

Criação do painel de instrumentos para motocicleta Honda modelo CRF230f

Dayan Morás ¹, Vitor Tumelero Valente ²

TCC2 - Curso de Engenharia de Controle e Automação

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul – Campus Farroupilha
Farroupilha, Brasil

¹ dayanmoras@gmail.com, ² vitor.valente@farroupilha.ifrs.edu.br

Resumo — Este artigo trata da criação de um painel de instrumentos para motocicletas Honda, especificamente para o modelo CRF de 230 cilindradas. O painel deve ser confiável para ser utilizado em condições adversas, economicamente viável, de fácil instalação visando tornar o equipamento um produto e que apresente diversas informações para o condutor como: velocidade, RPM (Rotação por minuto) do motor, hora, entre outros. Desenvolveu-se esse artigo dentro da disciplina de Trabalho de Conclusão de Curso II do curso de Engenharia de Controle e Automação do IFRS - Campus Farroupilha e é a continuidade de um projeto similar desenvolvido na disciplina de Projeto Integrador nesse mesmo curso e instituição. Para a estrutura mecânica, utiliza-se um invólucro em alumínio usinado em máquinas CNC (Comando numérico computadorizado) devido às características do processo e das propriedades físicas do material, como resistência a impactos. Para apresentar os dados ao condutor, dedica-se um display de LDC (*Liquid crystal display*) programável. Para o processamento dos dados, um Arduino Nano realiza todos os cálculos e se comunica via protocolo RS232 com o display enviando as informações. Um RTC (Real time clock), que utiliza comunicação I2C, auxilia os registros e cálculos dependentes do tempo. Obtêm-se a velocidade de forma indireta por meio de um sensor indutivo na roda dianteira da motocicleta. A rotação do motor é lida por meio do sinal de tensão do alternador previamente condicionado e posteriormente entregue a uma entrada analógica. Um protótipo do equipamento foi desenvolvido e características como *layout*, brilho e tamanho de tela, suportes, sensores, materiais utilizados e botões funcionaram conforme o esperado dentro das requisições do projeto. Algumas funcionalidades como a medição da velocidade e da distância percorrida ficaram comprometidas devido à leitura do sensor da roda ser afetada pelo tempo demandado de processamento, bem como a iteração do piloto com o sistema. O RTC e o sensor de temperatura funcionam conforme o esperado e de maneira confiável. O projeto, como um todo, cumpre com os requisitos e ideais definidos. Trabalhos futuros poderiam transformar o protótipo em um produto funcional sem grandes investimentos de tempo e dinheiro.

Palavras-chave — CRF; Painel de instrumentos; Display Nextion.

I. INTRODUÇÃO

Muitas pessoas buscam na prática das trilhas com motocicletas um momento de relaxamento longe do estresse das cidades e encontram nessas atividades um momento para estar com familiares, amigos e em contato com a natureza, além de benefícios para a saúde, assim como na prática de

outros esportes. Motocicletas, como a Honda modelo CRF230f, são produzidas nacionalmente e destinadas ao uso *off-road*, sendo ideais para essa prática muito comum no Brasil. Como são motocicletas que não podem ser emplacadas e utilizadas em estradas pavimentadas, não há exigências quanto à iluminação (lanterna traseira, luzes de freio, piscas, entre outros), nem instrumentação (velocímetro, hodômetro, entre outros). Esses fatores a tornam uma motocicleta mais acessível em termos de custo, muito difundida e de manutenção simples [1].

A partir da utilização do equipamento, percebe-se que alguns instrumentos se fazem necessários. Um exemplo é o horímetro (ou hodômetro) que é utilizado para se ter a informação de quando o óleo do motor deve ser trocado, bem como quando alguma manutenção preventiva deve ser realizada, como por exemplo a limpeza do filtro de ar ou a troca da vela de ignição, entre outros. Alguns usuários optam por não instalar e assim realizam trocas constantes e desnecessárias de óleo, gerando impactos ambientais. No caso do óleo, o descarte inadequado gera consequências como a contaminação de água e do solo devido à presença de compostos tóxicos como metais pesados em sua composição [2]. Em outros casos, acabam adiando a troca de óleo, causando, assim, problemas de desgaste prematuro em componentes do motor, culminando, nos casos mais graves, na necessidade da substituição/reforma de componentes do mesmo.

É possível encontrar no mercado equipamentos que auxiliem os proprietários dessas motocicletas com o problema apresentado. Entre eles estão: horímetros, velocímetros padrões ou mesmo aplicativos para celulares como o Strava [3], que são adaptados para uso nessas motocicletas. Entretanto, essas tecnologias não são dedicadas para esse equipamento e requerem, geralmente, mão de obra especializada para sua instalação.

