



FUNDAÇÃO UNIVERSIDADE FEDERAL DO ABC

Edital N° 01/2017

**Especificações técnicas de sistemas de controle e aquisição de
dados para bancada de testes estáticos**

Aluna: Bárbara Antonia Serejo Mayrink (RA: 21029615)

Orientadora: Thaís Maia Araújo (UFABC)

Co-orientador: Leonardo Henrique Gouvêa (ITA)

Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas

São Bernardo do Campo 2018

Sumário

- 1 – Resumo do projeto
 - 2 – Sistemas propulsivos
 - 3 – Definição: bancada, motor e sensores de instrumentação
 - 3.1 – Termopar
 - 3.2 – Célula de carga
 - 3.3 – Transdutor de pressão e medidor de vazão
 - 3.4 – Sistema de ignição, eletroválvulas e câmara de combustão
 - 4 – Arduino e módulos amplificadores
 - 5 – Aquisição de dados
 - 5.1 – Temperatura
 - 5.2 – Empuxo
 - 5.3 – Pressão e vazão mássica
 - 6 – Sequência lógica do funcionamento e coleta de dados
 - 7 – Orçamento
 - 8 – Conclusão
- REFERÊNCIAS
- APÊNDICE A: Código para a aquisição de dados
- APÊNDICE B: Bibliotecas classes

1 – Resumo do projeto

O objetivo do presente trabalho foi realizar um estudo teórico e detalhado para determinar as especificações técnicas de sistemas de controle e aquisição de dados de instrumentos que deverão ser instalados em bancada de escala reduzida para testes estáticos de propulsores a jato ou câmaras de combustão de pequenos portes. Temperatura, pressão, empuxo do motor testado, vazão mássica dos combustíveis estão entre os principais parâmetros de desempenho propulsivo, portanto os sensores selecionados para realizar as medidas dessas grandezas foram, respectivamente, termopares, transdutores de pressão, célula de carga, medidores de vazão. Foi realizada uma cuidadosa seleção dos modelos e faixas de operação, de forma que os sensores selecionados pudessem ser facilmente encontrados no mercado nacional a custo acessível, para possibilitar uma futura instrumentação de bancada. Para o registro dos dados enviados pelos sensores foi selecionado o microprocessador Arduino, cujo modelo foi determinado baseando-se no número de entradas analógicas e digitais adequadas ao projeto. Um código de programação para aquisição e armazenamento de dados captados pelos sensores foi construído com linguagem de programação simples. Após a tipificação de todos os instrumentos essenciais, foi criado o diagrama de blocos descrevendo a sequência lógica do funcionamento e localização dos sensores de coleta de dados para um teste hipotético.

2 – Sistemas propulsivos

Um conceito amplo de propulsão é o ato de mudar o movimento de um corpo. Mecanismos de propulsão oferecem a força que movimenta os corpos que estão inicialmente em repouso, mudam a sua velocidade ou superam forças retardadoras quando o corpo é propelido através de um meio.

Os sistemas propulsivos por foguete podem ser classificados de acordo com o tipo de fonte de energia: química, nuclear, solar ou exóticas. De forma geral, a propulsão por foguete utiliza a energia liberada em reações químicas de combustão, ao interagir combustível e oxidante num

sistema de ignição adequado para o aparato [1]. No que se diz respeito à propulsão química, os sistemas propulsivos podem ser subdivididos em três tipos de propelentes principais: líquido, sólido e híbrido.

Os sistemas propulsivos que fazem uso de propelente sólido são os mais antigos e mais simples de serem projetados, porém uma desvantagem é que, a partir do momento em que a queima começa, não é possível controlá-la até que todo o suprimento de combustível acabe, cujo manuseio é considerado o menos seguro dentre os sistemas propulsivos [2].

Já os sistemas propulsivos que empregam propelente líquido são mais recentes, e, ao contrário dos foguetes à base de propelente sólido, apresentam capacidade de controle de empuxo bem como múltiplas partidas, sendo normalmente mono ou bi-propelentes [3].

Por fim os foguetes à base de propelentes híbridos são aqueles cujo o combustível e o oxidante são armazenados separadamente e, normalmente, em diferentes fases, apresentando características tanto dos propelentes líquidos quanto dos sólidos. Algumas das principais vantagens e desvantagens de propelente híbrido são apresentadas na Tabela 1 [4]. Pelas vantagens de segurança e baixo custo que o sistema híbrido oferece, este tipo de propelente parece ser atrativo. As desvantagens não chegam a ser problemas, pois como a proposta da bancada deste projeto é para fins educativos, os testes devem ser para motores de pequeno porte e o balanceamento da razão de mistura (oxidante + combustível) e a precisão das curvas de empuxo são responsabilidades do desenvolvedor do sistema propulsivo.

Tabela 1 - Vantagens e desvantagens do sistema propulsivo híbrido

Vantagens	Desvantagens
Segurança na fabricação, estocagem e operação com menores riscos de explosão ou detonação	Não há garantia para o funcionamento em motores de grande escala
Sistemas relativamente baratos	É necessária uma mistura precisa para que não haja variação no impulso específico ao longo do processo

Embora a opção do presente trabalho seja para o desenvolvimento de uma bancada de teste para sistemas propulsivos híbridos é possível, por meio de pequenas alterações, principalmente, nas

linhas de alimentação, testar motores à base de propelentes líquidos monopropelentes e sólidos. Portanto no código estabelecido para o microprocessador de sinais tais considerações serão levadas em conta, mas não na parte de instrumentação, a qual será voltada apenas para propelente híbrido.

Bancadas de testes servem para a análise, investigação e demonstração da evolução temporal de diferentes parâmetros propulsivos, dentre eles o empuxo gerado pela ejeção dos gases produzidos na combustão do propelente e ejeção com velocidade supersônica pela tubeira [5, 6, 7]. Esta medição será realizada por meio de uma célula de carga.

Pode-se separar os parâmetros de medidas em dois grupos, os termodinâmicos e os mecânicos. No caso dos termodinâmicos diz-se que os gases provenientes da reação de combustão dos propelentes são ejetados com velocidades supersônicas por um bocal de Laval (convergente-divergente), também chamado de tubeira. Em altas temperaturas e pressões, reações de combustão carecem de análises teórico-experimentais, de modo que, as influências sobre o sistemas propulsivo não possam ser totalmente avaliadas [2]. Medidas de pressão podem ser realizadas por transdutores de pressão, os quais serão avaliados para serem adaptados à bancada.

No caso dos parâmetros mecânicos é válido analisar maneiras para que a temperatura de tais gases não derreta as estruturas essenciais do aparato, ou motor, e como resfriar estas regiões após a passagem destes gases. De acordo com o estado físico do propelente existe uma classe de equipamentos de medida que auxiliam o bom funcionamento de testes e missões [3]. Na bancada em questão, a temperatura será medida com o auxílio de termopares, que deverão ser escolhidos dependendo do intervalo necessário.

3 – Definição: bancada, motor e sensores de instrumentação

Um valor médio de potência dos propulsores, que possam ser seguramente utilizados e operados em laboratórios didáticos e/ou de pesquisas acoplados à uma bancada, foi definido ser na ordem de 5N [8]. A estrutura mecânica, onde o propulsor é acoplado, foi adotada de Tarifa et al. [9]. Desta maneira, as especificações e definições dos modelos e faixa de operação de sensores,

assim como a localização da instalação, foram pensadas para sistemas híbridos, mas com pequenas alterações, motores à base de propelentes líquidos monopropelentes e sólidos, também poderão ser utilizados sem necessidade de alteração no código estabelecido para o microprocessor.

