UTILIZAÇÃO DE SOFTWARE LIVRE (OPEN SOURCE) PARA GERENCIAMENTO DE ROBÔS AUTÔNOMOS

Carlos Henrique Gomes Taborda Lopes (PROVIC), Ariangelo Hauer Dias (Orientador), e-mail: 15062926@uepg.br.
Universidade Estadual de Ponta Grossa/Departamento de Informática.

Engenharia Elétrica, Eletrônica Industrial, Sistemas e Controles Eletrônicos
Palavras-chave: *Open Source*, VANT, Automação.

Resumo
Com o constante avanço da robótica constantemente vemos o uso de robôs nas mais diversas tarefas possíveis, mas para isso antes é necessário programá-lo. Devido a possibilidade que a internet nos proporcionou de estarmos conectados surgiu a iniciativa de códigos *Open Source* que possibilita que códigos sejam compartilhados entre a comunidade de desenvolvedores fazendo assim com que códigos sejam reaproveitados e com que custos sejam reduzidos. Utilizando esses conceitos buscou-se o desenvolvimento de um VANT (Veículo Aéreo Não Tripulado) para voos em áreas de risco e plantações.

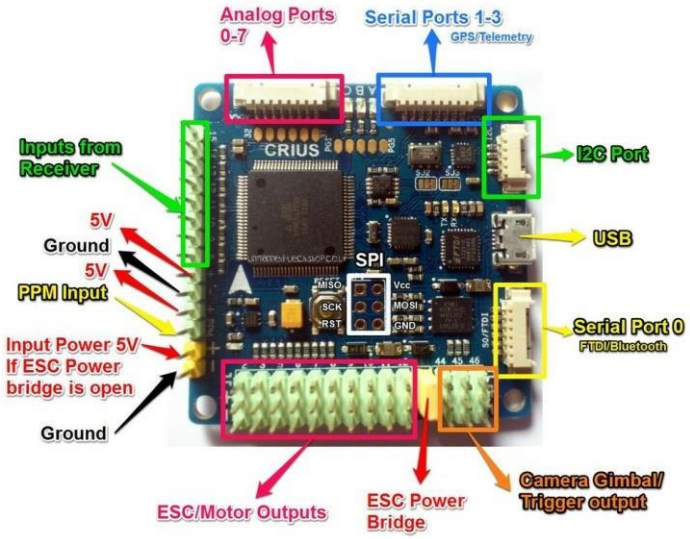
Introdução

O uso da robótica já é algo que nos deparamos constantemente, principalmente na indústria, isso deve-se ao avanço tecnológico que possibilitou reduzir os preços para a construção de robôs, aumentou seu desempenho e fez possível a miniaturização dos componentes.
Em nosso dia a dia robôs nos auxiliam em diversos deveres, entretanto muitos deles ainda necessitam de um operador humano para operá-lo, no entanto buscamos torná-los autônomos fazendo com que sejam capazes de executar suas tarefas de maneira eficiente e sem a necessidade de auxílio humano.
Para que isso ocorra é necessário programar o robô para execução de seus deveres passando a ele uma sequência de instruções por meio de *software*, mas por tratar-se de uma utilização específica muitos do softwares necessitam que para cada alteração de hardware seja necessário reprogramar o código fazendo com que os *softwares* já desenvolvidos sejam vendidos ou então descartados a cada atualização, tornando seu uso exclusivo.
Portanto, muitos dos *softwares* atuais são de exclusividade da empresa que o desenvolveu ou o comprou, porém muitos desenvolvedores seguem o conceito de *open source (código aberto),* que consiste no desenvolvimento e disponibilidade aberta do código, possibilitando assim que a comunidade de desenvolvedores aperfeiçoe ou faça mudanças no código conforme seja a sua necessidade.
Segundo esse conceito buscou-se para desenvolvimento do projeto a automação de um controlador de voo, que será utilizado para criação de um VANT com objetivo de executar voos autônomos em áreas de risco ou plantações.
Controladores de voos funcionam como o “cérebro” da aeronave, ou seja, são placas de circuito com sensores integrados que controlam a orientação, aceleração, além de permitir a conexão de outros periféricos.

