do sistema eletroeletrônico. O projeto eletroeletrônico do veículo possibilita ao piloto visualizar a velocidade, a distância percorrida e os níveis de combustível e bateria, fazendo com que ele se sinta seguro durante a competição.

Este trabalho tem como objetivo apresentar o desenvolvimento de um sistema supervisor e de telemetria via XBee para o veículo *off road* da equipe Inconfidentes Baja, de modo que as informações do painel de instrumento possam ser monitoradas remotamente. Com isso, a equipe poderá analisar, diagnosticar ou prevenir problemas de forma antecipada, o que pode proporcionar melhor desempenho durante as competições.

### 1.1 Objetivo geral

Desenvolver sistema de telemetria e supervisor do Inconfidentes Baja de modo que a equipe possa analisar no boxe os dados do painel de instrumentos em tempo real.

### 1.2 Objetivos específicos

- Desenvolver sistema de telemetria utilizando XBee;
- Desenvolver supervisor utilizando Visual Studio®;
- Integrar instrumentos do veículo para obter um sistema otimizado;
- Calibrar instrumento.

### 1.3 Justificativa do trabalho

O projeto Baja SAE da Universidade Federal de Ouro Preto - UFOP, que é formado por estudantes, tem como objetivo projetar e construir um Baja, que é um carro do tipo *buggy* para competição *off road* com âmbitos regional, nacional e internacional. O sistema eletroeletrônico do veículo possibilita ao piloto visualizar a velocidade, a distância percorrida e os níveis de combustível e bateria. Para um melhor monitoramento do processo será desenvolvido neste trabalho um sistema de telemetria e supervisor. Os dados do sistema eletroeletrônico serão transmitidos para um servidor por telemetria onde serão tratados e disponibilizados para o seu acompanhamento via supervisor, instalado em um computador no boxe. Dessa forma, os integrantes da equipe poderão acompanhar os problemas do Baja em tempo real, como por exemplo uma pane seca ou elétrica. Assim, alguns problemas poderão ser solucionados mais rapidamente, o que pode melhorar o desempenho da equipe na competição. Além disso, este sistema pode possibilitar uma pontuação melhor no projeto da categoria eletroeletrônica da equipe.

### 1.4 Estrutura do trabalho

No primeiro capítulo faz-se uma introdução do trabalho ressaltando a sua importância. O segundo possui uma revisão de literatura do tema. O terceiro trata a respeito do sistema de telemetria e supervisor do Inconfidentes Baja e a metodologia utilizada. No quarto capítulo, são apresentados os resultados do projeto. No quinto, faz-se a conclusão do trabalho através dos resultados obtidos durante o estudo e as sugestões para trabalhos futuros deste projeto.

## 2 SISTEMA SUPERVISÓRIO E DE TELEMETRIA

Nas seções subsequentes, será apresentada uma breve descrição sobre o sistema supervisório e de telemetria e cada um dos componentes do sistema desenvolvido neste trabalho.

### 2.1 Definição

A medição e análise de dados a distância sem fio chama-se telemetria. Etimologicamente, a palavra telemetria se originou a partir do grego *tele*, que significa “remoto”, e *metron*, que quer dizer “medida” (VISSOTTO JUNIOR 2004 apud SALLES et al., 2016).

Para visualização e armazenamento dos dados usa-se um sistema supervisório. A combinação da telemetria e do supervisório compõem, neste caso, o sistema SCADA (Supervisory Control and Data Acquisition) (BAILEY; WRIGHT 2003 apud CURVÊLLO, 2008, p. 22).

Conforme Boyer (2004), SCADA é uma tecnologia que permite ao usuário localizado em uma estação mestre coletar dados de uma ou mais instalações remotas e enviar instruções de controle pré-definidas para as instalações remotas (apud CURVÊLLO, 2008, p. 22).

Reduções de consumo de combustível e de pneus, bem como da frequência de manutenção e do número de acidentes são algumas das vantagens do uso de sistemas SCADA em veículos de competição (TEIXEIRA; TOURNIER, 2015).

### 2.2 Microcontrolador PIC16F877A

Segundo Souza (2000), pode-se definir o microcontrolador como um “pequeno” componente eletrônico, dotado de “inteligência” programável, utilizado no controle de processos lógicos.

O controle de processos deve ser entendido como o controle de periféricos, tais como: led's, botões, display's de sete segmentos, display's de cristal líquido (LDC), resistência, relês, sensores diversos (pressão, temperatura, etc.) e muitos outros. São chamados de controles lógicos, pois a operação do sistema baseia-se nas ações lógicas que devem ser executadas, dependendo do estado dos periféricos de entrada e/ou saída (SOUZA, 2000, p. 3).

Os microcontroladores PIC são uma família de dispositivos fabricados pela Microchip. Utilizando uma arquitetura RISC, com frequências de *clock* de até 40Mhz, até 2048k *word* de memória de programa e até 3968 bytes de memória RAM. Além disso, podem ser encontrados com diversos periféricos internos, como: até quatro temporizadores/contadores, memória EEPROM interna,

gerador/comparador/amostrador PWM, conversores A/D de até 12 bits, interface de barramento CAN, I<sup>2</sup>C, SPI, entre outros (PEREIRA, 2002, p. 35).

Conforme a Microchip (2003), as características do microcontrolador PIC16F877A são:

- Arquitetura RISC (set de instruções reduzidas);
- Velocidade de Operação: até 20Mhz;
- Memória de Programa (Flash): 8K x 14 *words*;
- Memória de Dados (RAM): 368 x 8 bytes;
- Memória EEPROM: 256 x 8 bytes;
- 8 níveis de STACK;
- 14 tipos de interrupções;
- WDT (*WatchDogTimer*);
- Proteção do Código;
- *In-Circuit Serial Programming* (ICSP) por 2 pinos;
- Opção de selecionar o oscilador;
- Ampla faixa de tensão de Operação: 2,0V à 5,5V;
- Capacidade de corrente por pino: 25mA;
- Timer0: 8-bit *timer/conter*;
- Timer1: 16-bit *timer/conter*;
- Timer2: 8-bit *timer/conter*;
- Modo PWM (10-bit de resolução);
- 8 canais analógicos (10-bit de resolução);
- Modo SSP (*Synchronous Serial Port*) com protocolos SPI e I2C;
- Modo de Comunicação USART;
- Comunicação Paralela;
- Ampla faixa de tensão de operação (2,0V a 5,5V);
- Faixa de temperatura comercial e industrial;
- Baixo consumo de energia.

A Figura 2-1 apresenta o diagrama dos pinos do PIC16F877A:

### 40-Pin PDIP

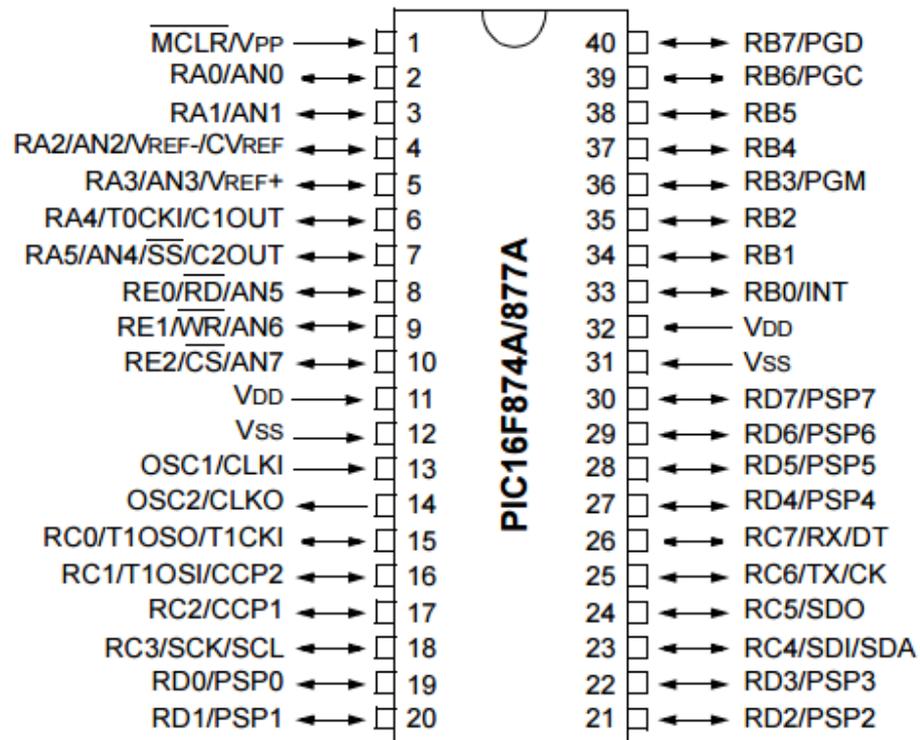


Figura 2-1 Diagrama de pinos do PIC16F877A

Fonte: Microchip (2003)

A descrição dos pinos desse microcontrolador é apresentada na tabela abaixo (Tabela 2-1):

Tabela 2-1 - Descrição dos pinos do PIC16F877A

Fonte: Microchip (2003)

Nome do pino	Número do pino	Tipo I/O/P	Descrição
MCLR/VPP	1	I/P	Master <i>Clear</i> (entrada) ou tensão de programação (saída).
RA0/AN0	2	I/O e I	I/O Digital e Entrada Analógica 0.
RA1/AN1	3	I/O e I	I/O Digital e Entrada Analógica 1.
RA2/AN2/VREF-/CVREF	4	I/O; I; I e O	I/O Digital; Entrada Analógica 2; Entrada de tensão (baixa) de referência A/D e Comparador VREF de saída.
RA3/A3/VREF+	5	I/O; I e I	I/O Digital; Entrada Analógica 3; Entrada de tensão (alta) de referência A/D.
RA4/T0CKI/C1OUT	6	I/O; I e O	I/O Digital; Entrada do Timer 0 e Saída do Comparador 1.
RA5/AN4/SS/C2OUT	7	I/O; I; I e O	I/O Digital; Entrada Analógica 4; Entrada SPI escrava e Saída do Comparador 2.
RB0/INT	33	I/O e I	I/O Digital e Interrupção externa.
RB1	34	I/O	I/O Digital.
RB2	35	I/O	I/O Digital.
RB3/PGM	36	I/O e I	I/O Digital e pino que habilita a programação do ICSP em baixa tensão.