Definido o empuxo médio de operação dos propulsores, que são acoplados à estrutura mecânica, deu-se início às determinações dos modelos, faixas de operação e localizações da instalação dos sensores de instrumentação, que foram baseadas em um extenso levantamento bibliográfico e pesquisas de mercado. Termopares, transdutores de pressão, célula de carga, medidores de vazão estão entre os principais sensores que foram considerados para serem acoplados à bancada para coleta de dados de temperatura, pressão, empuxo do motor testado, vazão mássica dos combustíveis, respectivamente. Convém mencionar que os sensores foram cuidadosamente escolhidos para que pudessem ser facilmente comprados e encontrados no mercado nacional, cujas justificativas de escolhas são esclarecidas no decorrer deste relatório. Portanto, uma configuração geral, indicando os pontos em que os sensores deverão ser fixados, é mostrada no desenho esquemático da Figura 1, desenvolvido na plataforma Canva fluxograma. Medidas de temperaturas deverão ser coletas na câmara, nas pré e pós câmaras de combustão por meio de termopares. O valor do empuxo deverá ser medido pela célula de carga e dados de pressão também devem ser coletados das pré e pós câmaras de combustão. Fazem parte do sistema de ignição as eletroválvulas que são responsáveis pela liberação e fechamento dos combustíveis.

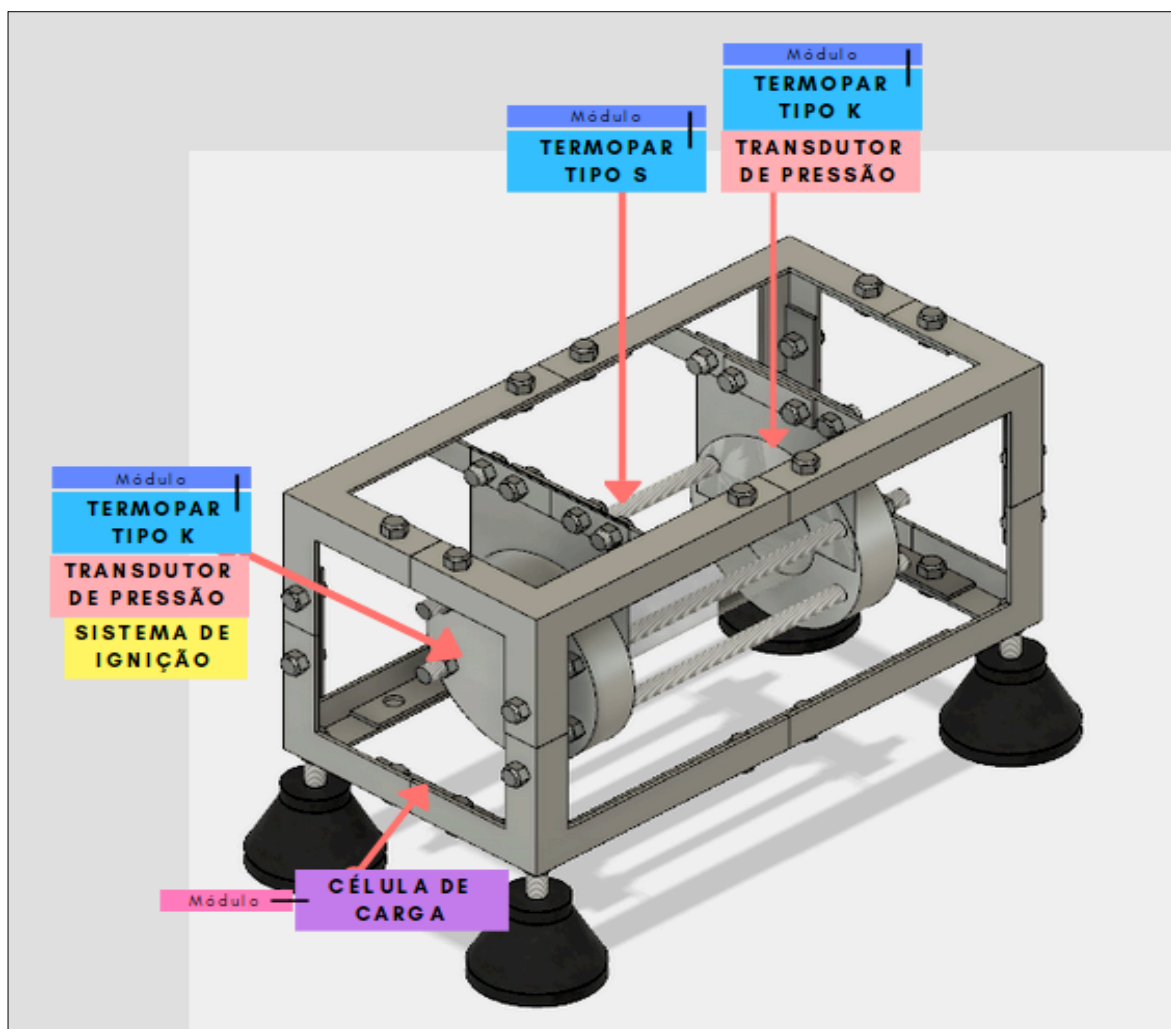


Figura 1 – Localização dos principais sensores na bancada de propulsor híbrido.

3.1 – Termopar

A fim de que seja possível caracterizar a eficiência propulsiva do propelente e do propulsor testados, é essencial que medidas de temperaturas sejam tomadas. Nos processos de combustão, que são altamente energéticos, as temperaturas alcançadas são acima das suportadas por termômetros tradicionais e economicamente mais acessíveis, sendo nestes casos, necessário que sejam realizadas medidas indiretas de temperatura utilizando os termopares.

Termopares são termômetros de resistência elétrica (RTD) que se baseiam no princípio da variação da resistência elétrica dos metais com a temperatura. Os RTDs são extremamente

precisos, possuem excelente repetitividade de leituras, têm resposta linear e se diferenciam pelos materiais utilizados na construção, sendo os tipos mais conhecidos: J, K, E, S e T [10]. Para a especificação destes sensores é necessário conhecer os materiais que compõe o par bimetálico, a faixa de operação [°C] e a sensibilidade do sensor [$\mu\text{V}/^\circ\text{C}$].

Para o uso neste trabalho foram selecionados dois termopares do tipo K para serem instalados na pré e pós câmara de combustão e um do tipo S para instalação dentro do motor no meio da câmara, ambos podem ser vistos a direita e a esquerda, respectivamente, na Figura 2. Os termopares do tipo K, par bimetálico Cromel (Cromo-Níquel) e Alumel (Alumínio-Níquel) são os com custos mais acessíveis para a faixa de operação que abrange temperaturas de -200°C até 1350°C e possuem sensibilidade de, aproximadamente, $40\mu\text{V}/^\circ\text{C}$. Os termopares do tipo S, par bimetálico Platina e Ródio-Platina são sensores protegidos por tubos de cerâmica ou porcelana, com faixa de operação abrangendo temperaturas de -50°C até 1758°C (ponto de fusão da platina) e com sensibilidade de, aproximadamente, $15\mu\text{V}/^\circ\text{C}$ [11].



Figura 2 - Termopares tipo K (direita) e tipo S (esquerda) [12].

3.2 – Célula de carga

A célula de carga é um dispositivo eletromecânico que transforma, analogicamente, deformações mecânicas, por exemplo, fadiga, em sinais elétricos, que precisam ser amplificadas para serem lidas por um microprocessador (*datalogger*).

Existem variações em tamanho e material das células de carga, e como o projeto é testar propulsores de com empuxo médio de 5 N (Newtons), foi decidido adotar um sensor com limite acima do necessário. Portanto, a célula de carga 108PA do tipo *Single Point* para 1Kg, i.e., 10 N,

foi escolhida. Apresentada na Figura 3, tal célula possui limite de medidas de 10VCC à 15VCC (Volts em corrente contínua) e erro combinado, ou seja, o desvio máximo entre a reta de referência e a curva de medição, incluindo os efeitos de não linearidade e repetição, de $1\text{mV/V} \pm 20\% \text{ FS}$, em que FS significa *Full Scale* [13].



Figura 3 - Célula de carga modelo 108PA para 1Kg [14].