Materiais e Métodos

Para a realização do projeto de pesquisa foi selecionado o micro controlador de vôo *Crius All In On Pro Flight Controller Multi Wii* (Figura 1), que possui suporte para os *firmwares* MegaPirateNG, MultiWii, ArduPlaneNG, ArduCopterNG e AeroQuadand, entretanto para o desenvolvimento do projeto foram utilizados os firmwares MegaPirateNG e MultiWii.

O micro controlador possui saídas para suportar até 8 motores, por possuir 8 canais de recepção, 4 portas seriais que podem ser utilizadas para debug, módulo Bluetooth, OSD, sistema de posicionamento global (GPS) e Telemetria, 2 saídas para servos PITCH e ROLL, uma saída para servo de ativação da câmera, 6 saídas analógicas para ligação de dispositivos, uma porta I2C para uso de outros sensores ou do próprio dispositivo, regulador de tensão LDO (*Low Dropout*) de 3.3V e 5V, um micro controlador ATMega2560, 6 eixos MPU 6050 que podem ser usados como giroscópio ou acelerador além da possibilidade de processar movimento, magnetômetro digital de 3 eixos modelo HMC5883L, conversor de nível lógico a



bordo padrão RoHS.

Figura 1 – Crius AIOP V2.0

Para a utilização da CRIUS (Figura 1) é necessário que seja feito o envio do código para o micro controlador utilizando um cabo micro USB e o software Arduino, durante o envio do código ao micro controlador os leds TX RX localizados ao lado da porta USB irão piscar.

