https://eventos.utfpr.edu.br//sicite/sicite2020

Protocolo de comunicação sem fio para CanSats

Wireless communication protocol for CanSats

Arthur Hiroyuki Cavequia Takahashi
arthurhct@hotmail.com
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, PR, Brasil

Luís Fernando Caparroz Duarte
lfduarte@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, PR, Brasil

Rodrigo Augusto Borges Bustos
rabbustos@outlook.com
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, PR, Brasil

Kayque Saviti da Silva
Kako.saviti@hotmail.com
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, PR, Brasil

RESUMO

A informação das grandezas medidas por dispositivos de monitoramento remoto deve percorrer seu trajeto de forma confiável desde sua origem até seu destino. Este trabalho apresenta o desenvolvimento de um protocolo de transmissão de dados que possibilita comunicação sem fio entre uma sonda de monitoramento remoto do tipo CanSat e uma estação radio base, permitindo a realização da configuração do pacote de dados transmitido, a verificação da integridade da informação transmitida e o reenvio da informação em caso de detecção de erro. O protocolo desenvolvido foi implementado em hardware e testado. O resultado dos testes revelou que o protocolo funciona conforme especificado.

PALAVRAS-CHAVE: Monitoramento remoto. Protocolo de comunicação. CanSat.

ABSTRACT

The information regarding the measurements performed by remote monitoring devices must travel in a reliable way from its origin to its destination. This work presents the development of a communication protocol that allows wireless communication between a CanSat remote monitoring probe and a base station, able to perform the configuration of the transmitted data package, with data integrity verification and retransmission of the information in case of error detection. The developed protocol was implemented in hardware and tested. The test results revealed that the protocol works according to its specification.

KEYWORDS: Remote sensing. Communication protocol. CanSat.

Recebido: 03 set. 2020.
Aprovado: 01 out. 2020.

INTRODUÇÃO

CanSats são satélites em miniatura com formato típico de uma lata de refrigerantes, que dá origem ao seu nome. Limitados a massa de até 1kg, são classificados como picossatélites. Comumente estes pequenos satélites são utilizados como sondas em instrumentação meteorológica e podem ser lançados por foguetes ou içados por balões.

Seu desenvolvimento é adotado como porta de entrada nos estudos avançados da área de instrumentação eletrônica por fazer uso de sensores cujos dados são empregados em estudos climáticos e ambientais. Em CanSats é possível encontrar sensores de temperatura, umidade relativa, pressão, gases, altitude e posicionamento global, dentre outros (LASTOVICKA-MEDIN, 2016).

No desenvolvimento dos CanSats é comum fazer uso de sensores de prateleira, comercializados em formato modular, que além de fazer a transdução da grandeza medida, também fazem o condicionamento e a conversão analógico-digital do sinal, entregando em sua saída um dado digital. Normalmente esses sensores permitem a configuração de seu funcionamento, de forma que é desejável que a comunicação com o sensor seja bidirecional. As interfaces seriais mais comuns neste cenário são UART, SPI e I^2C.

Dentre as interfaces citadas, a I^2C se apresenta como amplamente utilizada em sensores, pois apresenta vantagens como o endereçamento de 7 bits, que permite a comunicação de vários sensores em um único barramento e robustez, garantindo confiabilidade nas operações de leitura e escrita. No entanto, o uso das interfaces UART e SPI também é bastante expressivo, ao passo que alguns periféricos dependem quase que exclusivamente das mesmas, como exemplo, os módulos de GPS e cartões SD, que fazem uso de UART e SPI, respectivamente (FRENZEL JR, 2015).

Protocolo de comunicação

Um protocolo de comunicação pode ser definido como um conjunto de regras com as quais dois ou mais dispositivos se comunicam, de modo que seja possível validar as informações trocadas. Na prática, o protocolo de comunicação tem a mesma função de uma linguagem, permite a troca de informação de modo compreensível entre os entes envolvidos. Protocolos de comunicação são essenciais em redes de comunicação, ao passo que estabelecem o modo como as comunicações são realizadas, assim também como a sintaxe e a semântica das mesmas (POPOVIC, 2018).