O posicionamento da célula de carga durante o teste influencia muito em como esta deverá ser previamente calibrada. A calibração de uma célula de carga é feita com o uso de uma máquina de ensaios universal, regida por equações lineares, cujo procedimento segue as etapas de 1 a 3:

1. Sem que haja qualquer aplicação de força, a célula é ligada e testada ($F=0\text{N}$), retornando um valor x arbitrário que deverá ser anotado: $0\text{N} \Rightarrow x$
2. Após obter o valor x , aplica-se uma força conhecida sobre a célula ($F = \alpha\text{N}$), neste caso devolve um valor y arbitrário: $\alpha\text{N} \Rightarrow y$
3. Com os dois valores de forças que foram aplicadas na célula, 0 e α , e com os dois valores devolvidos, x e y , é possível, por meio de uma equação linear, associar quaisquer valores devolvidos pela célula.

3.3 – Transdutor de pressão e medidor de vazão

Um transdutor de pressão, também chamado de transmissor de pressão, é um instrumento que converte a informação da pressão em um sinal elétrico, neste caso, analógico. Tal conversão é alcançada pela deformação física do sensor ligado ao diafragma do transdutor de pressão e

conectados em uma configuração de ponte de Wheatstone. A pressão aplicada ao transdutor produz uma deflexão do diafragma, que causa deformação aos sensores. A deformação produzirá uma alteração de resistência elétrica proporcional à pressão [15].

Neste projeto foi especificado um transdutor de pressão para ser aplicado em diversos pontos da bancada de testes, respeitando a faixa de pressão que o motor alcança, que para o caso do presente projeto, foi um transdutor de pressão convencional e adequado para a situações gerais de baixa até relativamente altas pressões. Como o objetivo principal é obter dados referentes aos fenômenos no interior do motor, então foram tipificados transdutores para medir a pressão no interior da câmara de combustão, o que se faz necessário que o motor seja perfurado e, através de um pequeno tubo reforçado será medida a pressão de estagnação dentro da câmara.

Além da faixa de pressão de operação, cuidadosamente definida para evitar que aconteçam picos de pressão, é importante também especificar a natureza do sinal de saída para que haja adequação ao *datalogger*, portanto, foi escolhido o transdutor MPM4150, indicado na Figura 4, com faixa de operação de 0 a 16 bar (0 a 232 psi) e sinal analógico de saída de 4mA a 20mA [16].



Figura 4 - Transdutor de pressão MPM4150 [16]

Já o medidor de vazão é um dispositivo que permite determinar o volume de um fluido que passa através de uma seção de escoamento por unidade de tempo [17]. Um medidor de vazão convencional tem como benefícios a alta precisão e apresentação de dados nas unidades desejadas. Os principais tipos deste sensor são os eletrônicos, analógicos e tipo Doppler, cada um com funcionamento específico.

Contudo, os medidores de vazão oferecidos no mercado apresentam alto custo e encareceram demais o presente projeto, o que levou à busca de uma alternativa mais econômica que resultou na escolha do tubo de venturi, representado na Figura 5. As pressões do fluido nas seções (1) e (2) são medidas por dois transdutores de pressão instalados. Assim é possível calcular as velocidades do escoamento incompressível, e pelos princípios de Bernoulli e da continuidade da massa é possível determinar vazão volumétrica e vazão mássica do escoamento, conhecendo a massa específica do fluido [18]. Para a bancada projetada serão utilizados dois tubos de venturi, um para medidas de pressão dentro do motor e outro para as pressões do oxidante, consequentemente, quatro transdutores de pressão serão necessários.

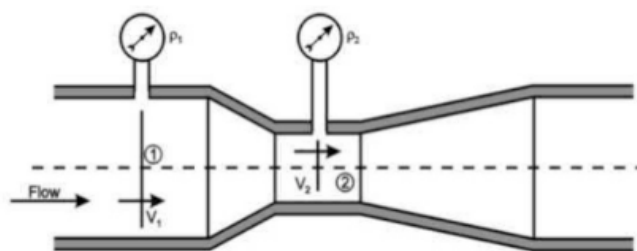


Figura 5 - Esquema do tubo de Venturi com dois transdutores de pressão [18].

3.4 – Sistema de ignição, eletroválvulas e câmara de combustão

Ignição é o mecanismo, ou processo, de início da combustão do propelente. Cada tipo de propelente requer um sistema de ignição adequado, entretanto, é de se esperar que a maior parte dos combustíveis sofram combustão ao entrar em contato com ar e calor [4]. A ignição ocorre na pré câmara de combustão.

Eletroválvula, ou válvula solenóide, é um dispositivo controlado eletromecanicamente, contendo um solenóide com um núcleo ferromagnético no seu interior. Este núcleo ferromagnético atua como êmbolo que, na posição de repouso fecha um pequeno orifício de saída. Em consequência da passagem de corrente elétrica é gerado um campo eletromagnético e este exerce uma força sobre o êmbolo. Fora da posição de repouso este êmbolo permite a passagem do

propelente ou combustível para a linha de alimentação [8]. Para a bancada desenvolvida são necessárias duas eletroválvulas, pois trata-se de um sistema para testes com combustíveis híbridos que, normalmente, são armazenados separadamente.

A câmara de combustão para o caso aqui estudado é o próprio propulsor, pois o sistema híbrido escolhido foi o que utiliza como combustível o PPMA (polipropileno) que, quando em contato com oxidante, igniza e inicia a combustão.

4 – Arduino e módulos amplificadores

O Arduino é uma plataforma eletrônica de hardware livre e de placa única, de custo relativamente baixo e bastante acessível no mercado. A praticidade da plataforma Arduino garante, por meio de uma linguagem de programação simples, que o projeto da bancada seja ainda mais acessível para testes variados. Logo, será usado a plataforma Arduino como *datalogger* e será necessária a construção de um código de programação para aquisição e armazenamento de dados captados pelos sensores da bancada de testes. Existem vários tipos de Arduino - NANO, MICRO, UNO, MEGA, etc. - sendo uma das principais diferenças o número de portas lógicas digitais e analógicas. Portanto, para que o número de portas estimado neste projeto fosse atendido, o Arduino MEGA 2560 foi considerado o mais adequado, pois possui 16 entradas analógicas e 54 digitais. Na Figura 6 é disponibilizada uma imagem da placa do arduino, de modo que as siglas dos pinos de entradas e saídas podem ser observadas e, na Tabela 2, é apresentada a associação das siglas com a descrição das características funcionais dos pinos.

