Sistema de Controle de Aceleração

De LRM Wiki

Os sistema que atualmente empregamos para o controle de aceleração do robô Carina I (http://lrm.icmc.usp.br/carina/) consiste em um conjunto dado em parte por hardware e em parte por software. O objetivo é criar não só um elemento que permita a comunicação entre o PC e o carro, mas também envie os sinais necessários ao controlador do veículo para que atue no motor da formar como desejarmos.

Acelerador do Carina

Cada um dos elementos que compõe esse sistema são detalhados nas seções que se seguem, contendo desde características e informações descritivas até os aspectos técnicos de funcionamento.

Tabela de conteúdo

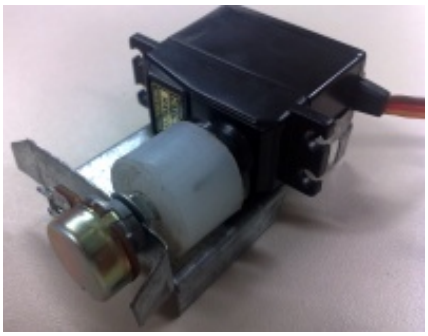
Hardware

Foi desenvolvido um dispositivo controlador, a que chamamos simplesmente de "Acelerador do carro", composto por um servo mecanismo acoplado a um potenciômetro analógico linear que atua diretamente nos sinais enviados ao controlador do carro. Um kit Arduino Duemilanove (http://arduino.cc/en/Main/ArduinoBoardDuemilanove) , executando um software embarcado (firmware v2.1), tem por objetivo prover os sinais de controle necessários ao funcionamento do servo motor, para os leds do display, além da troca de dados com o PC através de uma porta USB/serial.

Utilizamos um servo motor Hextronik HX5010, comum entre aeromodelistas, que proporciona até 6.50 kg-cm de torque e velocidade de 0.16 segundos para uma rotação de 60 graus. A tensão de 5V para a alimentação do motor foi obtida diretamente do Arduino (pinos do slot POWER).

A posição do motor pode ser variada através de uma modulação da largura do pulso enviado ao motor, ou seja, através de um sinal de controle conhecido como PWM (http://pt.wikipedia.org/wiki/Modula%C3%A7%C3%A3o_por_largura_de_pulso) . Uma das vantagens de trabalharmos com o Arduino é que não temos a necessidade de implementar o PWM pois algumas de seus pinos já fazem essa modulação através do mapeamento de sinais digitais em analógicos.





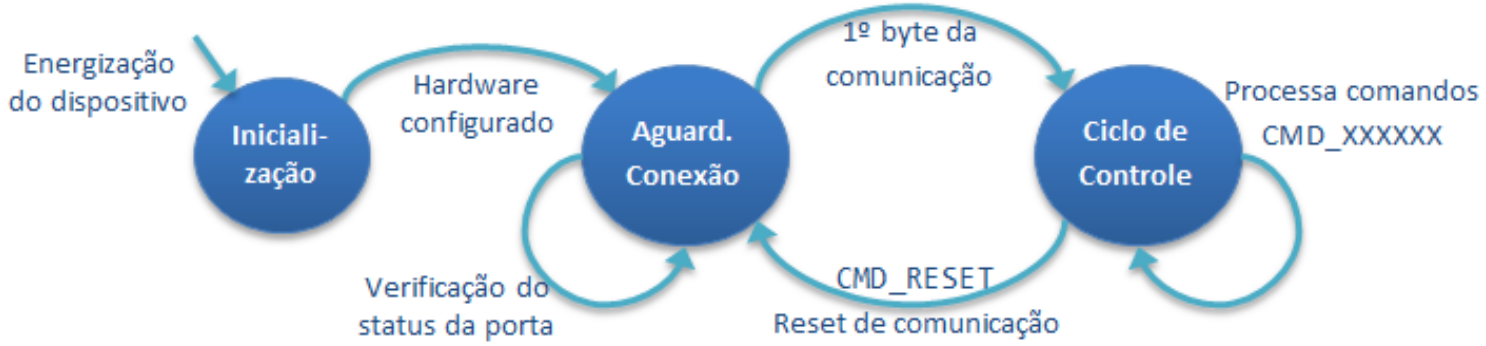

Software

Firmware

Um protocolo de comunicação foi estabelecido, de modo que todas as operações realizadas pelo controlador operem através de transações que se baseiam em um conjunto de operações definidas pelos identificadores a segui:

```
#define CMD_SETVEL      0xF5
#define CMD_STATUS      0xF6
#define CMD_SERVORESET  0xF7
#define CMD_RESET       0xF8
#define CMD_PING        0xF9
#define CMD_PONG        0xFA
```

Foram definidos 3 estados básicos de execução: *inicialização*, *aguardando conexão* e *ciclo de controle*.



Quando energizado, o Arduino carrega o código armazenado em sua memória e inicia a execução da função `setup()`. Nesta é feito o mapeamento dos pinos do micro-controlador que serão utilizados como entrada ou saída, e coloca-se o sistema em um estado *aguardando conexão*.

```
void setup() {
  pinMode(SERVO_PIN,OUTPUT);
  pinMode(YELLOWLED_PIN, OUTPUT);
  for (int thisPin = 0; thisPin < 7; thisPin++)  {
    pinMode( statusLedsPins[thisPin], OUTPUT );
    digitalWrite(statusLedsPins[thisPin], LOW);
  }
  establishContact();
}
```

Concluída a configuração básica do dispositivo, ocorre uma mudança automática do estado de *inicialização* para *aguardando conexão* (codificado na rotina `establishContact()`). Ao alcançar este estado é definida a velocidade de transmissão e se tem início o monitoramento da porta, colocando o dispositivo na condição de espera por algum dado de comunicação.

```
void establishContact() {
  Serial.begin(9600);
  resetServo();
  digitalWrite(YELLOWLED_PIN, LOW);
  while (Serial.available() <= 0) {
    waitingConnection();
  }
}
```

Este estado pode ser facilmente identificado pelo efeito *Kinght rider* dos leds de status. Ao receber o primeiro byte de comunicação, passa-se para o estado de trabalho ou também chamado de *ciclo de controle* e identificado visualmente pela intermitência do led amarelo.

PC

Para interfacear com o dispositivo, empacotando as informações segundo o protocolo de comunicação estabelecido e realizar a decomposição correta das respostas obtidas, foi desenvolvido duas classes em C++ para servir de driver (disponível em Arquivo:Acceleration.zip).

Updates

versão 2.1 foi acrescentada um par de palavras de controle para monitoramento da conexão (CMD_PING e CMD_PONG)