RB4	37	I/O	I/O Digital.
RB5	38	I/O	I/O Digital.
RB6/PGC	39	I/O e I	I/O Digital Programação de <i>Clock</i> do ICSP.
RB7/PGD	40	I/O e I/O	I/O Digital Programação de dados do ICSP.
RC0/T1OSO/T1CKI	15	I/O; O e I	I/O Digital; Saída do Timer 1 e Entrada do <i>clock</i> externo do Timer 1.
RC1/T1OSI/CCP2	16	I/O; I e I/O	I/O Digital; Entrada do Timer 1 e Entrada do Capture 2, Saída do Compare2, Saída do PWM2.
RC2/CCP1	17	I/O e I/O	I/O Digital; Entrada do Capture1, Saída do Compare1, Saída do PWM1.
RC3/SCK/SCL	18	I/O; I/O e I/O	I/O Digital; Entrada/Saída Síncrona Serial do <i>Clock</i> para o modo SPI e Entrada/Saída Síncrona Serial do <i>Clock</i> para o modo I <sup>2</sup> C.
RC4/SDI/DAS	23	I/O; I e I/O	I/O Digital; Dados de entrada do SPI e Dados I/O do I <sup>2</sup> C.
RC5/SDO	24	I/O; O	I/O Digital; Dados de saída do SPI.
RC6/TX/CK	25	I/O; O e I/O	I/O Digital; Transmissão assíncrona USART e <i>Clock</i> síncrono USART1.
RC7/RX/DT	26	I/O; I e I/O	I/O Digital; Recebimento assíncrono USART e Dados síncronos USART.
RD0/PSP0	19	I/O e I/O	I/O Digital e Dados da Porta escrava paralela.
RD1/PSP1	20	I/O e I/O	I/O Digital e Dados da Porta escrava paralela.
RD2/PSP2	21	I/O e I/O	I/O Digital e Dados da Porta escrava paralela.
RD3/PSP3	22	I/O e I/O	I/O Digital e Dados da Porta escrava paralela.
RD4/PSP4	27	I/O e I/O	I/O Digital e Dados da Porta escrava paralela.
RD5/PSP5	28	I/O e I/O	I/O Digital e Dados da Porta escrava paralela.
RD6/PSP6	29	I/O e I/O	I/O Digital e Dados da Porta escrava paralela.
RD7/PSP7	30	I/O e I/O	I/O Digital e Dados da Porta escrava paralela.
RE0/RD/AN5	8	I/O; I e I	I/O Digital; Controle de leitura para Porta Escravo Paralelo e Entrada Analógica 5.
RE1/WR/AN6	9	I/O; I e I	I/O Digital; Controle de escrita para Porta Escravo Paralelo e Entrada Analógica 6.
RE2/CS/AN7	10	I/O; I e I	I/O Digital; Controle de seleção de Chip para a Porta Escrava Paralela e Entrada Analógica 7.
OSC1/CLKI	13	I e I	Cristal Oscilador ou Entrada do <i>clock</i> externo.
OSC2/CLKO	14	O e O	Cristal Oscilador ou Saída do <i>clock</i> .
VSS	12 e 31	P	Referência de terra para os pinos de lógica e I/O.
VDD	11 e 32	P	Fornecimento positivo para pinos de lógica e I/O.

### 2.3 Comunicação Serial – I<sup>2</sup>C

O protocolo de comunicação I<sup>2</sup>C (*Inter-Intergrad Circuit*) foi criado pela Philips® para facilitar o desenvolvimento de aparelhos eletrônicos de consumo geral. Trata-se de um protocolo síncrono de dois fios ou linhas, sendo uma linha de *clock* (chamada de SCL) e outra de dados (chamada de SDA). Esse protocolo permite a ligação de diversos dispositivos nas mesmas linhas, formando um autêntico barramento de comunicação serial, ou uma rede de dispositivos (PEREIRA, 2003, p. 274). A Figura 2-2 apresenta um exemplo do barramento I<sup>2</sup>C:



Figura 2-2- Exemplo de barramento I²C

Fonte: SOHN (2011)

“Como todo protocolo síncrono, o I²C também é do tipo mestre-escravo, no entanto também suporta o chamado *multimastering*, ou seja, a presença de diversos mestres simultaneamente no mesmo barramento” (PEREIRA, 2003, p. 274). Ainda segundo Pereira (2003, p. 275), durante uma comunicação, somente um dos mestres pode estar ativo, ou ocorrerá uma colisão no barramento. O componente mestre será responsável pelo controle de comunicação, gerando o *clock* de sincronismo e o escravo ficará sempre aguardando o *clock* enviado pelo mestre (SOUZA, 2003, p. 198). Tanto o mestre quanto o escravo podem transmitir ou receber dados, mas o controle é sempre do mestre (SOUZA, 2003, p. 203).

Para evitar conflitos na via de dados (SDA), os pinos são chaveados como entrada ou saída (impõe somente o nível baixo, forçando o GND), conforme a necessidade imposta pelo protocolo. Por esse fato, um resistor de *pull-up* é necessário, sendo recomendado valores de 10kΩ a 1kΩ. Os valores mais baixos (até 1kΩ) podem ser necessários para velocidades muito elevadas ou uma rede com muitos periféricos (SOUZA, 2003, p. 203).

Segundo Pereira (2003, p. 275) a característica de funcionamento do protocolo baseia-se em:

- 1) A informação presente na linha de dados (SDA) somente é lida durante a fase alta da linha de *clock* (SCL);
- 2) Somente é permitido alterar o nível da linha de dados (SDA) durante a fase baixa da linha de *clock* (SCL);
- 3) Quando o barramento não está em uso, ambas as linhas permanecem desligadas (portanto forçadas em nível “1” pelos resistores de *pull-up*).

Para sinalizar a condição de início da transmissão ou *START*, o dispositivo força a linha SDA de nível alto para nível baixo, durante a fase alta de *clock*. Esta violação indica aos dispositivos que uma transmissão terá início (PEREIRA, 2003, p. 276).

Para sinalizar o fim de uma transmissão, chamada de condição de parada ou *STOP*, é feita a transição de nível baixo para nível alto da linha SDA durante a fase alta da linha SCL (PEREIRA, 2003, p. 276). A Figura 2-3 apresenta o diagrama de sinais do protocolo I<sup>2</sup>C:

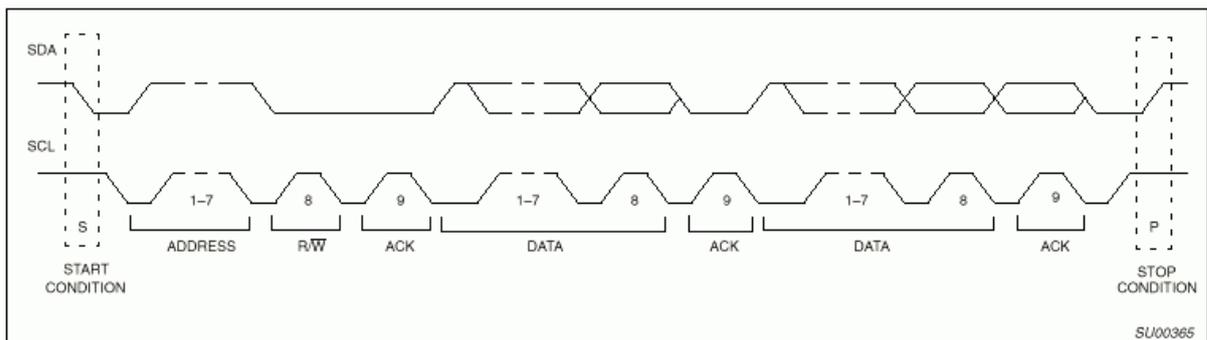


Figura 2-3- Diagrama de sinais do protocolo I<sup>2</sup>C

Fonte: Firegard (2017)

“Após o bit de *START*, são transmitidos oito bits de dados, iniciando pelo MSB. Após o último bit (LSB) o receptor deve gerar uma condição de reconhecimento (*acknowledge*), o que é feito forçando a linha SDA em nível “0” antes do nono pulso de *clock* da linha SCL.” (PEREIRA, 2003, p.276)

“Caso o receptor não reconheça o dado (mantendo a linha SDA em “1” durante o nono pulso da linha SCL), o transmissor deve abortar (gerando uma condição de parada) e reiniciar a transmissão.” (PEREIRA, 2003, p.276)

Atualmente o protocolo suporta velocidades de até 3,4 Megabits por segundo. Mas na realidade, a grande maioria dos dispositivos I<sup>2</sup>C é limitada a velocidades de 100 ou 400 Kbps (PEREIRA, 2003, p. 275).

### 2.3.1 Comunicação Serial – USART

O nome USART significa *Universal Synchronous Asynchronous Receiver Transmitter*. É um protocolo universal, que também é conhecido como SCI (*Serial Communications Interface*) e possui dois modos distintos de trabalho: o sincronizado e o não-sincronizado (SOUZA, 2003, p. 243).

### 2.3.2 Modo assíncrono

A comunicação é feita somente com duas vias; entretanto, como este não é um modo sincronizado, essas duas vias são utilizadas para dados. Uma delas para transmissão (TX) e a outra para recepção (RX). Isso possibilita que as informações sejam enviadas e recebidas ao mesmo tempo, cada qual na sua via. Este recurso é conhecido como *Full Duplex*. Esse modo é utilizado, por exemplo, na porta serial dos computadores para implementar o padrão RS-232, mas pode ser utilizado para acesso a outros sistemas também (SOUZA, 2003, p. 243).

Segundo Souza (2003, p. 244) a sincronização dos dados na transmissão é feita através do *Baud Rate* ou velocidade de transmissão, normalmente indicada em bits por segundo, ou bps. Para que o sistema funcione, o tamanho dos dados (intervalo de cada bit) deve ser completamente padronizado, e ambos os lados devem ser ajustados para o mesmo valor. O tempo de duração de cada bit é:

$$T_{BIT} = \frac{1}{BaudRate}$$

“Com isso, existe somente um sincronismo de tempo feito para transmissão/recepção de cada byte. Esse sincronismo é conseguido através do *Start bit*.” (SOUZA, 2003, p. 244)

As vias possuem seu estado padrão como sendo nível alto. Temos, então, uma situação de *stand-by*. Quando um lado inicia uma transmissão, ele força seu TX para nível baixo, mantendo-o assim pelo tempo  $T_{BIT}$ . Essa borda de descida é reconhecida pelo outro lado (em RX) e é suficiente para iniciar o processo de sincronização para recebimento desse byte. Esse pulso em nível baixo é chamado de *Start Bit* (SOUZA, 2003, p.244).

Depois disso, os dois lados já sabem o que fazer. TX enviará então os 8 bits de dados, todos eles com o mesmo tamanho do *Start Bit*. Como RX soube exatamente o momento de início do *Start Bit*, ele deixa passar o tempo e depois coleta os 8 bits, pegando o dado mais ou menos no meio do tempo do bit ( $T_{BIT}/2$ ) (SOUZA, 2003, p.244).

Por último, para garantir o processo, TX envia um *Stop Bit*, que nada mais é que outro bit com valor fixo em 1, garantindo assim que a linha voltará ao seu estado padrão e o sistema voltará ao *stand-by*, ficando apto ao próximo dado. O lado RX deve considerar a leitura do *Stop Bit* para garantir que nenhum erro grosseiro aconteça com a recepção. Caso o *Stop Bit* seja 0 (zero), pode ter acontecido um erro de temporização, e no seu lugar foi lido o bit 8 ou, então o próximo *Start* (SOUZA, 2003, p.244).

### 2.3.3 Modo Síncrono

Essa comunicação utiliza o mesmo conceito do padrão I<sup>2</sup>C, chamada de *Half Duplex* cujo é um modelo que trabalha-se com somente duas vias. Uma é destinada ao *clock* (CK) e a outra aos dados (DT). Desta forma, os dados devem trafegar em uma única via, impossibilitando a transmissão e recepção simultâneas. Essa comunicação pode ser utilizada para a troca de dados com outros microcontroladores ou diversos periféricos existindo no mercado, tais como A/D's, D/A's, memórias e etc. (SOUZA, 2003, p.245)

Esse modo opera com Mestre e Escravo, sendo o *clock* sempre gerenciado pelo Mestre. Para o Mestre, a via CK é sempre uma saída e para o Escravo ela é sempre uma entrada. Quanto a via de dados, ela muda constantemente de sentido, hora para transmissão, hora para recepção. Assim sendo, para qualquer uma das pontas, a via DT é saída para a transmissão e entrada para recepção (SOUZA, 2003, p.245).