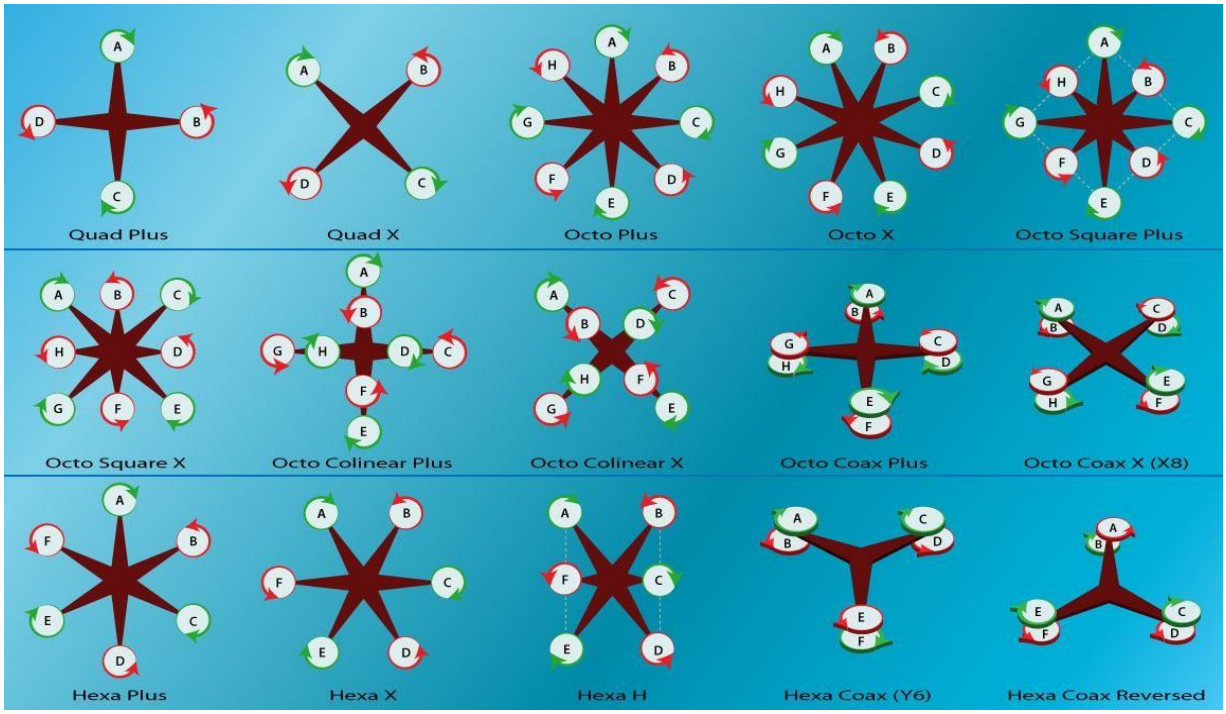


Figura 2 – Layout de Motores para VANTs.

Antes de se fazer o envio é necessário primeiramente configurar o módulo de GPS e layout dos motores (Figura 2), a definição do layout dependerá do *firmware*.
Para o *firmware* MegaPirateNG deve-se configurar o arquivo APM_Config.h, é necessário definir o GPS como GPS_PROTOCOL_UBLOX e o layout de motores como QUAD_FRAME, para o firmware MultiWii é o layout de motores será definido como QUADX e o GPS como UBLOX.
Como estações de controle em solo para voo existem inúmeros softwares possíveis de se utilizar tais como o Mission Planner, APM Planner, MAVProxy, e QGroundControl, durante o desenvolvimento do projeto foi utilizado o Mission Planner.
Uma estação de controle em solo é um software de controle executado em um computador presente no solo geralmente próximo a área de voo que o VANT realizará. É utilizado para criação e controle de todo planejamento de voo, alguns oferecem um mapa onde é possível definir pontos de passagem obrigatórios e acompanhar todo progresso de voo com exibição dos dados coletados pelos instrumentos presentes no VANT.

Resultados e Discussão
Com o aumento do número de motores o VANT possuiria maior potência possibilitando maior tempo de voo, maior elevação e aumento na quantidade de peso que se pode carregar, entretanto necessitaria de mais baterias e uma implementação de voo mais complexa.
Os VANTs no layout QUAD X possuem possuem maior velocidade, estabilidade e economia de bateria em relação ao QUAD +, e a alocação da câmera central permite um campo de visão melhor no layout QUAD X.
O firmware MegaPirateNG possui uma implementação mais intuitiva sem necessitar de muitas configurações em relação ao firmware MegaPirateNG.
O firmware MultiWii possibilita a utilização de qualquer versão do Arduino IDE

enquanto o MegaPirateNG possui códigos específicos para cada versão do Arduino IDE e necessita da instalação de bibliotecas adicionais.

Conclusões

Foi possível concluir que o *firmware* utilizado implica drasticamente no modelo de desenvolvimento do VANT, pois é o principal responsável pela execução do voo, controlando fatores como a comunicação por telemetria, a velocidade dos motores que implicará assim na direção do voo, etc.
Para seleção do *layout* de motores a ser adotado é necessário primeiramente verificar o suporte fornecidos pelos firmwares do micro controlador.
O micro controlador Crius não será mais produzido o que implica que para uma implementação futura do projeto será necessário a revisão dos firmwares e componentes utilizados.

Referências

MultiWii**. Firmware - GitHub mirror of the official SVN multiwii project**. Disponível em: <https://github.com/MegaPirateNG/ardupilot-mpng>

MegaPirateNG**.** **MegaPirateNG**. Disponível em: <https://github.com/MegaPirateNG/ardupilot-mpng>

Gaza07. **The Crius All In One Pro Flight Controller (AIOP) Multi Wii Manual rev 1.00.** Disponível em: <http://rctimer.com/download/Crius_AIOP_Manual_MWC.pdf>

Jumpy07, Gaza07. **Crius AIOP V2.0 Guide for MegaPirateNG.** Disponível em: <http://www.xxl-modellbau.de/abbildungen/quadcopter/flugsteuerung/crius_aiop_v2/Crius_AIOP_V2_0_Guide_for_MegaPirateNG.pdf>