Dentre suas principais funções destaca-se a definição do formato das mensagens trocadas, o estabelecimento de como as mensagens são processadas e o preestabelecimento de ações a serem tomadas quando situações inesperadas ocorrem. Geralmente protocolos são modelados ou ainda definidos com base em máquinas de estados finitos, com as quais estados e transições de estados podem ser definidos formalmente (POPOVIC, 2018).

Uma mensagem em um protocolo está sujeita a uma padronização de formato, ou seja, definição de codificação, definição de campos de informação e seus respectivos tamanhos. É recorrente a utilização de três campos: o cabeçalho,

o *payload* e o *checksum*. O cabeçalho é comumente utilizado para sinalização e controle, o *payload* carrega a informação e finalmente o *checksum* é utilizado para verificar a validade da informação, podendo ser calculado por meio de diferentes métodos como o Adler-32 e o CRC32 (POPOVIC, 2018).

MATERIAIS E MÉTODOS

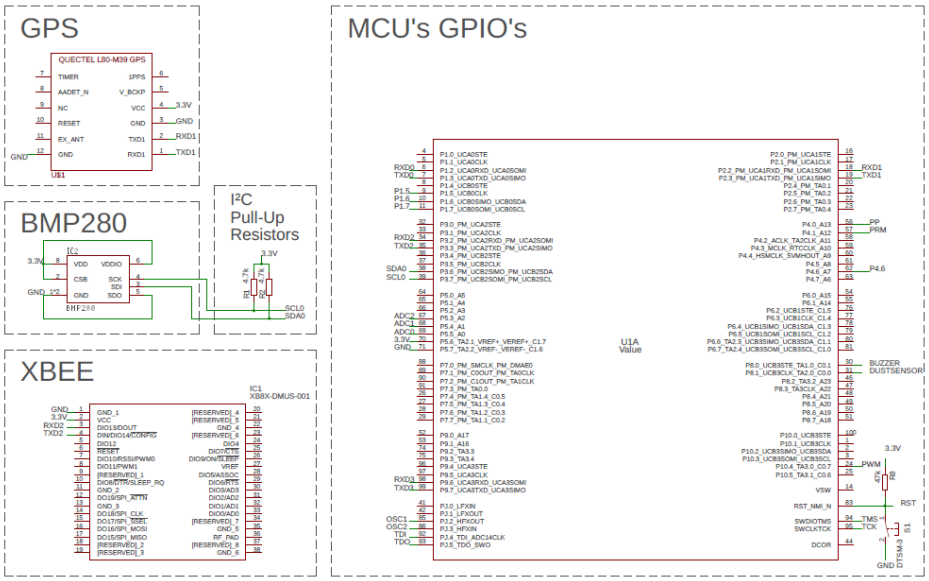
A abordagem utilizada para desenvolver o sistema basicamente consiste nas quatro fases demonstradas no diagrama da figura 1, fases essas que normalmente são utilizadas no desenvolvimento de protocolos de comunicação (POPOVIC, 2018).

Figura 1 – Diagrama de desenvolvimento do sistema.

Fonte: Autoria própria (2020).

Esta abordagem foi aplicada com base no circuito da figura 2, composto pelos principais componentes de um CanSat, sendo eles o microcontrolador (MSP432P401R), o GPS (Quectel L80), o sensor de pressão e temperatura (BMP280) e o transceptor (Xbee SX).

Figura 2 – Circuito esquemático do sistema de teste desenvolvido.



Fonte: Autoria própria (2020).

Estes componentes foram escolhidos por serem compatíveis com os componentes permitidos para serem utilizados na CanSat Competition, uma competição acadêmica internacional de desenvolvimentos de CanSats, realizada anualmente pela American Astronautical Society, na qual os integrantes deste trabalho estão inscritos para participar com o resultado do trabalho.

Desta forma, o protocolo programado no microcontrolador foi especificado para permitir a configuração e a recepção dos dados do GPS e do sensor de pressão e temperatura por meio de comunicação sem fio através do transceptor.

O ambiente de desenvolvimento integrado utilizado na programação do protocolo foi o *Code Composer Studio*, da *Texas Instruments*, fabricante do microcontrolador adotado, que disponibiliza gratuitamente a biblioteca *Driverlib*, usada para realizar a configuração do microcontrolador.