“O tempo de duração de um bit também define o *Baud Rate*, do mesmo modo descrito na comunicação assíncrona.” (SOUZA, 2003, p.245)

“Neste caso, o nível baixo (0) é o padrão para a via CK. Para a via DT, não existe um padrão obrigatório, pois ele não opera sem CK. Porém, recomendamos mantê-la também em nível baixo quando não está sendo usada.” (SOUZA, 2003, p.245)

“Neste padrão, também o bit menos significativo (bit 0) é enviado primeiro.” (SOUZA, 2003, p.245)

## 2.4 Sistema de Supervisório

Segundo Barreto, Miyage e Silva (1993, p.87) os sistemas supervisórios são utilizados visando monitorar e controlar, em tempo real, um sistema já implementado, permitindo ao usuário a visualização ou até mesmo a mudança de parâmetros de controle (apud ZIRZANOWSKY, 2007, p. 18). Conforme Araújo (2000, p. 11) eles visam facilitar o acompanhamento dos mais variados processos, através da leitura de dados destes processos através de sensores. Estes dados são disponibilizados aos usuários através de interfaces (apud ZIRZANOWSKY, 2007, p. 17).

“Os componentes físicos de um sistema de supervisão podem ser resumidos em: sensores e atuadores, redes de comunicação, estações remotas (aquisição / controle) e de monitoração central (software computacional)” (SALVADOR; SILVA, 2017).

Os sensores são dispositivos acoplados aos equipamentos controlados e monitorados pelo sistema supervisor, que convertem o valor de grandezas físicas tais como velocidade, luminosidade e temperatura, para sinais analógicos e digitais interpretáveis pela estação remota. Os atuadores são utilizados para atuar sobre o sistema, acionando e desligando equipamentos. O processo de controle e aquisição de dados se inicia nas estações remotas com a leitura dos valores atuais dos dispositivos que a elas estão associadas e seu respectivo controle (SALVADOR; SILVA, 2005, p.2 apud ZIRZANOWSKY, 2007, p. 18).

A rede de comunicação é a plataforma por onde as informações fluem das estações remotas para a estação de monitoração central. Levando em consideração os requisitos do sistema e a distância a cobrir, pode ser implementada através de cabos par traçado, fibras ópticas, linhas *dial-up*, linhas dedicadas, radiofrequência, etc. As estações de monitoração central são as unidades principais dos sistemas supervisor, sendo responsáveis por recolher a informação gerada pelas estações remotas e agir em conformidade com os eventos detectados, podendo ser centralizadas num único computador ou distribuídos por uma rede de computadores, de modo a permitir o compartilhamento das informações coletadas (SALVADOR; SILVA, 2005, p.2 apud ZIRZANOWSKY, 2007, p. 18).

## 2.5 Sistema de Telemetria

A transmissão de dados entre as unidades de um sistema a ser supervisionado pode ser feita via rede de computadores fixa ou sem fio. Alguns exemplos dessas tecnologias são: RS-232, RS-442, RS-485, linhas telefônicas (discadas ou dedicadas), radiofrequência, micro-ondas, satélite e redes padrão Ethernet (ADAMI JUNIOR, 2006 apud CURVÊLLO, 2008, p. 23).

A criação de redes de sensores sem fio é uma tecnologia emergente que tem uma ampla gama de aplicações importantes, incluindo a vigilância do meio ambiente, os espaços inteligentes, os sistemas médicos e a exploração robótica (YE et al., 2002 apud ROMERO et al., 2015).

O módulo XBee é um dispositivo usado para se comunicar via rede sem fio, por meio do padrão IEEE 802.15.4. Ele possui uma lista de recursos, como o controle de acesso ao meio e a detecção de erro para garantir a entrega e integridade de dados (HUSSIN, 2012). Essa rede oferece excelente imunidade a interferências, capacidade de hospedar milhares de dispositivos (mais que 65.000), e possuem taxas de transferências de dados entre 20 kbps e 250 kbps (BARONTI et al., 2007 apud SANT'ANA et al., 2004).

Há três tipos de topologias do padrão 802.15.4: árvore, estrela e malha (Figura 2-4) (SALEIRO, 2009).

Uma das principais vantagens dos módulos XBee é o fato de suportar redes em malha, onde, havendo vários caminhos possíveis, é possível eliminar falhas de comunicação no caso da falha de um nó da rede, pois os nós intermediários redirecionam a informação até que ela chegue ao destinatário (SALEIRO, 2009).

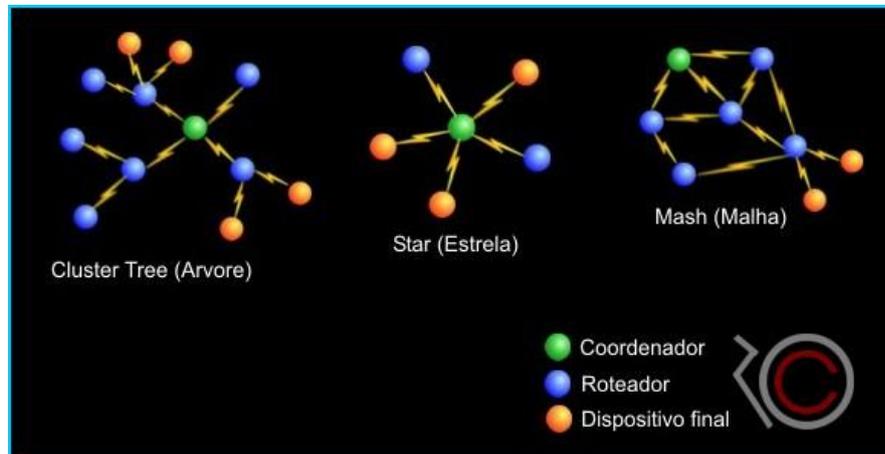


Figura 2-4- Topologia de redes 802.15.4

Fonte: (MESSIAS, 2015)

Nos módulos XBee/XBee-Pro™ há três opções de antenas (Figura 2-5). Os tipos são: chicote (um pedaço de fio de ~2,5 cm), conector (para antena externa) e chip, sendo esta a mais compacta. Com o dos tipos chicote e externa é possível direcionar o feixe de sinal e, assim, melhorar a performance da rede (MESSIAS, 2015). Cada antena se adapta melhor a um tipo de aplicação.

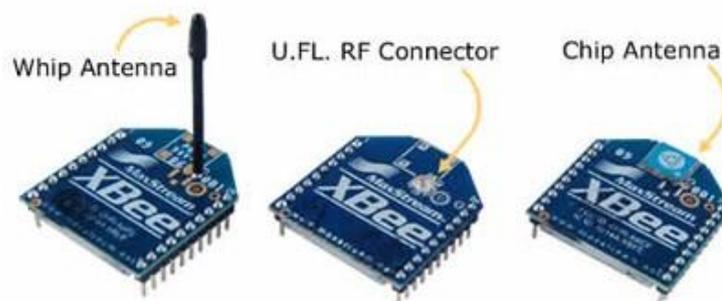


Figura 2-5- Tipos de antenas dos módulos XBee

Fonte: (MESSIAS, 2015)

O XBee pode ser conectado a um computador por meio do módulo UART 5 PIN-CP2102.

O diagrama de pinos do XBee®/XBee-PRO® RF está descrito na Figura 2-6.

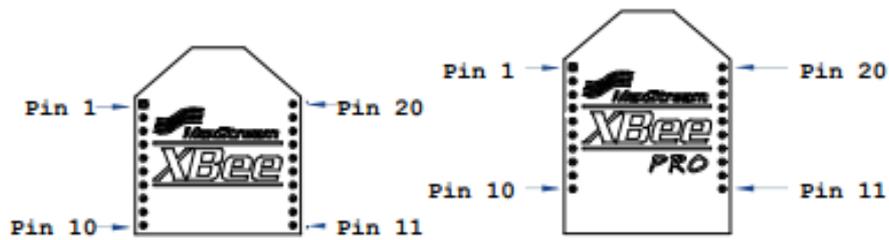


Figura 2-6 – Diagrama de pinos do XBee®/XBee-PRO® RF

Fonte: (DIGI INTERNATIONAL INC, 2009)

Algumas características do XBee- Pro® são:

- Alcance Indoor: até 30 m;
- Alcance Outdoor: até 1600 m;
- Taxa de dados do RF: 250.000 bps;
- Taxa de dados da interface serial: 1200 bps – 250 kbps;
- Tensão de Alimentação: 2.8 – 3.4 V;
- Ligações mínimas: VCC (pino 1), GND (pino 10), DOUT (pino 2) e DIN (pino 3);
- Conexões mínimas para atualização de firmware: VCC, GND, DIN, DOUT, RTS e DTR (DIGI INTERNATIONAL INC, 2009).

### 3 METODOLOGIA

Para adquirir as informações do painel de instrumentos do Inconfidentes Baja foram utilizados dois microcontroladores PIC16F877A, que se comunicam em topologia de rede mestre e escravo através da comunicação I<sup>2</sup>C. O microcontrolador mestre possui a função de coletar os dados dos sensores de velocidade, nível de combustível e nível de bateria. Já o microcontrolador escravo possui a função de centralizar as informações recebidas do escravo, enviá-las via radiofrequência ao programa supervisorio e mostrá-las em displays de sete segmentos ao piloto do veículo (Figura 3-1). No sistema de telemetria foi empregado a tecnologia XBee. O XBee transmissor foi conectado ao microcontrolador escravo, que envia as informações do painel de instrumentos para o XBee receptor. Já o XBee receptor foi conectado à porta USB/Serial de um computador para apresentar os dados por meio de um supervisorio (FIG.3-2).

