

<b>CURSO:</b>	<b>ENGENHARIA ELETRÔNICA</b>	
<b>DISCIPLINA:</b>	<b>Eletrônica Embarcada</b>	<b>TURMA: A</b>
<b>SEMESTRE:</b>	<b>2º/2019</b>	
<b>PROFESSORA:</b>	<b>Gilmar Silva Beserra</b>	
<b>ALUNO:</b>	<b>Lucas Gonçalves Campos e Wemerson Fontenele Sousa</b>	
<b>MATRICULA:</b>	<b>170016757 e 170024130</b>	

## Ponto de Controle 4

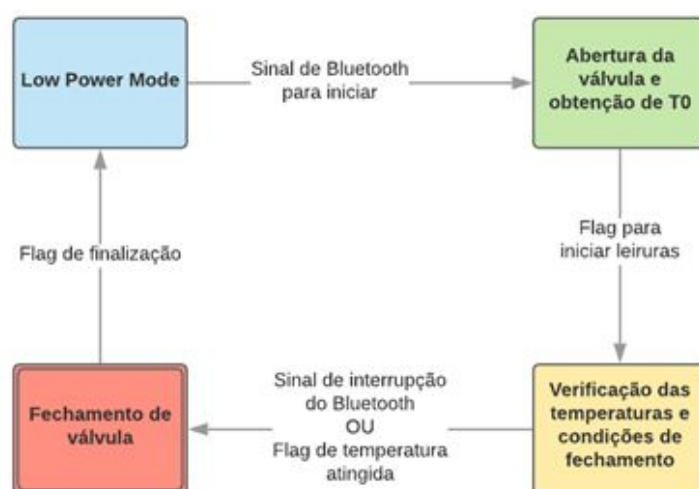
### 1. Resumo

No atual ponto de controle, especificaremos os testes realizados até o momento, no que diz respeito à implementação do projeto na MSP430, através do ambiente de desenvolvimento integrado Code Composer Studio. Explicando o que foi desenvolvido desde o ponto de controle 3 até o momento.

### 2. Desenvolvimento

#### 2.1 Recapitulação

Foi desenvolvido durante o ponto de controle 3, onde esta etapa está melhor explicada, um código inteiramente em C, tendo como base, a máquina de estados ilustrada abaixo.



**Figura 1: máquina de estados do projeto**

Como citado no próprio ponto de controle, faltava ser desenvolvida uma função "*Send\_Float*", responsável por enviar os dados de temperatura com uma maior precisão para a interface com o usuário, já que estavam sendo enviados apenas o número inteiro referente a esta temperatura. Além disso, foi citada a possibilidade deste código ser

facilmente otimizado, a partir da implementação de funções críticas para o projeto em assembly, para realizar tarefas em nível de máquina, portanto, otimizando estas etapas do código. Estes foram os pontos focados durante este ponto de controle no que diz respeito a software.

Além disso, foi citado que seria iniciada a preparação da estrutura física para o protótipo do sistema de abastecimento de motores de foguetes, o que também foi iniciado durante este ponto de controle.

## 2.2. Ponto de Controle 4 - Software

Primeiramente, durante a parte de software, foi desenvolvida a função responsável por enviar variáveis do tipo float, no caso, a função *“Send\_Float”*, que consiste em uma função que recebe uma variável do tipo float e a separa em uma variável contendo apenas sua parte inteira, e outra parte que contém apenas sua parte fracionária com duas casas de precisão, o código se utiliza das funções *“Send\_Int”* e *“Send\_String”*, já explicadas no ponto de controle 3, para enviar a parte inteira, após isso, um ponto, e então enviar a parte fracionária para que o usuário receba o resultado como sendo uma variável do tipo float com duas casas de precisão. Há ainda, o caso particular de uma variável negativa, a função também o identifica e envia, caso necessário, um sinal de negativo, antes de enviar os valores.

Após isso, foram desenvolvidas em assembly, duas sub rotinas críticas para o projeto, que anteriormente estavam sendo realizadas em C, sendo a primeira delas a função atraso, utilizada constantemente para o controle de velocidade de rotação do motor, e atraso entre as leituras de temperatura. A função em assembly, continua recebendo uma variável do tipo inteiro, e produzindo um atraso em ms, igual a esta variável, porém agora esta função se chama *“delay”*, para manter o padrão das demais funções com nomes em inglês.

Além disso, foi desenvolvida a função *“convert”*, responsável por realizar a conversão AD dos dados obtidos pelo sensor de temperatura. Esta função porém, apenas realiza a conversão AD e passa o valor para uma variável, e não deve ser confundida com a função *“leitura”* desenvolvida no código em C. Já que, para se obter o dado referente a temperatura, este valor será posteriormente tratado a fim de se obter o valor de temperatura correto. Tal função não foi reproduzida completamente em assembly, pois o tratamento para se obter o valor real de temperatura envolve operações de complexa implementação em assembly, como multiplicação e divisão com ponto flutuante, além disso, a função leitura utiliza as funções desenvolvidas para o módulo bluetooth já realizadas em C para o envio dos dados de temperatura obtidos, o que também traria grande dificuldade para sua implementação em assembly.

## 2.3. Ponto de Controle 4 - Estrutura

A estrutura do projeto se resume a um aparato puramente demonstrativo, o qual demonstra as principais funções do controlador com suas devidas adaptações à situação de teste. Ela consiste em uma placa de madeira vertical pregada a outra menor, a qual serve de apoio. Nela estão presos uma vasilha e um recipiente plástico no qual será

depositado água com gelo. Neste recipiente plástico foi implantado uma torneira, a qual será acoplado o motor de passo. A idéia é permitir que o motor gire a válvula da torneira, permitindo a passagem da água gelada para o outro pote, no qual encontra-se o sensor de temperatura para detecção da variação de temperatura, ativando o fechamento da válvula.

Atualmente, a estrutura está em cerca de 75% terminada, restando apenas o acoplamento do motor e alocação da parte eletrônica - na região traseira da placa vertical, assim como alguns outros detalhes e teste final. Abaixo segue imagens da estrutura no atual estágio:



**Figura 2: foto frontal da estrutura do projeto**



**Figura 3: foto traseira da estrutura do projeto**

### **3. Dificuldades Encontradas**

Basicamente não houveram grandes contratempos para essas últimas atividades. O próximo trabalho que requerirá atenção será a conclusão da estrutura de demonstração, assim como realizar a integração da eletrônica com o aparato. Posteriormente, serão necessários alguns testes para encontrar os melhores valores de variação para uma demonstração adequada.

### **4. Conclusão**

A parte de software do código que já havia quase que por completo sido desenvolvida em C, foi agora concluída com o desenvolvimento da função "Send\_Float". E o código que estava completamente em C, foi otimizado a partir do desenvolvimento das funções em "delay" e "convert", que eram funções repetidamente utilizadas pelo projeto, o que traz grande otimização no que diz respeito a velocidade e consumo e todas as demais qualidades características da linguagem assembly.

Quanto à estrutura de demonstração do projeto, está está pronta na ordem de 75%, com todos os materiais comprados e disponíveis para finalização. Ela servirá apenas para exposição das funcionalidades do projeto, não tendo utilidade na função fim do mesmo.