O protocolo de comunicação foi baseado na mensagem descrita na figura 3.

Figura 3 – Exemplo de mensagem padrão do protocolo elaborado.

DUDE & 7 & 008 & A F 6 7 C 0 8 7 & 0AAC0253 \r

Fonte: Autoria própria (2020).

A codificação escolhida para os dados foi a hexadecimal codificada em ASCII, com uso de caracteres especiais para controle e manuseio da informação. O caractere & foi adotado como caractere separador e o caractere *carriage return* (\r em linguagem C ou 0x0D em hexadecimal) foi adotado como terminador de linha, o que facilitou o processamento das mensagens devido a compatibilidade com editores de texto.

O *checksum* adotado foi o Adler-32, por exigir baixa capacidade computacional. Ainda que seja menos confiável que o CRC32, pode ser calculado por meio do cálculo de dois *checksum's* de 16 bits e posteriormente os concatenando em uma variável inteira de 32 bits (MAXINO, 2009).

Cada mensagem possui um total de 5 campos. O quadro 1 apresenta a descrição de cada um dos campos.

Quadro 1 – Descrição dos campos presentes nas mensagens do protocolo.

Campo	Tamanho	Exemplo	Descrição
Cabeçalho	4 bytes	DUDE	Indica o início da transmissão
Comando	1 byte	7	Indica o tipo de conteúdo do pacote
Nº de bytes	3 bytes	008	Número de bytes do pacote transmitido
Pacote	Variável	AF67C087	Informação transmitida
Checksum	8 bytes	0AAC0253	Permite a verificação da integridade da informação recebida

Fonte: Autoria própria (2020).

A estrutura de mensagem apresentada permite o envio de pacotes de tamanhos variáveis, fornecendo informações de controle como o *checksum* e a quantidade de bytes do pacote.

Todas as transações do protocolo são baseadas em comandos, podendo ser divididos em comandos de controle e comandos de dados. Os comandos de controle regulam o início, meio e fim das transações de modo que o fluxo de informações de ambas as fontes ocorra adequadamente, ou seja, permitindo que o sistema opere segundo o modo *half duplex*. Os comandos de dados permitem o envio de informações de fato, como dados de sensores.

A dinâmica do protocolo foi baseada em uma dinâmica similar ao do protocolo I^2C, operando em modo mestre-escravo. O dispositivo mestre (microcontrolador) controla a ordem em que as informações são enviadas pelos dispositivos escravos (sensores), enquanto os dispositivos escravos enviam as informações periodicamente, de acordo com o estabelecido pelo dispositivo mestre.

O protocolo permite o envio de 7 comandos diferentes que permitem tanto o controle de operações como a transmissão dos dados. O quadro 2 relaciona os comandos e suas respectivas funções.

Quadro 2 – Comandos suportados pelo protocolo com exemplos, descrição e tamanho.