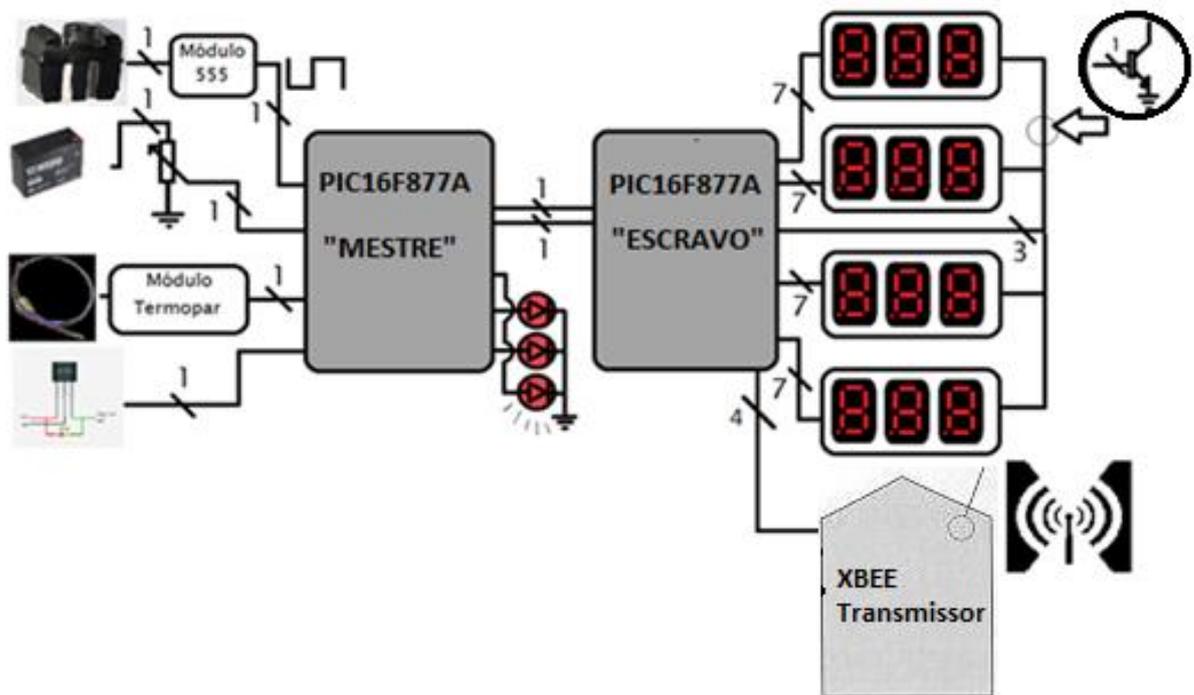


Figura 3-1- Esquema de ligação do painel de instrumentos

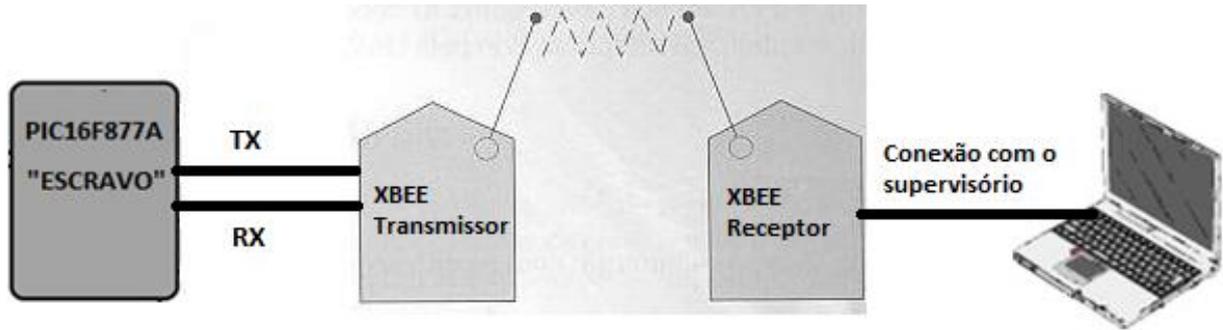


Figura 3-2 - Esquema de ligação do XBee transmissor e do receptor

Em um projeto de sistemas embarcados, os requisitos do sistema devem ser levados em consideração para que a seleção do microcontrolador seja realizada de modo adequado. Nesse sentido, ressalta-se que o superdimensionamento do sistema embarcado em relação à frequência de processamento, ao tamanho das memórias e ao número de periféricos, por exemplo, pode onerar o custo final do projeto.

Neste trabalho, o microcontrolador escravo realiza a multiplexação de 14 displays de sete segmentos, o que demanda 21 portas de saída digital, e estabelece uma comunicação UART e outra I<sup>2</sup>C. Por isso, selecionou-se um dispositivo de 40 pinos.

### 3.1 Comunicação I<sup>2</sup>C

No microcontrolador mestre, para configurar os pinos SDA e SCL usou-se o seguinte comando: `#use I2C(MASTER, SDA=PIN_C4, SLOW, SCL=PIN_C3, NOFORCE_SW);`.

Para o mestre transmitir os dados para o escravo criou-se uma função denominada *void envio\_I2C()* no PIC mestre. Nessa função, depois de inicializada a comunicação I<sup>2</sup>C através do comando *i2c\_start()*, é informado o endereço do dispositivo escravo que irá receber os dados através do comando *i2c\_write(0xa0)*. Em seguida os dados são transmitidos e depois finaliza-se a transmissão. O código está exemplificado pela Figura 3-3:

---

```
1: int8 bat, msbdist=0, lsbdist=0;
2: long int vel, comb;
3: void envio_I2C ()
4: {
5:     i2c_start(); //Inicializa a comunicação I2C
6:     i2c_write(0xa0); //Endereço do dispositivo escravo
7:     delay_ms(100);
8:     i2c_write (bat); // Envia o valor da bateria do veículo
9:     i2c_write ('a'); // Envia dado para checagem
10:    delay_ms(10);
11:    i2c_write (vel); // Envia o valor da velocidade do veículo
12:    i2c_write ('b'); // Envia dado para checagem
13:    delay_ms(10);
14:    i2c_write (msbdist); // Envia bit mais significativo do valor da distância
15:    i2c_write ('c'); // Envia dado para checagem
16:    delay_ms(10);
17:    i2c_write (lsbdist); // Envia bit menos significativo do valor da distância
18:    i2c_write ('d'); // Envia dado para checagem
19:    delay_ms(10);
20:    i2c_write (comb); // Envia o valor do combustível
21:    i2c_stop(); //Finaliza a transmissão
22:    delay_ms(2000)
23: }
24: void main () { ...
```

---

Figura 3-3 – Código para o envio de dados via I<sup>2</sup>C

No comando para a configuração dos pinos SDA e SCL do dispositivo escravo, é informado também o endereço do dispositivo, que é o mesmo destino informado na função “*void envio\_I2C()*” do dispositivo mestre. O comando declarado foi: *#use I2C(SLAVE, SDA=PIN\_C4, SLOW, SCL=PIN\_C3, ADDRESS=0xa0, NOFORCE\_SW);* .

Dentro da função *main()* do escravo foi criada uma condição para a recepção dos dados. O código está descrito na Figura 3-4.

---

```

1. void main()
2. {
3.   while (1)
4.   {
5.     if (i2c_poll())
6.     {
7.       valor[0] = i2c_read(); //Recebe endereço
8.       if (valor[0]==160)
9.       {
10.        valor[1] = i2c_read(); //Recebe valor da velocidade
11.        bat= valor[1];
12.        valor[2] = i2c_read(); //Recebe dado para checagem
13.        if (valor[2]=='a') {...}
36.        printf ("B");
37.        printf (":");
38.        printf ("%02u", bat); //Valor da bateria que será enviado
                                para o supervisorio pela Serial
39.        printf (":");
40.        printf ("%02u", vel); //Valor da velocidade que será enviado
                                para o supervisorio pela Serial
41.        printf (":");
42.        printf ("%03u", msbdist ); //Valor do bit mais significativo que
                                será enviado para o supervisorio pela Serial
43.        printf (":");
44.        printf ("%03u", lsbdist ); //Valor do bit menos significativo que será
                                enviado para o supervisorio pela Serial
45.        printf (":");
46.        printf ("%02u", comb); //Valor do combustível que será enviado para
                                o supervisorio pela Serial
47.        printf (":");
48.      }
49.    }
50. }

```

---

Figura 3-4 - Código para o recebimento de dados via I<sup>2</sup>C

Neste projeto a comunicação entre o escravo e o XBee transmissor foi feita via UART (*Universal Asynchronous Receiver Transmitter*). No escravo fez-se a seguinte declaração para configuração dos pinos de recepção e transmissão de dados: `#use rs232 (baud=9600, xmit=PIN_C6, rcv=PIN_C7);`

Para o envio dos dados do PIC escravo para o XBee transmissor foi desenvolvido o algoritmo descrito na Figura 3-5:

---

```

1. void main()
2. {
3.   while (1)
4.   {
5.     if (i2c_poll())
6.     {
7.       valor[0] = i2c_read(); //Recebe endereço
8.       if (valor[0]==160)
9.       {
10.        valor[1] = i2c_read(); //Recebe valor da velocidade
11.        bat= valor[1];
12.        valor[2] = i2c_read(); //Recebe dado para checagem
13.        if (valor[2]=='a') {...}
36.        printf ("B");
37.        printf (":");
38.        printf ("%02u", bat); //Valor da bateria que será enviado
                                para o supervisorio pela Serial
39.        printf (":");
40.        printf ("%02u", vel); //Valor da velocidade que será enviado
                                para o supervisorio pela Serial
41.        printf (":");
42.        printf ("%03u", msbdist ); //Valor do bit mais significativo que
                                    será enviado para o supervisorio pela Serial
43.        printf (":");
44.        printf ("%03u", lsbdist ); //Valor do bit menos significativo que será
                                    enviado para o supervisorio pela Serial
45.        printf (":");
46.        printf ("%02u", comb); //Valor do combustível que será enviado para
                                    o supervisorio pela Serial
47.        printf (":");
48.      }
49.    }
50. }

```

---

Figura 3-5 - Código para o envio de dados via RS232

A tensão máxima que o pino RX do XBee pode tolerar é 3,3V. Por isso, para fazer a conexão do pino TX do escravo no RX do XBee, fez-se um divisor de tensão com o valor de saída de 3,3V. Já o pino TX do XBee foi ligado de forma direta no RX do escravo, pois o microcontrolador reconhece 3,3V como nível lógico alto.

O XBee receptor foi conectado ao computador por meio de um adaptador UART-USB. Portanto, foi necessário configurar a porta serial de maneira adequada, durante o

desenvolvimento do software supervisor, para que a comunicação fosse estabelecida. Segue abaixo o esquema elétrico feito no ISIS 7.8 Professional (FIG. 3-6):

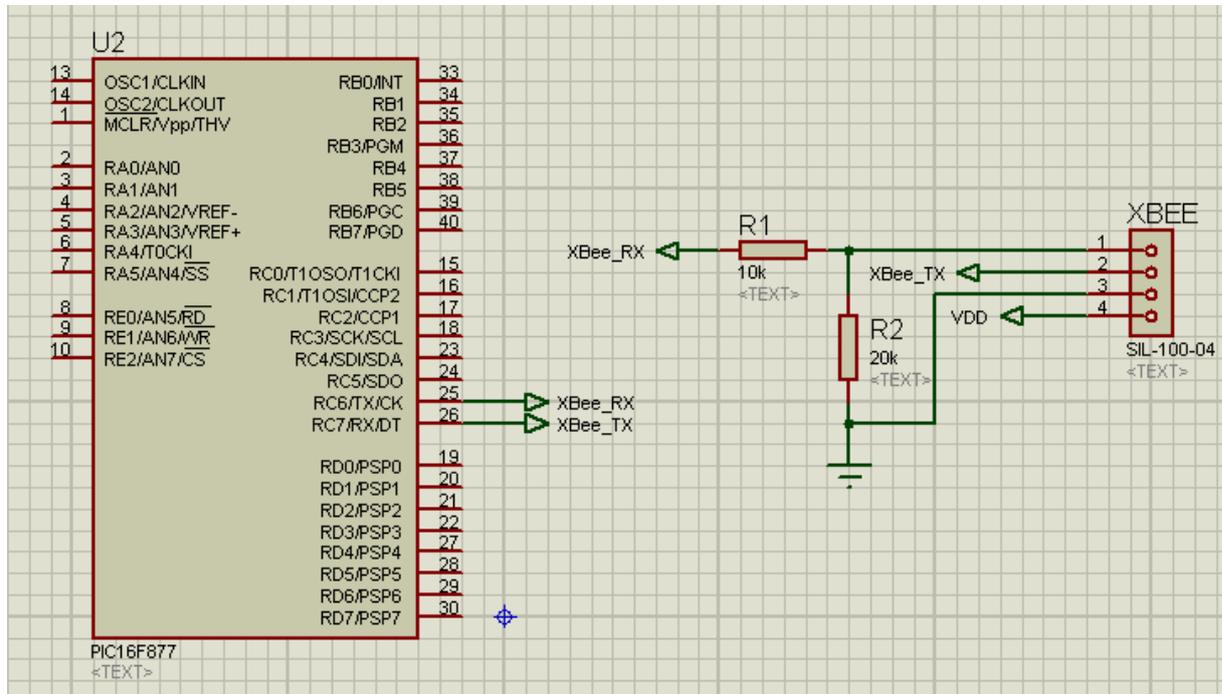


Figura 3-6 - Esquema de ligação do XBee com o microcontrolador

A configuração para o pacote de dados lido no XBee receptor ser reconhecido pelo computador foi feita pelo supervisor.

### 3.2 Supervisor

O software supervisor apresentado neste trabalho foi desenvolvido em linguagem de programação C Sharp, por meio do ambiente de desenvolvimento integrado Visual Studio.