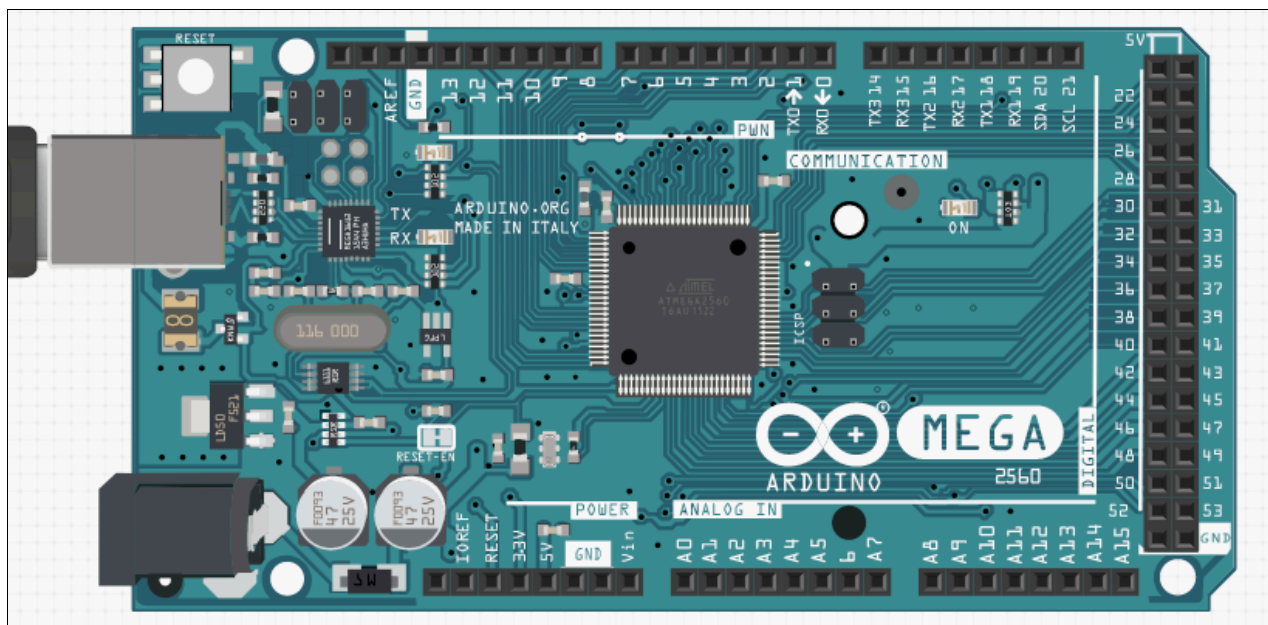


Figura 6 - Arduino MEGA 2560 [19].

Tabela 2: Características funcionais relacionadas às siglas dos pinos do Arduino MEGA 2560.

Arduino MEGA 2560	
Pinos de alimentação	Característica
IOREF	Fornece uma tensão de referência para que <i>shields</i> possam selecionar o tipo de interface apropriada.
Reset	Pino conectado a pino de RESET-EN do microcontrolador.
3.3V	Fornece tensão de 3,3 V com corrente máxima de 50 mA.
5V	Fornece tensão de 5 V para alimentação de <i>shields</i> e circuitos externos.
GND	Pinos de referência, <i>ground</i> ou terra.
Vin	Pino para alimentar a placa por meio de <i>shield</i> ou bateria externa.
Demais pinos	Característica
A0 ao A15	Analógicos
22 ao 53	Digitais
20 (SDA) e 21 (SCL)	TWI
Serial 0 (RX) e 1 (TX) Serial 1: 19 (RX) e 18 (TX) Serial 2: 17 (RX) e 16 (TX) Serial 3: 15 (RX) e 14 (TX)	Comunicação Serial
2 a 13 e 44 a 46	Saídas PWM (Modulação por Largura de Pulso)

Módulos amplificadores têm a função de aumentar a amplitude dos sinais que são emitidos pelos sensores e processados pelo *datalogger*. Tais módulos garantem uma leitura mais versátil dos

dados sem alterar a informação contida de cada sensor instalado. De modo mais técnico pode se dizer que a derivada da taxa de variação do sinal amplificado deve ser um múltiplo inteiro da derivada do sinal original assim como as raízes de ambos devem se manter iguais. Foram adotados módulos amplificadores para os sensores célula de carga e termopares.

Para o sinal enviado pelo sensor da célula de carga foi escolhido o módulo amplificador ICA 5S analógico, representado na Figura 7. O módulo amplificador digital MAX31856, Figura 8, foi escolhido para ampliar os sinais de temperaturas fornecidos pelos termopares tipos K e S. Este módulo necessita ser conectado em quatro entradas digitais específicas do *datalogger*. Cada um dos três termopares especificados para a bancada terá o seu respectivo módulo MAX31856 e, desta forma, serão necessárias 12 entradas digitais.

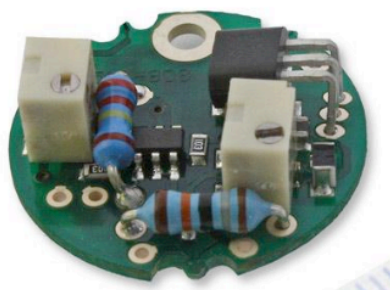


Figura 7 -Módulo amplificador ICA 5S para a célula de carga [20].

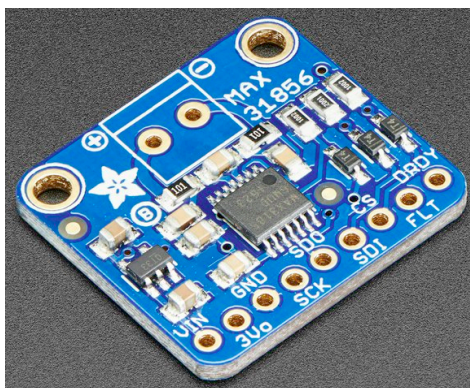


Figura 8 -Módulo amplificador modelo MAX31856 para termopares [21].

Resumidamente são especificadas, na Tabela 3, as quantidades e modelos dos sensores e módulos amplificadores, com a indicação dos tipos de sinais de saída, analógico ou digital. Observa-se que para compor o projeto da bancada foram selecionados sete transdutores de pressão, sendo quatro destes direcionados para a medida indireta de vazão mássica, um sensor célula de carga, quatro módulos amplificadores e três termopares.

Tabela 3 - Sinal de saída para cada tipo de sensor

Sensor	Modelo	Sinal de saída	Qnt.	Saídas necessárias
Transdutor de pressão	MPM4150	Analógico	7	7 analógicas
Célula de carga	108PA	Analógico	1	1 analógica
Módulo amplificador para a célula de carga	ICA 5S	Analógico	1	Nenhuma
Termopar	Tipo K	Não se aplica	2	Nenhuma
	Tipo S		1	
Módulos amplificadores para os termopares	MAX31856	Digital	3	12 digitais PWM

Deve-se prestar atenção também no tamanho dos cabos entre os analógicos e o microprocessador, pois quanto menor for o fio menor será o ruído da captação de dados. Para isto usa-se cabos coaxiais blindados.

5 – Aquisição de dados

Para a aquisição de dados foi construído um programa, definindo como fazer a coleta dos dados enviados pelos sensores tipificados e como armazená-los para posterior leitura dos parâmetros medidos. No Apêndice A pode ser verificado os passos do programa construído, lembrando que os parâmetros medidos foram temperatura, empuxo e pressão.

5.1 – Temperatura

Para a aquisição de dados de temperatura foram utilizadas as 12 saídas PWM's disponíveis no Arduino MEGA, da 2 até 13 indicadas com preenchimentos na Figura 9, para os três termopares

especificados. Estes pinos são entradas digitais que têm como função a modulação ou alteração da largura do pulso de um sinal de onda quadrada.

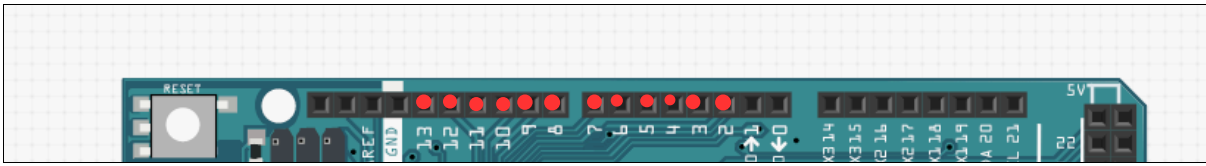


Figura 9 — Pinos do arduino MEGA 2560 utilizados para aquisição de dados de temperatura.

Para o funcionamento do módulo amplificador digital MAX31856 devem ser adicionadas à biblioteca do arduino os códigos `Classes.h` e `Classes.cpp`, disponibilizadas nos Apêndices B1 e B2, respectivamente; e também a adição da biblioteca `Adafruit_MAX31856.h` para que o código principal possa funcionar [22].

Antes de inicializar o programa, as variáveis precisam ser identificadas, assim o termopar do tipo K (da pré-câmara) será conectado nas PWM's 10, 11, 12 e 13, o outro termopar do tipo K (da pós-câmara) será conectado nas PWM's 9, 8, 7 e 6 e, por fim, o termopar do tipo S (do meio da câmara) será conectado nas PWM's 5, 4, 3 e 2. O próximo passo é a inicialização de cada um dos tipos necessários de termopar, isto é, `MAX31856_TCTYPE_K` para cada um todos termopares do tipo K e `MAX31856_TCTYPE_S` para o termopar do tipo S. Para cada um dos três termopares o programa será rodado da mesma forma, por mais que estes sensores estejam em localizações diferentes. Durante o loop do programa serão computados valores de temperatura e as falhas associadas para cada um todos termopares, separadamente [22].