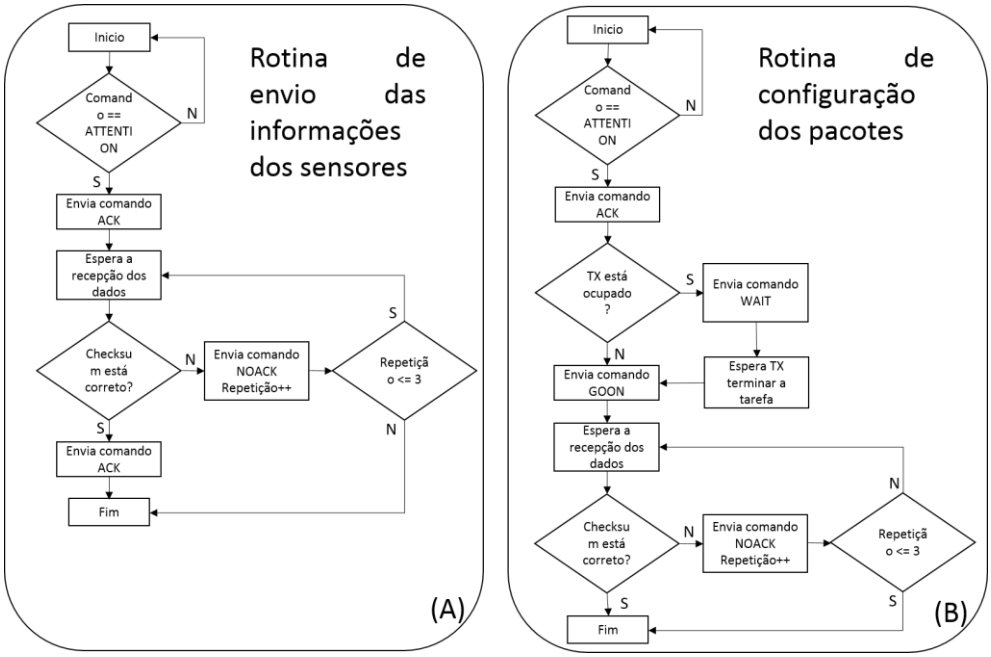
Comando	Nome	Descrição
1	ACK	Confirma a recepção da informação
2	NOACK	Indica que a informação não foi recebida ou ainda indica alguma incoerência
3	ATTENTION	Sinaliza o início de uma transmissão
4	WAIT	Indica que o receptor está ocupado
5	GOON	Indica que o receptor não está ocupado
6	PACKETCONFIG	Configuração dos dados a serem enviados
7	DATA1	Dados dos sensores

Fonte: Autoria própria (2020).

Os comandos de 1 a 5 são comandos de controle, enquanto os comandos 6 e 7 são comandos de transmissão de dados, que em conjunto, formam os processos de troca de informação.

Foram desenvolvidos dois processos de troca de informação, um para configuração de parâmetros da transmissão e outro para a transmissão dos dados. Os fluxos de dados permitidos pelo protocolo estão representados na figura 4.

Figura 4 – Comandos suportados pelo protocolo com exemplos, descrição e tamanho.

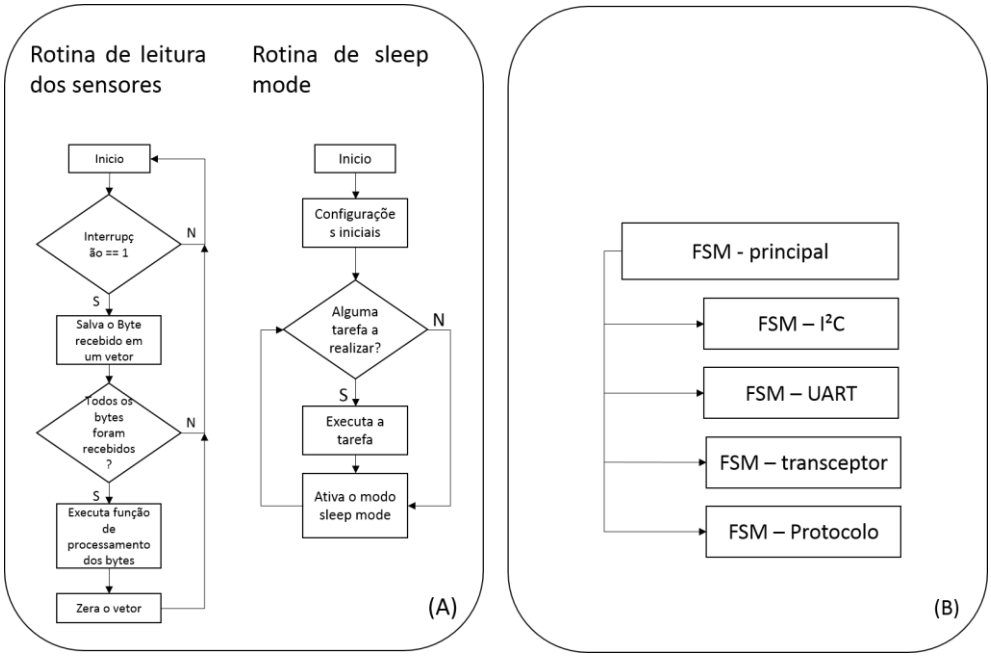


Fonte: Autoria própria (2020).