O XBee receptor se comunica com o computador por meio do módulo UART. Logo, para o computador conseguir reconhecer os dados do XBee deve-se trabalhar no supervisor com o módulo serial. O método *SerialPort.ReadExisting()* lê todos os bytes imediatamente disponíveis na porta serial, baseado na codificação, no fluxo e no buffer de entrada do objeto *SerialPort*. Assim, criou-se uma variável para armazenar os dados recebidos pela porta serial, como no exemplo: *InputData = serialPort1.ReadExisting()*.

Neste projeto a variável *InputData* recebe um pacote de dados do tipo *string* contendo o valor da velocidade, do nível de combustível, do nível de bateria e da distância percorrida pelo Inconfidentes Baja em tempo real. Criou-se uma função para que no instante em que chegar o pacote de dados, deve-se dividi-lo em vários vetores de acordo com cada informação, ou seja,

o valor da velocidade seja armazenada no *vetor[0]*, o nível de combustível no *vetor [1]* e assim sucessivamente.

Na tela do supervisor, desenvolvido no Visual Studio, foram configurados três *gauges*, que são elementos gráficos para representar em forma de ponteiros (como se fosse um velocímetro) o valor da velocidade do carro (em quilômetros por hora), bateria do carro (em porcentagem) e nível de combustível do carro (em porcentagem). A tela ainda possui uma *label* que irá mostrar o valor da distância percorrida pelo carro e um *xyGraph* que é um elemento gráfico do tipo linha que mostrará a variação da velocidade pelo tempo. Abaixo deste gráfico, foram configurados dois botões: um denominado iniciar e outro finalizar - responsáveis para iniciar e terminar a geração do gráfico respectivamente.

O usuário deverá selecionar qual porta serial está conectada por meio de um *comboBox*. Após escolher a porta, o usuário deverá apertar no botão “conectar”. Assim, a comunicação será realizada com sucesso, aparecendo na barra de status que o dispositivo está conectado. Quando o usuário quiser desconectar o supervisor basta clicar no botão “desconectar”.

Outro elemento adicionado foi o *DataGridView*, que é um controle utilizado para exibir e editar dados em uma superfície tabular. Os valores dos sensores e os horários das leituras foram organizados neste elemento. Os dados são mostrados a cada três segundos.

Há possibilidade de exibir os dados recebidos e exibidos no *DataGridView* em uma planilha no Excel e também de abrir os arquivos salvos. A Figura 3-7 corresponde ao supervisor desenvolvido no Visual Studio, com seus objetos gráficos inseridos.

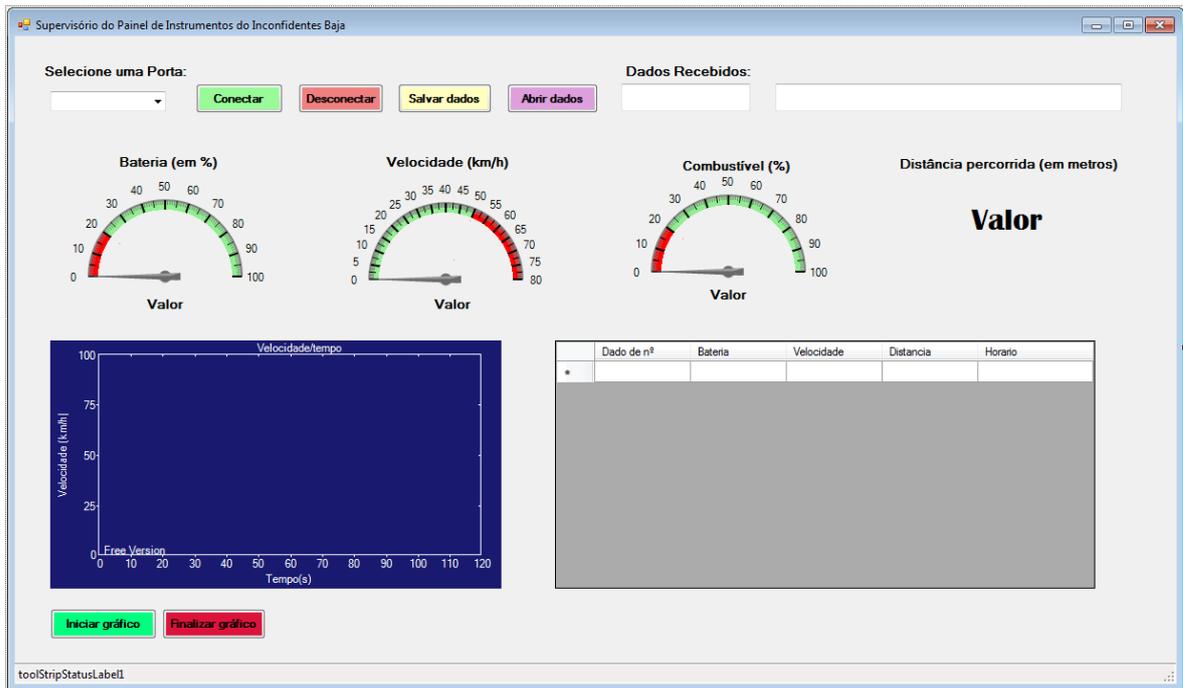


Figura 3-7 - Tela do supervísório desenvolvido para o Inconfidentes Baja

### 3.3 Simulação do Projeto

Para fazer a simulação de todo o projeto (FIG. 3-8) utilizou-se os softwares Proteus e o Eterlogic VSPE.

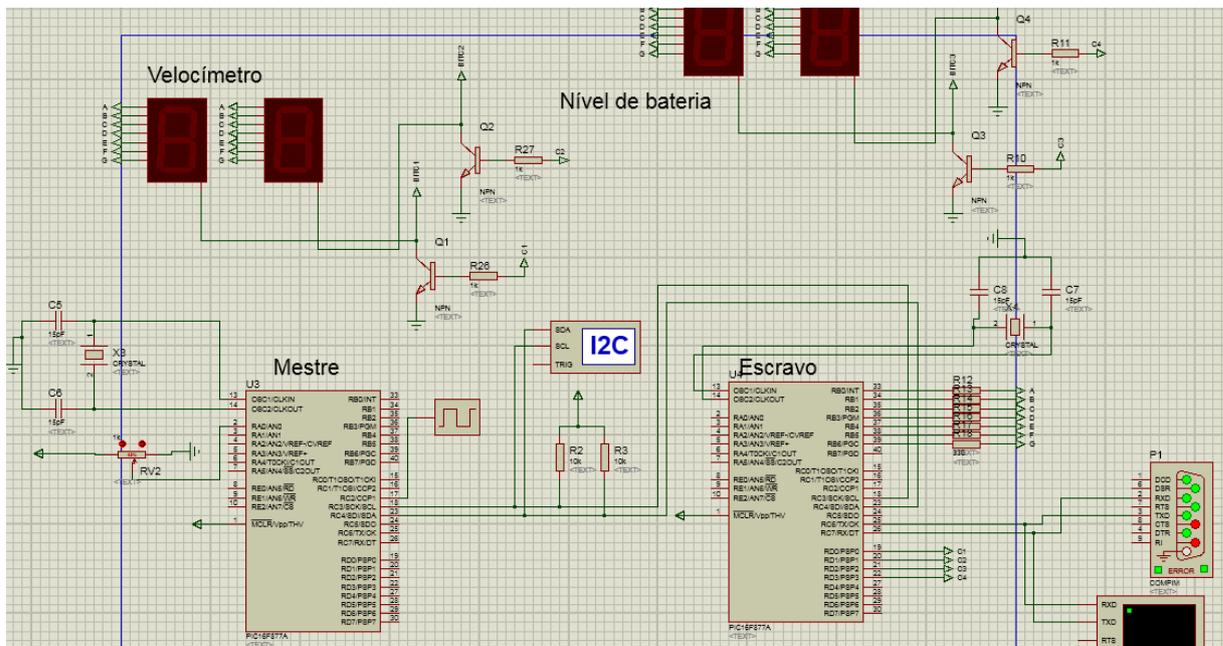


Figura 3-8 - Simulação do projeto no software Isis 8.2 Professional

O primeiro foi utilizado para fazer a simulação dos sensores (FIG 3-9), a comunicação entre os dois microcontroladores (FIG. 3-10) e a simulação do XBee transmissor (FIG. 3-11). Para

simular o XBee transmissor conectou nos pinos TX E RX do microcontrolador o componente COMPIM, que é o objeto que simula uma porta serial no ISIS.