5.2 – Empuxo

Para a aquisição de dados de empuxo, utilizando a célula de carga 108PA, foi utilizado o pino A0 do arduino MEGA 2560, identificado na Figura 10. Um ponto importante sobre a célula de carga é que por estar posicionada perpendicularmente à bancada de testes a gravidade não influencia na calibração deste sensor.

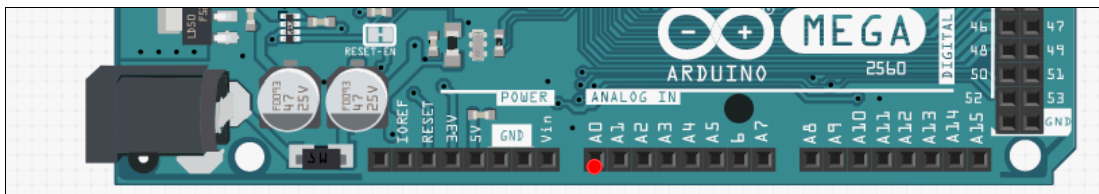


Figura 10 – Pino do arduino MEGA 2560 utilizados para aquisição de dados de empuxo.

A aquisição de dados de empuxo é iniciada com a leitura dos valores de tempo, valor devolvido pela célula e valor em quilogramas (corrigido). Quando o programa está rodando em *loop* é lido o valor devolvido pela célula e o valor em quilogramas (corrigido) com seu tempo associado (em intervalos de 0,01s).

5.3 – Pressão e vazão mássica

Para aquisição de dados de pressão foram utilizados sete pinos analógicos, do A1 ao A7, conforme representado na Figura 11.

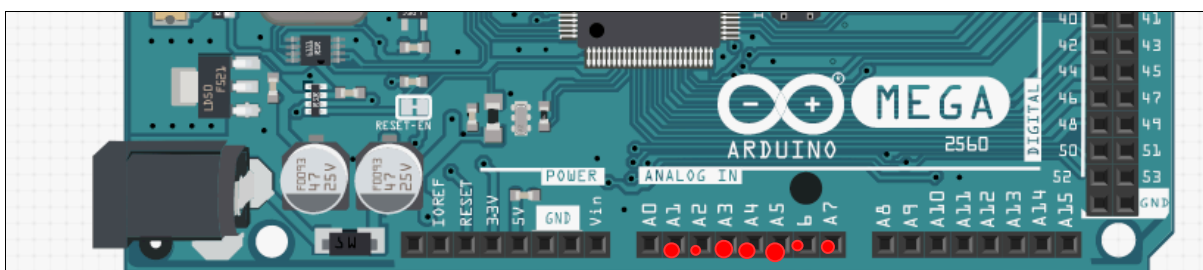


Figura 11 – Arduino MEGA 2560 – pinos utilizados para aquisição de dados de pressão.

Para aquisição destes dados não é necessária a adição de nenhuma biblioteca adicional mas precisa da identificação das variáveis. Os dois transdutores de pressão acoplados ao tubo de venturi do combustível ocuparão os pinos A1 e A2, enquanto os transdutores do tubo de venturi do oxigênio ocuparão os pinos A3 e A4. O transdutor presente na linha de alimentação ocupará o pino A5, o transdutor da pré-câmara o A6 e o transdutor da pós-câmara o A7. Os valores elétricos captados pelos pinos do arduino são em forma de tensão devolvida e precisam ser transformados em valores de pressão. Para a correção é utilizada a expressão indicada na Eq. 1, em que o valor 1024

corrige o valor analógico recebido em digital e 232 psi é a capacidade máxima do transdutor de pressão.

$$Valor\ recebido \frac{[V]}{1024} * 232 [psi] \quad (1)$$

Somente após a correção o valor poderá ser apresentado ao usuário.

A aquisição de dados vazão mássica do combustível e oxidante deverão ser calculadas por meio dos valores de pressão lidos pelos dois tubos de venturi, que foram devolvidos em A1-A2 e A3-A4.

6 – Sequência lógica do funcionamento e coleta de dados

Depois de definida toda a instrumentação necessária para a coleta de dados durante teste estático de motor a jato com empuxo médio em torno de 5N, foram definidos os modelos e faixas de operação adequados para cada sensor selecionado. Como resultado final é descrita a sequência lógica do funcionamento e localização dos sensores usados para a coleta de dados, identificando os modelos selecionados. Para ajudar na compreensão dos resultados foi criado um diagrama de blocos, utilizando a plataforma Canva fluxogramas, apresentado na Figura 12, que informa os passos da sequência do funcionamento de um teste hipotético na bancada de teste idealizada.

O início do teste se dá com a entrada de combustível e oxidante dentro da linha de alimentação, que é controlada por duas eletroválvulas. Para saber quanto de gás/líquido está entrando no sistema são usados medidores de vazão, que foi definido o tubo de venturi que utiliza dois transdutores de pressão MPM4150 para cada fluido.

Na linha de alimentação o combustível e oxidante se misturam e, para o processo, é interessante saber a pressão de estagnação desta mistura antes da ignição na pré-câmara, a qual é obtida por um transdutor de pressão instalado na linha.

Em seguida a mistura entra na pré-câmara de combustão onde sofrerá os efeitos da combustão. Na pré-câmara, no momento da ignição, ocorrem medidas de pressão por meio do

transdutor MPM4150 e temperatura com termopar tipo K e o módulo amplificador MAX31856. Para a medida de pressão é necessário que um tubo seja soldado perpendicular à parede do motor.

A temperatura no interior da câmara de combustão é medida, de forma indireta, na parede do motor, com o auxílio de um termopar do tipo S e módulo amplificador MAX31856.

Toda a geração de calor dentro da câmara fará com que o empuxo do motor possa ser medido, assim, a célula de carga 108PA e o módulo amplificador ICA 5S estão posicionados atrás do motor, próximos à pré-câmara para que possa ser medido o valor do empuxo do sistema.

Anterior ao leito catalítico (tubeira) existe uma região chamada de pós câmara, onde serão coletadas medidas de temperatura com o termopar do tipo K e módulo, e medidas de pressão por outro transdutor MPM4150. Já no leito catalítico o fluido é ejetado em temperaturas tão altas quanto dentro da câmara de combustão.

Todos os instrumentos essenciais selecionados para o teste, descritos na Figura 11, estão conectados ao *datalogger*, que neste caso é o Arduino MEGA 2560. Tal conexão foi representada com o uso de triângulos que são da mesma coloração dos sensores e módulos que estão agregados à bancada. No Arduino serão armazenados os dados de todo o processo por meio de um cartão SD.

Vale ressaltar que testes em que ocorrem combustão são extremamente rápidos, com duração de poucos minutos, o que requer que durante o levantamento de dados ocorra uma captação contínua para a melhor visualização dos parâmetros dentro e fora da câmara de combustão.