Além das rotinas do protocolo, o sistema também executa rotinas de leitura de sensores com geração de interrupção e a rotina de economia de energia do microcontrolador (*sleep mode*). Ambas as rotinas foram baseadas em máquinas de

estado (FSM), reguladas por uma rotina principal que coordena a execução do sistema como demonstra a figura 5.

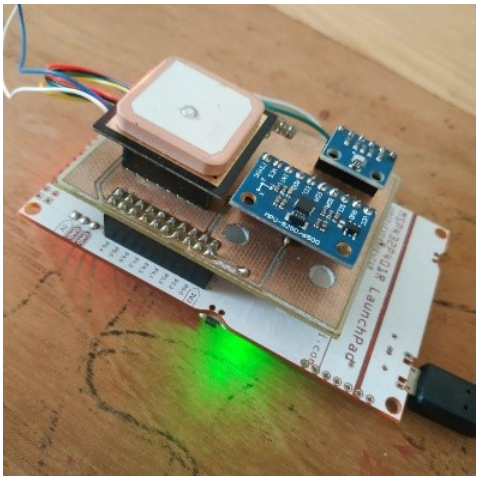
Figura 5 – Fluxograma generalizado das rotinas do microcontrolador.



Fonte: Autoria própria (2020).

O protocolo de comunicação desenvolvido foi implementado em hardware e testado tanto por meio de um terminal serial, no qual foram simuladas as rotinas de envio de informação, quanto por meio do transceptor, operando de forma integral. A placa de desenvolvimento utilizada nos testes é exibida na figura 6.

Figura 6 – Placa de desenvolvimento utilizada para teste.



Fonte: Autoria própria (2020).

RESULTADOS

As operações de configuração de pacotes e de envio de informação dos sensores funcionaram exatamente como descrito na figura 4.

O *checksum* se mostrou funcional, detectando corretamente os erros nas mensagens e permitindo o reenvio de informação.

Para testar o reenvio de informações, os erros nas mensagens foram propositalmente inseridos com a utilização de um terminal serial no lugar do transceptor Xbee SX. Como o transceptor adotado confere ao sistema uma probabilidade de erro de bit (BER) muito baixa, não foram observados erros de comunicação durante os testes com o sistema integral.

CONCLUSÃO

Com base no estudo das tecnologias envolvidas foi desenvolvido um protocolo de comunicação que permite realizar a transmissão dos dados dos sensores presentes em um CanSat, além de possibilitar a configuração destes sensores.

O protocolo foi implementado no sistema de teste e os resultados dos testes comprovaram que o protocolo proposto opera conforme sua especificação. A integridade dos dados transmitidos foi garantida em função da detecção de erro e reenvio da informação.

AGRADECIMENTOS

O presente trabalho foi realizado com o apoio da Fundação Araucária FA – Paraná/Brasil, a partir da concessão de uma bolsa do programa de iniciação científica, para tanto presta-se aqui o devido agradecimento a este apoio, fundamental para o desenvolvimento deste trabalho.

REFERÊNCIAS

FRENZEL JR, L. E. **Handbook of Serial Communications Interfaces:** A Comprehensive Compendium of Serial Digital Input/Output (I/O) Standards. Austin, TX, USA: Newnes, 2015.

LASTOVICKA-MEDIN, G. Nano/pico/femto-satellites: review of challenges in space education and science integration towards disruptive technology. **2016 5Th Mediterranean Conference On Embedded Computing (Meco)**, [S.L.], p. 357-362, jun. 2016. Disponível em: https://ieeexplore.ieee.org/document/7525781. Acesso em: 26 de jul. 2020.

MAXINO, T.C.; KOOPMAN, P.J. The Effectiveness of Checksums for Embedded Control Networks. **IEEE Transactions on Dependable and Secure Computing**, v. 6, n. 1, p. 59-72, 29 nov. 2009. Disponível em: https://ieeexplore.ieee.org/document/4358707. Acesso em: 28 ago. 2020.

POPOVIC, M. **Comunication Protocol Engineering.** Segunda edição. Boca Ranton, FL, EUA: CRC press, 2018.