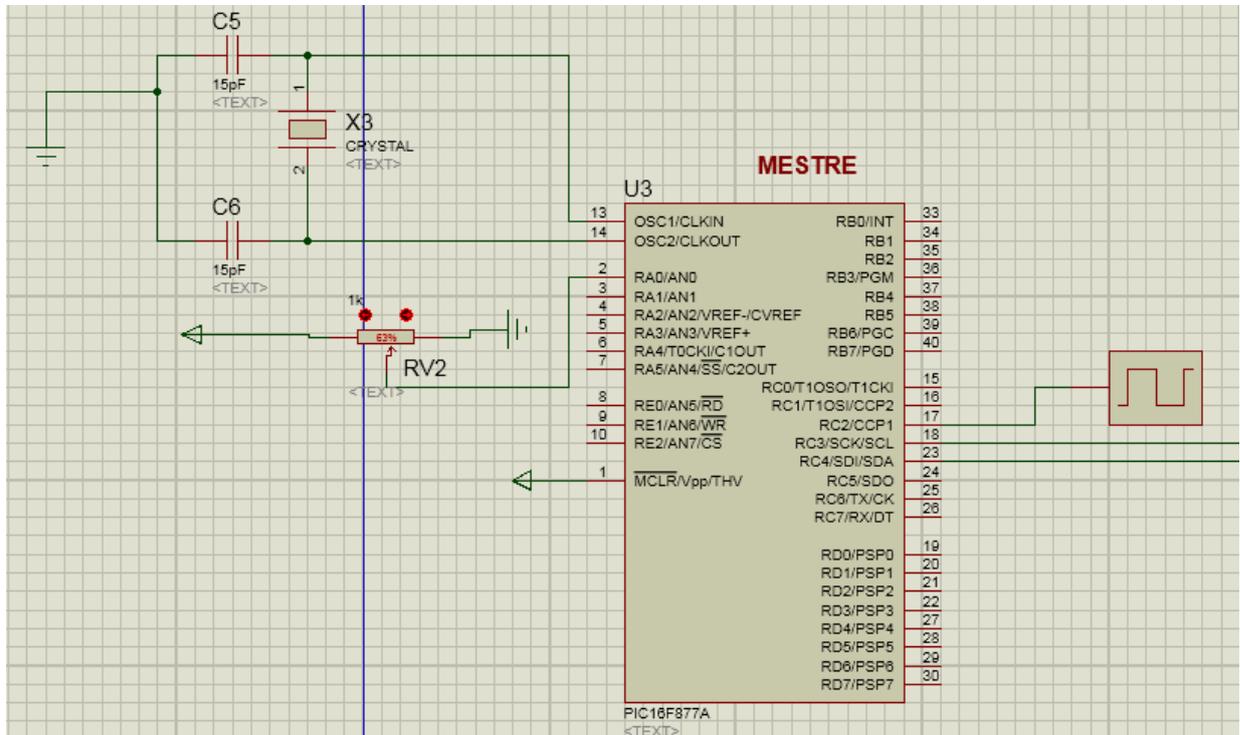


Figura 3-9 - Simulação dos sensores no Isis 8.2 Professional

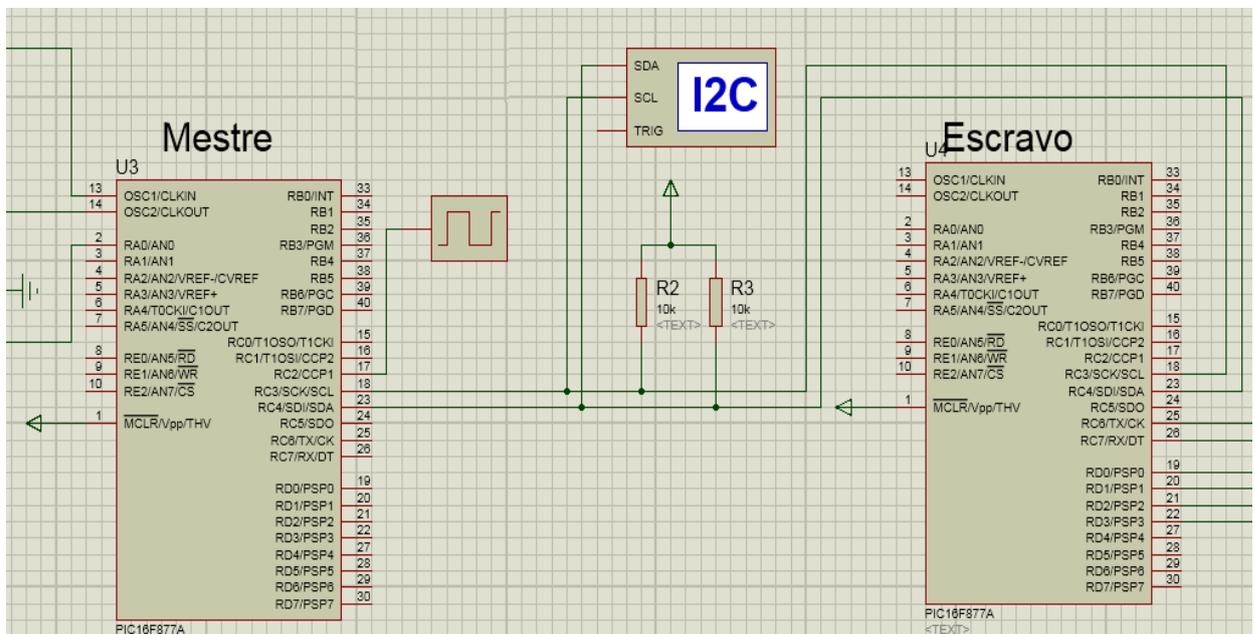


Figura 3-10 - Simulação da comunicação I2C no Isis 8.2 Professional

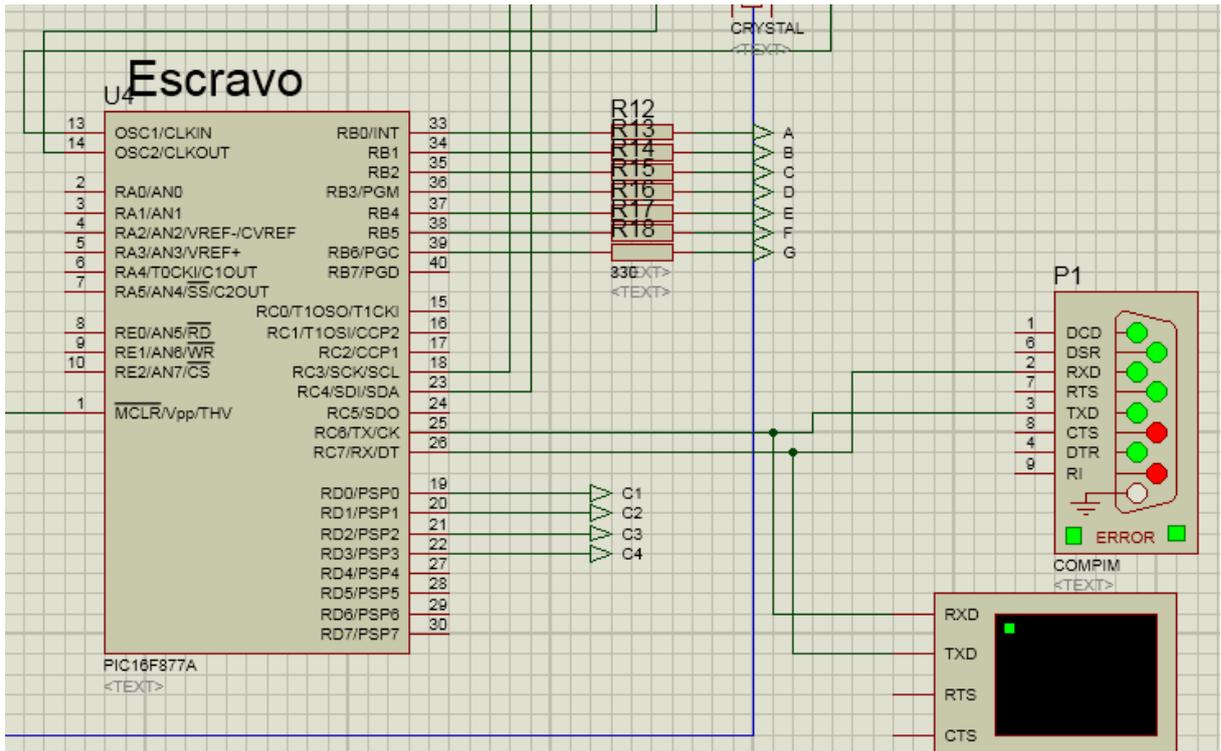


Figura 3-11 - Simulação do Xbee transmissor no Isis 8.2 Professional

A Figura 3-12 ilustra o projeto geral em modo *run*.

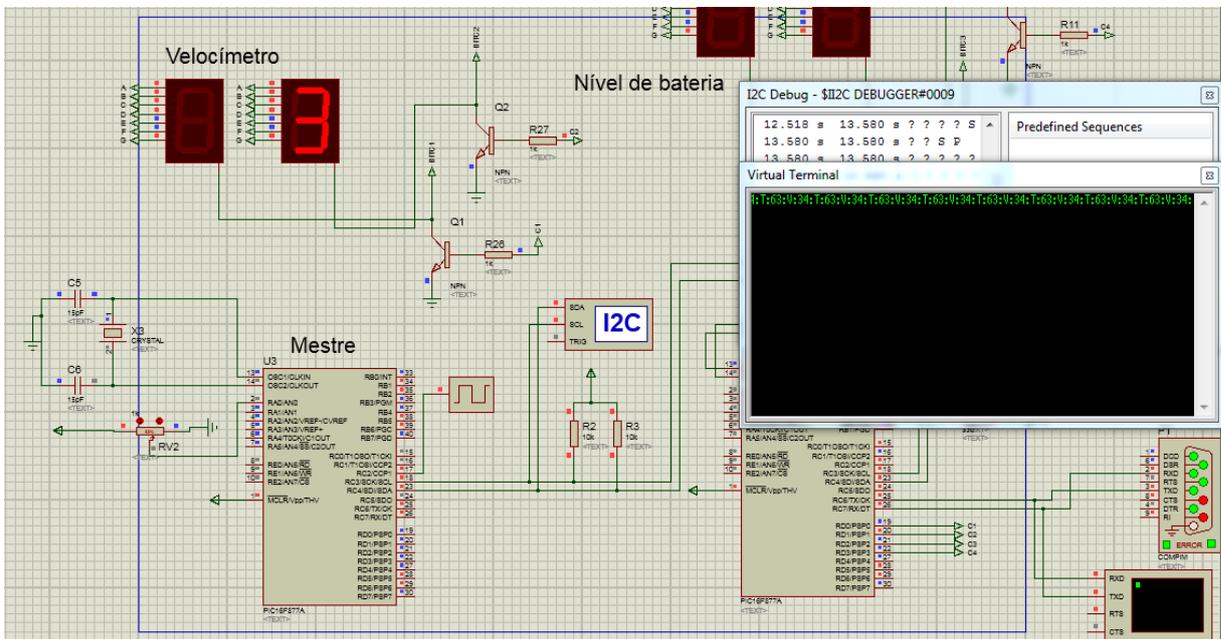


Figura 3-12 - Projeto geral em modo *run*

O segundo programa utilizado para fazer a simulação do projeto tem como funcionalidade testar aplicações que usam portas seriais. Ele é capaz de criar várias portas COM virtuais para

transmitir e receber dados. Com isso, criaram-se duas portas COM comunicando-se, uma para simular o XBee transmissor, através da transferência de dados do COMPIM (configurado como COM3); e outra para simular a conexão do XBee receptor com o supervisor (configurado como COM1), de acordo com a Figura 3-13.

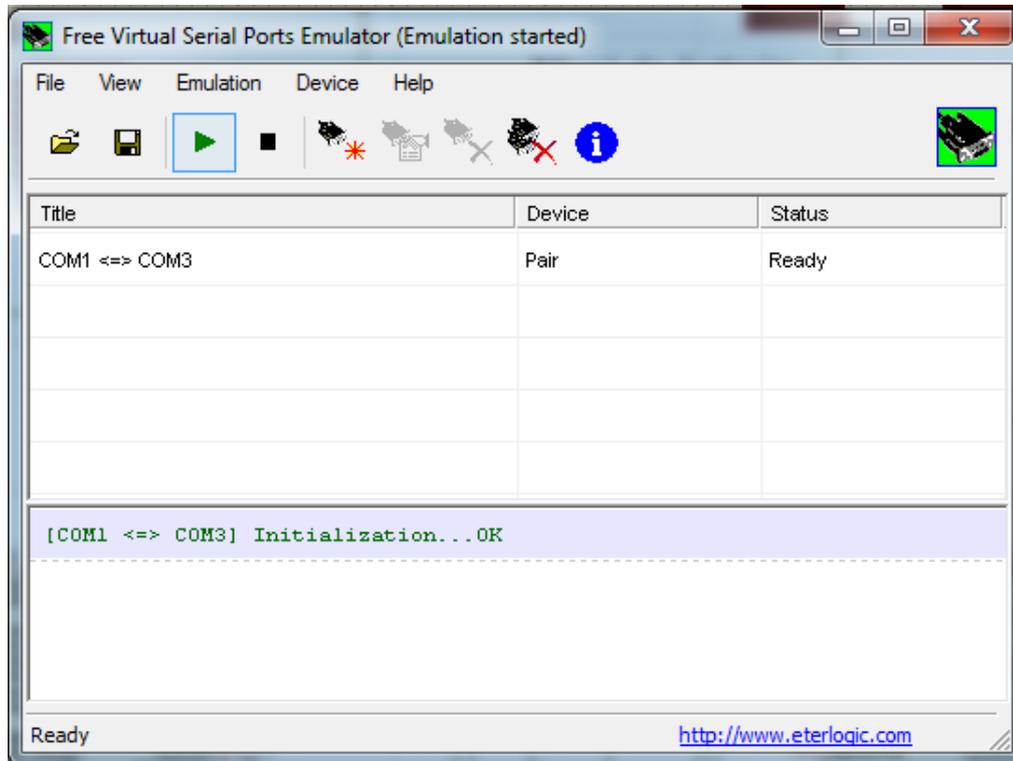


Figura 3-13 - Emulador de porta serial

#### 3.4 Placas de circuito impresso do projeto

O layout das duas placas de circuito impresso, mestre (FIG. 3-14) e escravo (FIG 3.15), foram desenvolvidas no ARES 7 Professional.