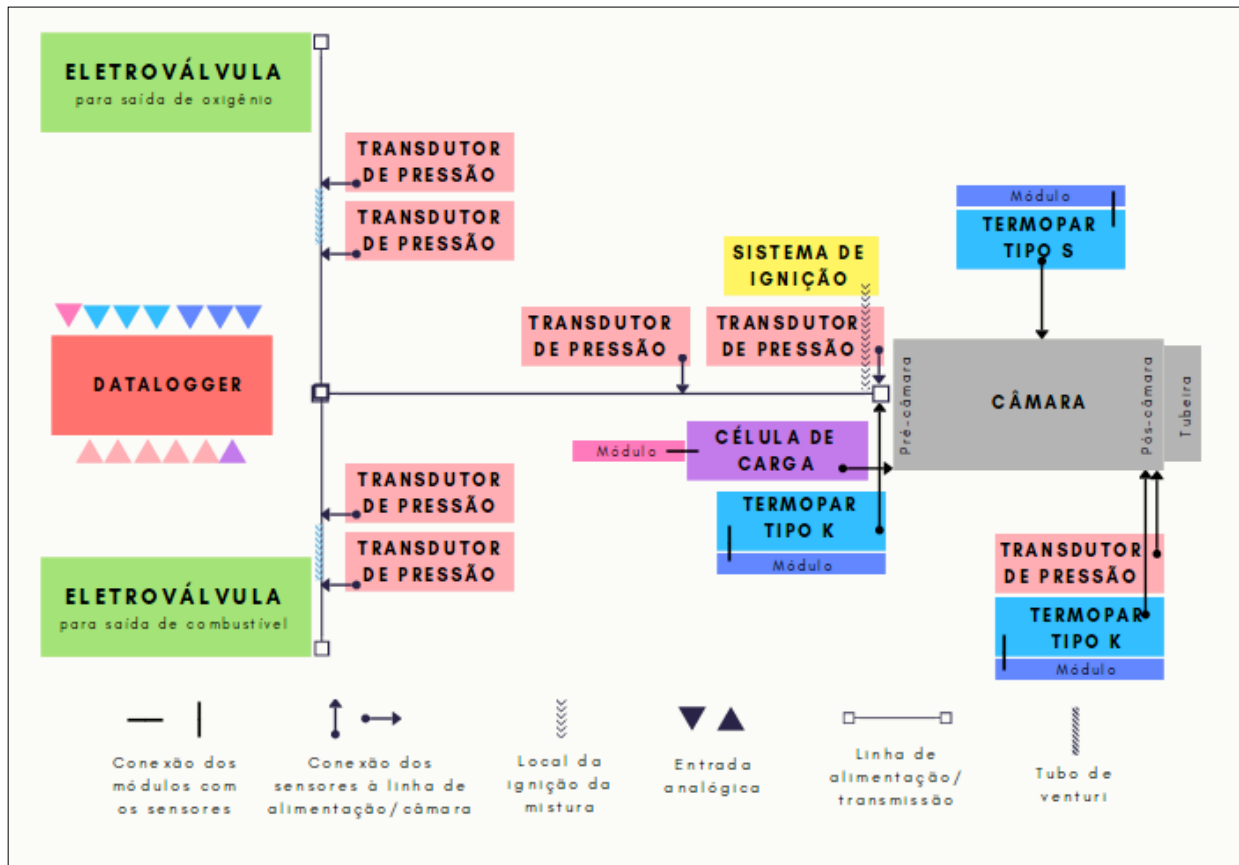


Figura 12 - Blocos orientados representando o funcionamento e localização dos sensores de coleta de dados da bancada de testes.

7 – Orçamento

As decisões sobre a seleção dos sensores para medidas de sistemas propulsivos foram baseadas em aspectos técnicos, além de segurança e custo. Portanto, durante as escolhas dos sensores foi possível realizar o orçamento detalhado, apresentado na Tabela 4. Considerando as quantidades estabelecidas pelo projeto desenvolvido e mencionando que as eletroválvulas, linha de alimentação, sistema de ignição e demais aparatos para conexão dos sensores são de responsabilidade do desenvolvedor da câmara de combustão, o custo total chegou próximo de quatro mil reais .

Tabela 4 - Orçamento dos sensores para o projeto desenvolvido.

Sensor	Modelo	Quantidade	Preço unitário R\$ (reais)	Preço total R\$ (reais)
Transdutor de pressão	MPM4150	7	270,00 [23]	1890,00
Célula de carga	108PA	1	300,00 [14]	300,00
Módulo amplificador para a célula de carga	ICA 5S	1	350,00 [24]	350,00
Termopares	Tipo K	2	75,00 [25]	150,00
	Tipo S	1	900,00 [26]	900,00
Módulos amplificadores para os termopares	MAX31856	3	70,00 [27]	210,00
Arduino	MEGA 2560	1	160,00 [28]	160,00
TOTAL				3960,00

Cotação do dólar em 04 de agosto de 2018: R\$ 3,71

8 – Conclusão

A completa tipificação dos instrumentos necessários para a aquisição de dados de bancada de escala reduzida para testes estáticos de propulsores a jato ou câmaras de combustão de pequenos portes foi realizada e termopares, transdutores de pressão, célula de carga, foram os sensores selecionados.

Concluiu-se que para medidas de temperatura os sensores ideais são os termopares do tipo S e do tipo K. Foram identificadas as necessidades de instalação de dois S e um K, com faixas de operação de -50°C até 1758°C e de -200°C até 1350°C, respectivamente.

Para aquisição de dados de pressão e vazão foi selecionado o transdutor MPM4150 com alcance de até 16 bar. Considerações especiais acerca da instalação desse sensor são indispensáveis, pois é necessária a perfuração do motor.

Célula de carga modelo 108pa, com alcance de 10N, foi o sensor selecionado para as medidas de empuxo, sendo que antes do primeiro uso, a calibração do sensor é essencial.

Especificados os sensores concluiu-se que seria necessário amplificar os sinais captados para temperatura e empuxo, para isto, foram escolhidos os amplificadores ICA 5S e MAX31856, respectivamente.

Para o registro dos dados enviados pelos sensores foi escolhido o microprocessador Arduino MEGA 2560, com 16 entradas analógicas e 54 digitais, considerado o mais adequado para o projeto.

Após as escolhas de todos os sensores, módulos amplificadores e microprocessador, um programa computacional foi elaborado para determinar como armazenar e como fazer a leitura dos dados enviados pelos sensores tipificados, especificando que a aquisição de dados deve ocorrer em intervalos de tempo de 0,01 segundo.

Além de segurança, baixo custo foi outro critério que esteve presente em todas as decisões de escolha dos equipamentos e, por meio de análise do orçamento detalhamento realizado, pode-se concluir que a instrumentação da bancada pode ser executada a um custo inferior a um mil dólares.

REFERÊNCIAS

- [1] SUTTON, G. P. *Rocket Propulsion Elements, An introduction to the Engineering of rockets*. New York: Wiley, 1992. 636.p.
- [2] GLASSMAN, I.; YETTER, R.A. *Combustion*. 4. ed. London: Academic Press, 2008.
- [3] KUO, K. K-Y. *Principles of Combustion*. 2. ed. New York: John-Wiley Interscience, 2005.
- [4] KEATING. E. L. *Applied Combustion*. 2. ed. Maryland: CRC, 2007.
- [5] CONTAIFER, R.A. *Desenvolvimento de uma bancada de testes de propulsores híbridos de baixo empuxo*. Dissertação (Mestrado em Engenharia e Tecnologia Espaciais/Combustão e Propulsão) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2009.
- [6] NEFF, T. REHBERGER, M.; MEROTH, A. *Thrust test bench for student rocket engines*. 2016 11th France-Japan & 9th Europe-Asia Congress on Mechatronics.
- [7] WRIGHT, A. M. et al. *A six degree-of-freedom thrust sensor for a lab scale hybrid rocket*. Measurement Science and Technology, Reino Unido, v. 24, n. 12, 125104, dec. 2013.
- [8] ENGLER, et al. *Building and testing of a lab scale hybrid rocket motor for acquisition of regression rate equations*. 24th ABCM International Congress of Mechanical Engineering. Curitiba. (2017).
- [9] TARIFA, M.C. et al. *Bancada de testes estáticos de pequenos propulsores: balança de empuxo e base de suporte*. Iniciação Científica, UFABC, 2008.
- [10] DE SOUZA, E. *Instrumentação e Sensores de Veículos Aeroespaciais*. 1 ed. 2018.
- [11] PARK, R. M. et al. *Manual on the Use of Thermocouples in Temperature Measurement*. 4. ed. ASTM International.
- [12] Termopares dos tipos K e S. Disponível em:
<<http://www.thermometricscorp.com/thermocouple.html>> Acesso em 16 de agosto de 2018.
- [13] Load cell 1 kg. Single Point. Aluminium. OIML approved. Disponível em:
<<https://www.vetek.com/>> Acesso em 26 de fevereiro de 2017.

