DESENVOLVIMENTO DE SISTEMAS DE SIMULAÇÃO E MONITORAMENTO DE VOO DE VANTS

Gabriel Teixeira Nakata[1], Ricardo Ferreira Martins[2], André Bittencourt Leal[3]

[1] Acadêmico do Curso de Engenharia Elétrica – CCT - bolsista PROBIC/UDESC
[2] Doutorando, Programa de Pós-Graduação em Engenharia Elétrica – CCT - ricardo.martins@udesc.br
[3] Orientador, Departamento de Engenharia Elétrica – CCT - andre.leal@udesc.br

Palavras-chave: *VANTS, Mission Planner, Arduino, Protocolo I2C, Protocolo MavLink, Sensoriamento.*

Atualmente, o mercado de *Veículos Aéreos Não Tripulados* (*VANTs*) vem ganhando reconhecimento em diversas áreas, especialmente na agricultura, na qual estes auxiliam no mapeamento de regiões, na detecção de problemas de plantio e na aplicação de defensivos agrícolas. Com a evolução dos tipos de serviços para *VANTs*, mais sensores e atuadores estão sendo embarcados nestas aeronaves para que elas possam cumprir missões de uma forma automática. Este trabalho teve como objetivo o desenvolvimento de novos recursos de sensoriamento (múltiplos sensores), no sentido de permitir que tais recursos possam ser embarcados em *VANTs* de diferentes marcas e modelos.

Nesta pesquisa, foi utilizada a controladora de voo ArduPilotMega (*APM*) - versão 2.6, desenvolvida pela *3D Robotics*. A *APM* possui o controlador *ATMEGA2560*, bem como os sensores: giroscópio, acelerômetro, magnetômetro, e entrada/barramento para *GPS* e compasso externo. O *firmware* usado pela *APM* 2.6 é o *ArduPilot*, com uma versão voltada para veículos multirotores (*ArduCopter*). Para programar o planejamento de missões, utilizou-se o software aberto *Mission Planner*, na sua versão 1.3.10.

Na fase inicial, realizou-se a análise dos códigos do *firmware* do *VANT*, o *ArduCopter* e do *software* de planejamento de missões, o Mission Planner, tendo com objetivo o entendimento da forma de transferência dos dados entre eles e como adicionar mais informações, bem como o estudo da versão do *ArduCopter* para a pesquisa. Com as informações adquiridas durante a análise, foi desenvolvido um código para uma placa adicional (*Arduino*) que realiza a leitura de sensores e envia os dados para o *APM*, via comunicação através do barramento *I2C*, e também um código para o *APM* que faz a leitura das informações que estão chegando neste barramento, armazenando-as em vetores.

Para validar o funcionamento do sistema, foram utilizados 2 sensores de distância ultrassônico (HCSR-04). Tais sensores são aplicados, por exemplo, para a identificação de obstáculos durante o voo, com o objetivo de evitar colisões da aeronave. Foi realizado um teste para verificar a comunicação *I2C* e outro para verificar a precisão dos sensores HC-SR04 junto com o envio de dados pelo *I2C*. Com a validação das leituras dos dados dos sensores no *APM*, o código de leitura do barramento *I2C* foi incluído no *firmware*, obtendo assim a leitura dos dados enviados pelo *Arduino*. O *firmware* do *ArduCopter* escolhido inicialmente foi a versão 3.2.1, devido a sua biblioteca ser atualizada. No entanto, devido ao fato desta versão não desabilitar o modo de simulação do controle do *software* e do *hardware*, também chamado de *HIL* (*Hardware-in-the-Loop*), foi feita a substituição para a versão 2.9.1.

A leitura dos valores dos sensores, pelo *Mission Planner*, foi realizada através da implementação no *firmware*, considerando uma variável já existente no sistema (*receiver_rssi*).

A Figura 1 apresenta o ambiente de testes utilizado nos ensaios, onde do lado esquerdo estão o *APM* e o Arduino acoplados através do barramento *I2C*, e as conexões com os sensores. Do lado direito, encontra-se o anteparo usado para a identificação das distâncias.

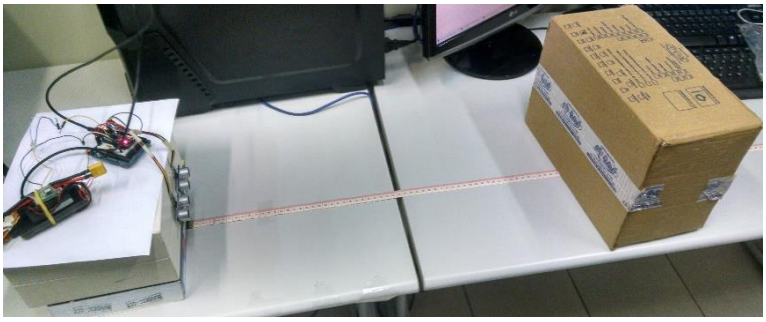

Fig. 1 *Estrutura de teste da distância real*

A Figura 2 mostra os resultados das informações obtidas pelos sensores. No primeiro momento, estas informações foram lidas na placa auxiliar (*Arduino Nano*), e posteriormente, transmitidas para o *APM* através do barramento *I2C*.

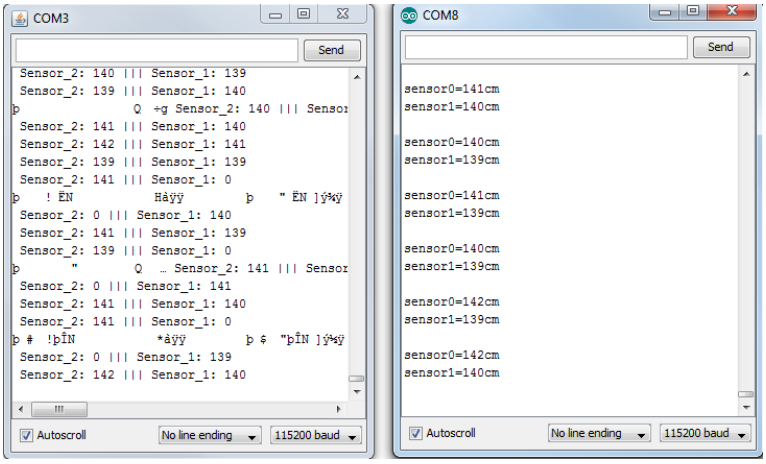

Fig. 2 *Dados recebidos pelo APM 2.6 (lado esquerdo); e dados dos sensores lidos pelo Arduino Nano 3V (lado direito)*

Com a realização deste trabalho, pode-se concluir que a comunicação funciona perfeitamente, o que permite a utilização desta estrutura para a inclusão de sensores embarcados em VANTs de um modo geral. Entretanto, novos testes e ajustes ainda precisam ser feitos a fim de melhorar a medição das distâncias. Vale destacar que este trabalho foi realizado ao longo de cinco meses, o que dificultou a análise e melhoria dos problemas encontrados.

Como trabalho futuro, é preciso fazer a integração das informações obtidas pelos sensores com o *APM*, no sentido de dar significado aos valores alcançados, e posteriormente, identificar comportamentos que o VANT deve adotar diante de condições de possíveis colisões.