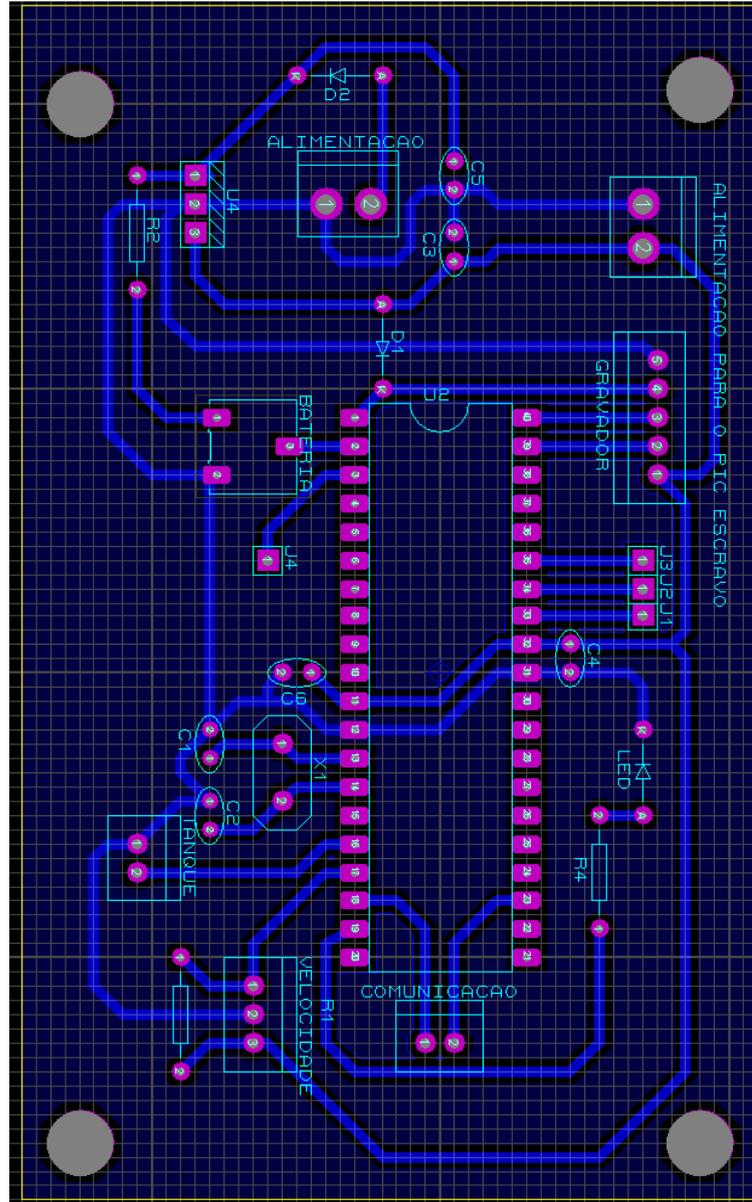


Figura 3-14 -Layout da placa de circuito Mestre

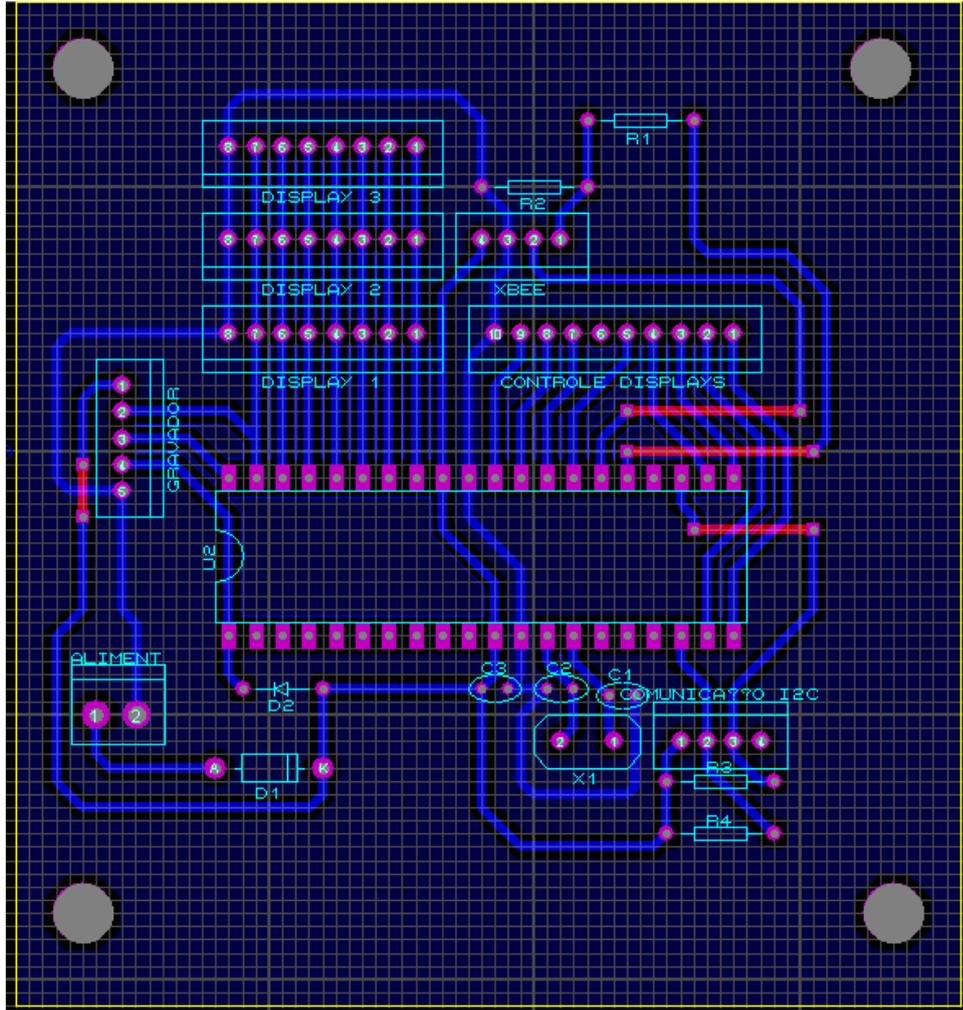


Figura 3-15 - Layout da placa de circuito Esclavo

## 4 RESULTADOS E DISCUSSÃO

O trabalho apresentado foi realizado de modo satisfatório, visto que foi desenvolvido um sistema integrado de telemetria e de supervisor para o Inconfidentes Baja.

De acordo com a Figura 4-1, pode-se observar uma simulação realizada no *software* Proteus. No momento em que se modificava os valores de entrada das portas referente aos sensores, praticamente no mesmo instante ocorria o envio dos novos valores desses instrumentos para a porta serial.

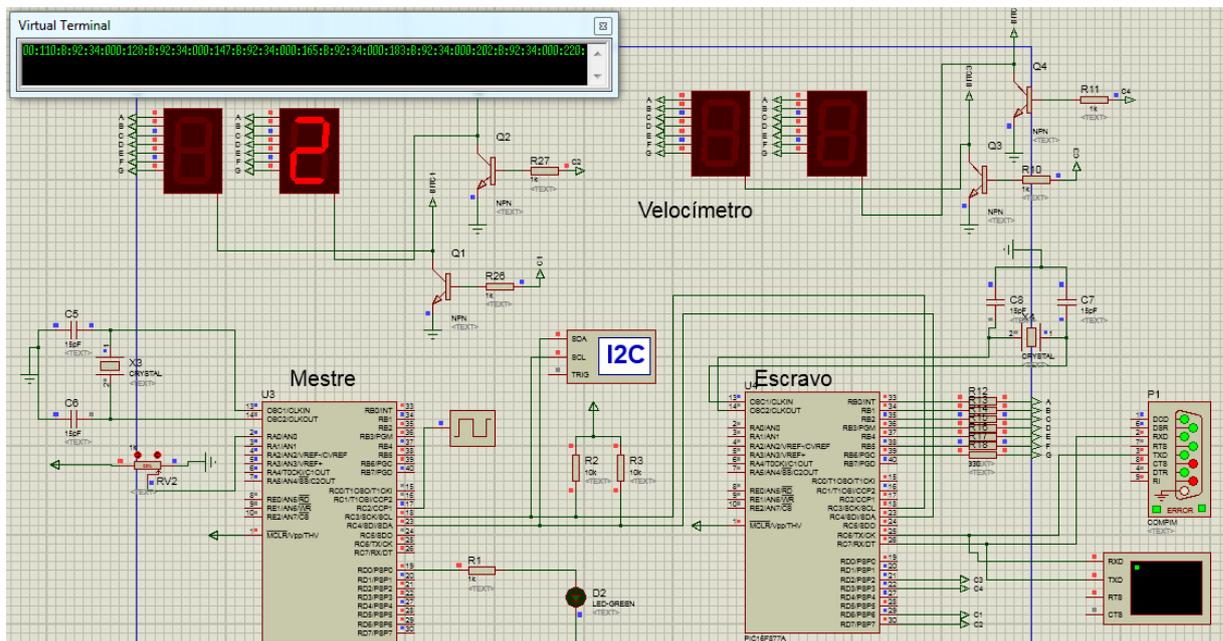


Figura 4-1 - Projeto geral em modo run no software ISIS

No teste do hardware - placas mestre e escravo - a comunicação via I<sup>2</sup>C ocorreu como desejado. Portanto, constatou-se que a programação foi implementada de forma correta, bem como o desenvolvimento das placas de circuito impresso.

Quanto ao funcionamento dos displays de sete segmentos, foi identificado um pequeno problema na visualização dos valores dos instrumentos. Pois o sistema apresentava todos os valores das medições nos displays durante 2 segundos e em seguida apagava por esse mesmo período, isso de forma cíclica.

A Figura 4-2 e a Figura 4-3 mostram as placas mestre e escravo desenvolvidas no projeto em funcionamento. Na primeira, os displays de sete segmentos do nível de bateria exibem o valor de 81% de carga. E na segunda figura, os displays de sete segmentos do velocímetro exibem 27

km/h. Pela percepção a olho nu vê-se os displays funcionando simultaneamente, mas por foto apenas um funcionando por vez.