-
- [14] AnyLoad 108PA Aluminum Single Point Load Cell. Disponível em:
<<https://tacunasystems.com/zc/load-cells/single-point-load-cells/aluminum/anyload-108pa-aluminum-single-point-load-cell>>. Acesso em 15 de fevereiro de 2018.
- [15] Solenoid Valve Types. Disponível em: <<https://tameson.com/solenoid-valve-types.html>>.
Acesso em 06 de setembro de 2017.
- [16] Transdutor de Pressão MPM4150. Disponível em:
<http://www.microsensorcorp.com/upload/article/180527/5_MPM4150+Pressure+Transmitter_1527411261.pdf>. Acesso em 17 de agosto de 2018.
- [17] MONTEIRO, C. P. NONATO D. POIT. K. *Tubo de Venturi*. UNESP, Sorocaba Disponível em: <<http://www.sorocaba.unesp.br/Home/Extensao/Engenhocas/relatorio-fisikana.pdf>>. Acesso em 15 de janeiro de 2018.
- [18] Esquema do tubo de Venturi com dois transdutores de pressão. Disponível em:
<<https://pt.slideshare.net/shaletks/venturi-meter-47426515>> Acesso em 16 de janeiro de 2017.
- [19] Plataforma Circuito. Disponível em <<https://circuito.io>>. Acesso em 18 de agosto de 2018.
- [20] Módulo Amplificar ICA 5S. Disponível em:
<<http://www.lcmsystems.com/res/ICA5%20SA%20manual.pdf>>. Acesso em 17 de agosto de 2018.
- [21] Módulo Amplificador MAX31856. Disponível em: <<https://www.adafruit.com/product/3263>>.
Acesso em 17 de agosto de 2018.
- [22] Módulo Amplificador MAX31856. Disponível em:
<https://github.com/adafruit/Adafruit_MAX31856/>. Acesso em 19 de agosto de 2018.
- [23] Transdutor de pressao MPM4150. Disponível em: <http://www.microsensorcorp.com/p-MPM4150_Pressure_Transmitter_83.htm>. Acesso em 18 de agosto de 2018.
- [24] Módulo Amplificador ICA 5S. Disponível em: <<https://www.micronmeters.com/product/ica-h-ica-s-miniature-strain-gauge-amplifier-converts-load-cell-to-4-20ma-0-10v>>. Acesso em 18 de agosto de 2018.

[25] Termopar do tipo K. Disponível em: <<https://www.ebay.com/bhp/type-k-thermocouple-wire>>.

Acesso em 19 de agosto de 2018.

[26] Termopar do tipo S. Disponível em: <hotkilns.com/type-s>. Acesso em 19 de agosto de 2018.

[27] Módulo Amplificador MAX31856. Disponível em: <<https://www.amazon.com/Adafruit-Universal-Thermocouple-Amplifier-Max31856/dp/B01LZBBI7D>>. Acesso em 19 de agosto de 2018.

[28] Arduino MEGA 2560. Disponível em: <<https://store.arduino.cc/arduino-mega-2560-rev3>>.

Acesso em 19 de agosto de 2018.

APÊNDICE A

O código completo criado para a aquisição de dados.

//Bibliotecas

```
#include <SPI.h> //Biblioteca necessária para o uso do cartão SD
#include <SD.h> //Biblioteca necessária para o uso do cartão SD
#include "Classes.h"//Biblioteca necessária para o uso do cartão SD
#include <Adafruit_MAX31856.h>
```

//Inicialização do modulo do termopar

```
Adafruit_MAX31856 max1 = Adafruit_MAX31856(10, 11, 12, 13);
Adafruit_MAX31856 max2 = Adafruit_MAX31856(9, 8, 7, 6);
Adafruit_MAX31856 max3 = Adafruit_MAX31856(5, 4, 3, 2);
```

//Pinos

/*Transdutor de pressao*/

```
int inputP1 = A1;
int inputP2 = A2;
int inputP3 = A3;
int inputP4 = A4;
int inputP5 = A5;
int inputP6 = A6;
int inputP7 = A7;
```

/*celula de carga*/

//Calibracao da celula de carga

```
#define __Xi 1 //Valor da celula sem massa
#define __Xf 1000 //Valor da celula com massa
#define __Yi 0 //Deve conter o valor 0
#define __Yf 0.5 //Peso real da massa
#define __dX ((__Xf) - (__Xi)) //Delta X
#define __dY ((__Yf) - (__Yi)) //Delta Y
#define __angC (__dY/__dX) //Coeficiente angular
#define __linC (__angC*__Xi) //Coeficiente linear
#define eq(X) ((__angC*X)- __linC) //Calb. 2018-07-07
```

SDCH SDC(53, "cell"); //cria o objeto SDC (cartão de memória SD) e atribui o cs ao pino 53
Helpful help; //cria o objeto help para utilizar a função eachT

void setup()

```
{
pinMode(LED_BUILTIN, OUTPUT);
digitalWrite(LED_BUILTIN, HIGH);
SDC.begin(); //inicia o SDcard
pinMode(analogPin, INPUT); //configura o pino A0 como entrada
```

```
/*Transdutor de pressao*/

/*Celula de carga*/

if (SDC) {
    digitalWrite(LED_BUILTIN, LOW);
    SDC.theFile.print("tempo");
    SDC.tab();
    SDC.theFile.print("vlr.cel");
    SDC.tab();
    SDC.theFile.print("vlr.Kg");
    SDC.tab();
    SDC.theFile.print("avg.cell"); //Cel filtrado
    SDC.theFile.println();
    SDC.close();
}

/*Termopar*/

Serial.begin (115200);

max1.begin();
max1.setThermocoupleType(MAX31856_TCTYPE_K);
max2.begin();
max2.setThermocoupleType(MAX31856_TCTYPE_K);
max3.begin();
max3.setThermocoupleType(MAX31856_TCTYPE_S);
}
void loop()
{

    // Termopar

    Serial.print("Cold Junction Temp: ");
    Serial.println(max1.readCJTemperature());

    Serial.print("Thermocouple Temp: ");
    Serial.println(max1.readThermocoupleTemperature());

    uint8_t fault1 = max1.readFault();
    if (fault1) {
        if (fault & MAX31856_FAULT_CJRange) {
            Serial.println("Cold Junction Range Fault");
        }

        if (fault1 & MAX31856_FAULT_TCRANGE) {
            Serial.println("Thermocouple Range Fault");
        }

        if (fault1 & MAX31856_FAULT_CJHIGH) {
            Serial.println("Cold Junction High Fault");
        }
    }
}
```

```
}

if (fault1 & MAX31856_FAULT_CJLOW) {
    Serial.println("Cold Junction Low Fault");
}

if (fault1 & MAX31856_FAULT_TCHIGH) {
    Serial.println("Thermocouple High Fault");
}

if (fault1 & MAX31856_FAULT_TCLOW) {
    Serial.println("Thermocouple Low Fault");
}

if (fault1 & MAX31856_FAULT_OVUV) {
    Serial.println("Over/Under Voltage Fault");
}

if (fault1 & MAX31856_FAULT_OPEN) {
    Serial.println("Thermocouple Open Fault");
}

}

Serial.print("Cold Junction Temp: ");
Serial.println(max2.readCJTemperature());

Serial.print("Thermocouple Temp: ");
Serial.println(max2.readThermocoupleTemperature());

uint8_t fault2 = max2.readFault();
if (fault2) {
    if (fault2 & MAX31856_FAULT_CJRANGE) {
        Serial.println("Cold Junction Range Fault");
    }

    if (fault2 & MAX31856_FAULT_TCRANGE) {
        Serial.println("Thermocouple Range Fault");
    }

    if (fault2 & MAX31856_FAULT_CJHIGH) {
        Serial.println("Cold Junction High Fault");
    }

    if (fault2 & MAX31856_FAULT_CJLOW) {
        Serial.println("Cold Junction Low Fault");
    }

    if (fault2 & MAX31856_FAULT_TCHIGH) {
        Serial.println("Thermocouple High Fault");
    }
}
```

```
if (fault2 & MAX31856_FAULT_TCLow) {
    Serial.println("Thermocouple Low Fault");
}

if (fault2 & MAX31856_FAULT_OVUV) {
    Serial.println("Over/Under Voltage Fault");
}

if (fault2 & MAX31856_FAULT_OPEN) {
    Serial.println("Thermocouple Open Fault");
}

}

Serial.print("Cold Junction Temp: ");
Serial.println(max3.readCJTemperature());

Serial.print("Thermocouple Temp: ");
Serial.println(max3.readThermocoupleTemperature());

uint8_t fault3 = max3.readFault();
if (fault3) {
    if (fault3 & MAX31856_FAULT_CJRange) {
        Serial.println("Cold Junction Range Fault");
    }

    if (fault3 & MAX31856_FAULT_TCRANGE) {
        Serial.println("Thermocouple Range Fault");
    }

    if (fault3 & MAX31856_FAULT_CJHigh) {
        Serial.println("Cold Junction High Fault");
    }

    if (fault3 & MAX31856_FAULT_CJLow) {
        Serial.println("Cold Junction Low Fault");
    }

    if (fault3 & MAX31856_FAULT_TCHigh) {
        Serial.println("Thermocouple High Fault");
    }

    if (fault3 & MAX31856_FAULT_TCLow) {
        Serial.println("Thermocouple Low Fault");
    }

    if (fault3 & MAX31856_FAULT_OVUV) {
        Serial.println("Over/Under Voltage Fault");
    }
}
```

```
    if (fault3 & MAX31856_FAULT_OPEN) {
        Serial.println("Thermocouple Open Fault");
    }

}

//Transdutor de pressao
/*
Input e a tensao devolvida pelo sensor
Output e a pressao com o calculo e correção
*/

inputP1 = analogRead(A1);
outputP1 = ((float)inputP1/1024*232);

inputP2 = analogRead(A2);
outputP2 = ((float)inputP2/1024*232);

inputP3 = analogRead(A3);
outputP3 = ((float)inputP3/1024*232);

inputP4 = analogRead(A4);
outputP4 = ((float)inputP4/1024*232);

inputP5 = analogRead(A5);
outputP5 = ((float)inputP5/1024*232);

inputP6 = analogRead(A6);
outputP6 = ((float)inputP6/1024*232);

inputP7 = analogRead(A7);
outputP7 = ((float)inputP7/1024*232);

//celula de carg
cel = analogRead(analogPin); // lê o valor retornado pela célula de carga
tempo = float(micros()) / 10000.0;
kg = eq(cel);    //converte o valor para Kg

if (SDC) {
    digitalWrite(LED_BUILTIN, LOW);
    SDC.theFile.print(tempo, 5);
    SDC.tab();
    SDC.theFile.print(cel);
    SDC.tab();
    SDC.theFile.print(kg);
    SDC.tab();
    SDC.theFile.println();
    SDC.close();
}
digitalWrite(LED_BUILTIN, HIGH);
}
```

APÊNDICE B

As classes.h e classes.cpp que estão descritas em B1 e B2, respectivamente, foram adicionadas à biblioteca do Arduino para que o código principal possa funcionar de forma efetiva.

B1: Classes.h

```
#include <SPI.h>
#include <SD.h>

class Helpful
{
    unsigned long conEach = 0;
    unsigned long timerEach = 0;
    unsigned long lapseT = 0;
    unsigned long T0 = 0;
    unsigned long endT = 0;
    bool forTstate = 0;
    bool one = 0;
    unsigned long count = 0;
    float maxi = 0, mini = 0;
public:
    bool mem;
    void begin();
    unsigned long counter();
    unsigned long getCount();
    unsigned long counterReset();
    bool eachN(const unsigned int &N);
    bool eachT(const float &T);
    void forT(const float &T);
    bool forT();
    float lapse();
    float sinceBegin();
    bool oneTime();
    void oneTimeReset();
    void comparer(const float &n);
    float getMax();
    float getMin();
};

class SDCH
{
    const uint8_t CS;
```

```
const String Fname0;
const String Ftype;
const uint8_t coef;
const unsigned long nMax;
unsigned long number = 0
```

B2: Classes.cpp

```
#include "Classes.h"
```

```
///Coleao de utilitarios
```

```
void Helpful::begin()
{
    T0 = micros();
}
unsigned long Helpful::counter()
{
    return count++;
}
unsigned long Helpful::getCount()
{
    return count;
}
unsigned long Helpful::counterReset()
{
    unsigned long C = count;
    count = 0;
    return C;
}
bool Helpful::eachN(const unsigned int &N)
{
    conEach++;
    return (N == 0) ? false : conEach%N == 0;
}
bool Helpful::eachT(const float &T)
{
    if (micros() - timerEach > (unsigned long)(T*1000000.0))
    {
        timerEach = micros();
        return true;
    }
    return false;
}
void Helpful::forT(const float &T)
{
    endT = micros() + (long)(T*1000000.0);
    forTstate = 1;
}
bool Helpful::forT()
```

```

{
    if (micros() > endT && forTstate)
    {
        endT = 0;
        return forTstate = 0;
    }
    return forTstate;
}

float Helpful::lapse()
{
    unsigned long tnow = micros(), t = tnow - lapseT;
    lapseT = tnow;
    return (float)t / 1000000.0;
}

float Helpful::sinceBegin()
{
    return float(micros() - T0) / 1000000.0;
}

bool Helpful::oneTime()
{
    if (!one)
    {
        one = 1;
        return true;
    }
    return false;
}

void Helpful::oneTimeReset()
{
    if (one) one = 0;
}

void Helpful::comparer(const float &n)
{
    maxi = (n > maxi) ? n : maxi;
    mini = (n < mini) ? n : mini;
}

float Helpful::getMax()
{
    return maxi;
}

float Helpful::getMin()
{
    return mini;
}

///Auxiliar para o uso do cartao SD
SDCH::SDCH(uint8_t csPin, String name, String type) :
CS(csPin), Fname0(name), Ftype((type.length() == 3) ? type : (String)"txt"), coef(8 -
name.length()), nMax(pow(10, (name.length() < 8) ? 8 - name.length() : 0))
{
    newName();
}

```

```

void SDCH::newName()
{
    Fname = Fname0;
    if (coef > 0)
    {
        if (String(number).length() > coef) number = 0;
        String n = String(number);
        Fname = Fname0;
        for (int i = coef - n.length(); i > 0; i--)    Fname += '0';
        Fname += n;
        number++;
    }
    else
    {
        String aux = Fname;
        Fname = "";
        for (int i = 0; i < 8; i++) Fname += aux[i];
    }
    Fname += '!' + Ftype;
}

bool SDCH::begin()
{
    if (!SD.begin(CS)) return false;
    else
    { //Inicializa o SD
        for (unsigned long int i = 0; i < nMax; i++)
        { //Cria um arquivo com nome novo
            if (!SD.exists(Fname)) break;
            else newName();
        }
        theFile = SD.open(Fname, FILE_WRITE);
        if (theFile)
        {
            theFile.close();
            return true;
        }
        else return false;
    }
}

bool SDCH::open()
{
    return theFile = SD.open(Fname, FILE_WRITE);
}

size_t SDCH::tab()
{
    return theFile.write('\t');
}

void SDCH::close()
{
    theFile.close();
}

```

```
SDCH::operator bool()
{
    return open();
}
String SDCH::getFname()
{
    return Fname;
}
```