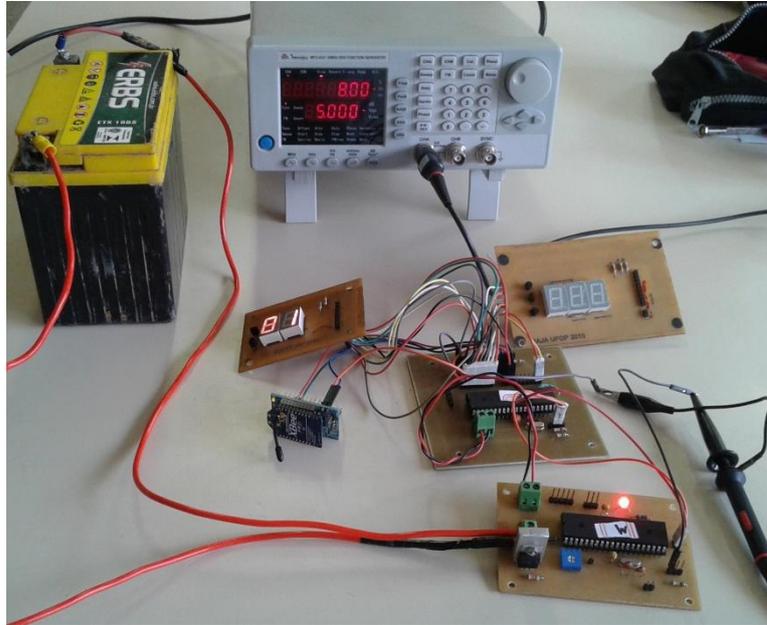


Figura 4-2 - Sistema em funcionamento mostrando 81% da carga da bateria através dos displays

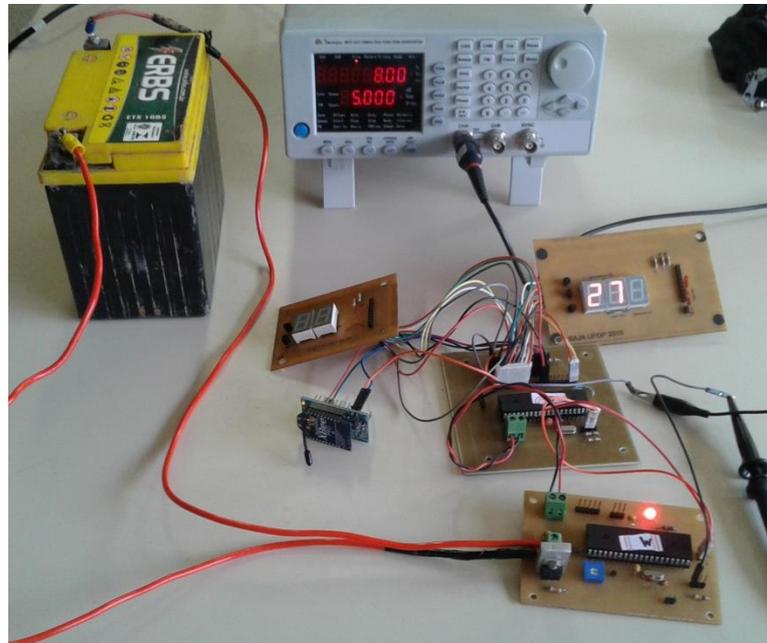


Figura 4-3 - Sistema em funcionamento mostrando a velocidade a 27 km/h através dos displays

O supervisório desenvolvido no Visual Studio funcionou como planejado, já que a variação dos instrumentos do veículo foi percebida de forma praticamente instantânea (Figura 4-4). Foi possível visualizar a velocidade, o nível de combustível e de bateria nos elementos gráficos, além do valor da distância percorrida. Os botões inseridos de iniciar e finalizar o gráfico funcionaram corretamente, assim como os de salvar e abrir dados.

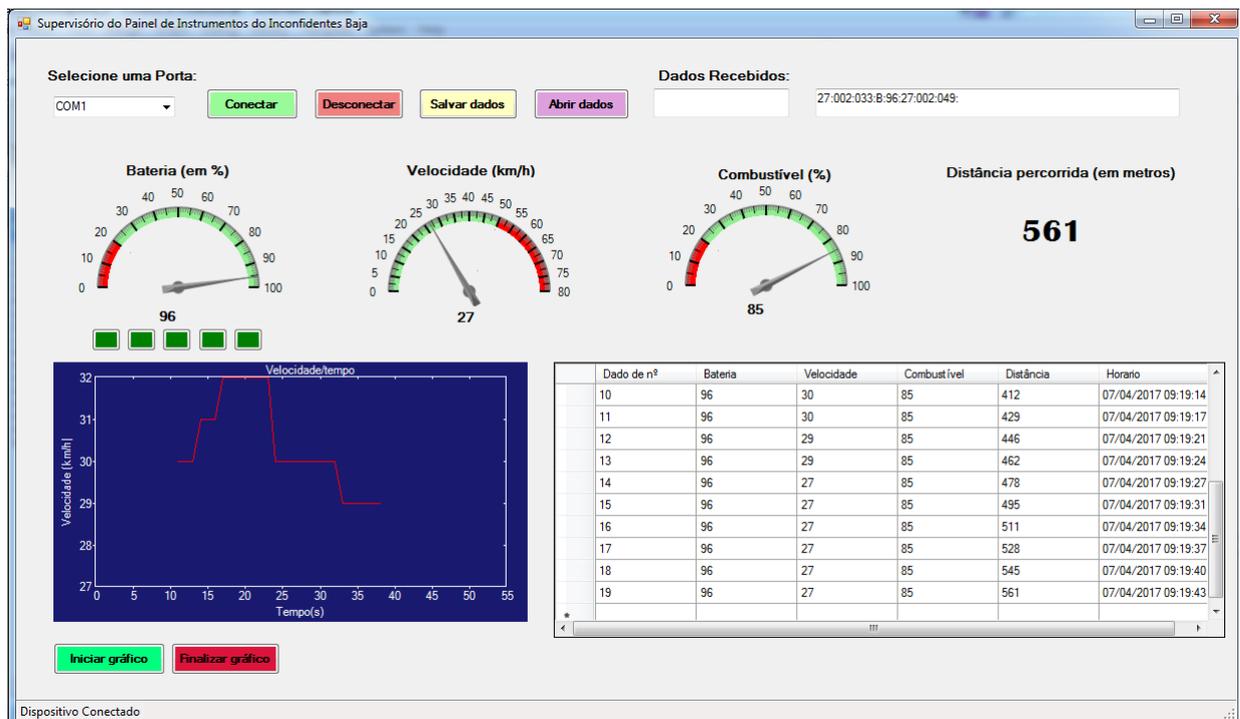


Figura 4-4 - Sistema supervisório em funcionamento

## 5 CONCLUSÃO

Este projeto foi desenvolvido visando criar um sistema de telemetria e supervisão eficiente para os integrantes do Inconfidentes Baja, de custo relativamente baixo e que gerasse à equipe do projeto uma boa avaliação na prova de apresentação do projeto elétrico na competição.

Com isso, inúmeras dificuldades foram atravessadas durante a realização do projeto, que foi desde a escolha dos componentes, a programação, a manipulação dos softwares utilizados, a confecção das placas de circuito impresso, até os testes reais feitos em laboratório. Espera-se que este projeto possa ser aprimorado constantemente pelas próximas equipes do projeto elétrico do Inconfidentes Baja.

O sistema desenvolvido neste trabalho poderá gerar maior segurança à equipe durante a competição, com a disponibilidade dos mensurandos indicados na tela de supervisão instalada no box. Assim, os problemas do veículo poderão ser evitados ou solucionados mais rapidamente, como uma pane seca, por exemplo.

Como continuação e aprimoramento deste trabalho, pretende-se instalar o sistema desenvolvido no novo veículo que está sendo projetado pela equipe Inconfidentes Baja para a próxima competição. Durante a Competição Baja Regional Sudeste que será realizado no final do ano de 2017 a subequipe da área da eletroeletrônica será avaliada na apresentação deste projeto.

Em relação à visualização dos dados nos displays de sete segmentos, pretende-se solucionar o problema identificado por meio do aperfeiçoamento dos algoritmos dos microcontroladores.

Pelo fato de não ter usado a memória não volátil do microcontrolador neste projeto, a informação da distância percorrida é perdida quando a placa de circuito impresso é desenergizada. Para solucionar este problema, pretende-se habilitar o uso da memória EEPROM.

Além disso, como alguns pinos do microcontrolador mestre não estão sendo utilizados, tem-se como proposta de aprimoramento a implementação de novos sensores, como um sensor de temperatura do motor e um acelerômetro para medir as vibrações do tanque.

## 6 REFERÊNCIAS BIBLIOGRÁFICAS

CURVÊLLO, R. **Um sistema de supervisão remota de eletrodomésticos em rede sem fio IEEE 802.15.4**, 2008. 123 f. Tese (Mestrado em Engenharia Elétrica) – Centro de Ciências Tecnológicas, Universidade Federal de Santa Catarina, Joinville, Santa Catarina, 2008.

DIGI INTERNATIONAL INC. XBee®/XBee-PRO® RF Modules. **Product Manual v1.xEx - 802.15.4 Protocol**. Estados Unidos da América. 2009

FIREGARD. **Protocolo I<sup>2</sup>C**. Disponível em: <<http://http://firegard.blogspot.com.br/p/protocolo-i2c.html>>. Data de acesso: 4 de abril de 2017.

HUSSIN, R. et al. Wireless traffic light controller for emergency vehicle through XBee and Basic Stamp Microcontroller. **Procedia Engineering**, v. 41, n. Iris, p. 636–642, 2012. Disponível em: <<http://dx.doi.org/10.1016/j.proeng.2012.07.223>>.

MESSIAS A. R. **Controle remoto e aquisição de dados via XBee/ZigBee (IEEE 802.15.4)**. Disponível em:< <http://www.rogercom.com/ZigBee/ZigBee.htm> >. Data de acesso: 27 de maio de 2015.

MICROCHIP TECHNOLOGY INC. **PIC16F87XA Data Sheet**. Estados Unidos da América.2003. 232p.

PEREIRA, F. **Microcontroladores PIC: Programação em C**. São Paulo: Érica, 2002. 358 p.

PEREIRA, F. **Microcontroladores PIC: Técnicas Avançadas**. 2. ed. São Paulo: Érica, 2002. 358 p.

SAE BRASIL. **Baja**. Disponível em: <<http://www.portal.saebrasil.org.br/programas-estudantis/baja-sae-brasil>>. Acesso em: 27 jan. 2017.

SALEIRO M.; EY E. **Zigbee, Uma abordagem prática**. s.d. 52f. Trabalho de Redes de Comunicação - Escola Superior de Tecnologia – EEE, Universidade do Algarve, Portugal.

SALLES, C. et al. A influência da Informação (Telemetria) na Gestão de Frota: O Estudo de Caso em uma Empresa de Médio Porte do Interior do Estado de São Paulo. **XXXVI Encontro Nacional de Engenharia de Produção, Contribuições da Engenharia de Produção para Melhores Práticas de Gestão e Modernização do Brasil**, João Pessoa, PB, p. 1–17, 2016.

SALVADOR, M; SILVA, A. P. G. **O que são sistemas supervisórios?** Disponível em: <[http://www.wectrus.com.br/artigos/sist\\_superv.pdf](http://www.wectrus.com.br/artigos/sist_superv.pdf)>. Acesso em: 13 de fevereiro de 2017.

SANT'ANA, A. R. et al. Sistema de Detecção de Colisão entre Veículos Utilizando GPS e Zigbee. **Anais do XX Congresso Brasileiro de Automática**, Belo Horizonte, MG, p.1466–1473, 2014.

SILABS. **SINGLE-CHIP USB TO UART BRIDGE**. Eletronic Publication, 2002.

SOHN, O. A. B. Z. **Desenvolvimento de aplicativo do driver sintonizador com diversidade para Linux embarcado em Set Top Box, 2011**. 69 f. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Departamento de Informática e Estatística, Universidade de Santa Catarina, 2011.

SOUZA, D. J; LAVINIA, N. C. **Conectando o PIC 16F877A: Recursos Avançados**. 1. ed. São Paulo: Érica, 2003. 280 p.

SOUZA, D. J. **Desbravando o PIC: Baseado no Microcontrolador 16F84**. 4. ed. São Paulo: Érica, 2000.

TEIXEIRA, F. C. R.; TOURNIER, R. Utilização de Telemetria para Diagnóstico Automotivo à Distância. **Blusher Engineering Proceedings**, v.2, n.1, p. 8, 2015.

ROMERO, C. A. V; JAIMES, J. E. B; GONZÁLEZ, D. C. P. (2015). Parámetros de configuración en módulos XBEE-PRO® S2B ZB para medición de variables ambientales. **Revista Tecnura**, v.19, n.45, p.141-157, 2015. Disponível: <http://dx.doi.org/10.14483/udistrital.jour.tecnura.2015.3.a11>.

WALDO, J. **Embedded computing and Formula One racing**. Pervasive Computing, IEEE v.4, n.3, p. 18 – 21, 2005.