Neste contexto, é proposto o desenvolvimento de um painel de instrumentos para motocicletas Honda, especificamente para o modelo CRF de 230 cilindradas, confiável para ser utilizado em condições adversas e de fácil instalação. Algumas informações devem ser apresentadas, tais como: velocidade, RPM do motor, hora, temperatura ambiente, tempo de motor ligado e distância percorrida. Algumas dessas informações visam reduzir o custo com

manutenções desnecessárias, trocas de óleo e possíveis impactos ambientais.

II. REVISÃO BIBLIOGRÁFICA

Existem alguns trabalhos já desenvolvidos que se assemelham com o proposto. Um exemplo disso é um sistema de telemetria desenvolvido na Universidade de Santa Maria [4]. Este projeto tem como objetivo desenvolver um sistema de telemetria para um *Kart Cross* utilizado em competições. O motivo para o desenvolvimento dessa tecnologia se origina na necessidade da análise de dados coletados, que é de vital importância para o aumento da performance da equipe em competições, assim como no desenvolvimento de novas tecnologias [4]. Neste projeto são coletadas, processadas e apresentadas informações como velocidade, RPM do motor, temperatura do óleo, entre outros, em tempo real para o piloto. Similarmente, esse projeto utiliza um Arduino Nano para processamento. O RPM é lido por meio do sinal enviado para a vela e a temperatura é lida por um sensor DS18B20. Os dados são também enviados para a equipe de apoio e apresentados em um aplicativo que pode estar até 1 km de distância [4]. Segundo o artigo, o sistema de telemetria se mostrou funcional e cumpriu de maneira satisfatória todas as tarefas requisitadas.

Há, ainda, um projeto que busca reduzir acidentes com motos através da implementação um sistema de aquisição de dados para uma moto de pista [5], mas que pode ser replicado para outras motocicletas. Esse sistema envia alarmes para o piloto, caso alguma falha seja encontrada. Foi criado um banco de dados com um histórico dessas informações, oferecendo, assim, mais segurança ao piloto. Diferentemente do sistema proposto por este artigo, a obtenção da velocidade é realizada por um sistema de GPS (Sistema de Posicionamento Global) e a leitura do RPM do motor é realizada por meio de um circuito baseado no circuito integrado LM2907, que converte um sinal pulsado proveniente de um sensor indutivo, interno no motor e que realiza a leitura da posição do virabrequim. Destaca-se que a leitura da velocidade apresentou um pequeno erro e possui um atraso de 1 s devido à conexão com o satélite. Esse erro é maior em baixas velocidades e diminui a medida que velocidade aumenta [5]. A leitura do RPM utiliza um sinal encontrado somente em motores com sistema de injeção eletrônica de combustível, diferentemente do modelo Honda CRF230f, no qual é utilizado um carburador para esse propósito.

Outro trabalho, desenvolvido para veículos de competição da Fórmula SAE, teve como objetivo desenhar e construir um painel de instrumentos para o veículo utilizado nessa competição pela Escola Superior Politécnica de Chimborazo do Equador [6]. Foi criado um painel de instrumentos, de acordo com o regulamento, para contribuir com o aumento da competitividade da equipe por fornecer informações vitais ao piloto em tempo real [6]. Destaca-se a utilização de um display da marca Nextion que permite a utilização de um programa dedicado, o Nextion Editor [7], para sua programação. Outro ponto importante salientado no trabalho foi a importância dada ao conforto do piloto, pois, foram utilizados caracteres com um tamanho relativo à distância do olho da tela, além da inclinação brilho e contraste ajustável [6].

Foi anteriormente desenvolvido pelo autor um projeto com os mesmos objetivos, na disciplina de Projeto Integrador do curso de Engenharia de Controle e Automação do IFRS - Campus Farroupilha. Destaca-se que a performance do equipamento ficou prejudicada, devido à utilização de um display que requisitava o processamento do microprocessador para seu funcionamento. Existe um comparativo entre o display Nextion e um display TFT ILI9341 [8] muito similar ao usado pelo autor anteriormente, que destaca a utilização de um código fonte menos extenso para sua implementação, requisitando menos memória e consequentemente menos poder de processamento.

III. FUNDAMENTAÇÃO TEÓRICA

Para melhor entendimento dos assuntos posteriormente tratados no item IV, apresentam-se definições, princípios físicos e/ou características dos principais componentes utilizados pelo projeto a seguir.

A. Sensor de efeito Hall

Trata-se de um sensor baseado no fenômeno físico descoberto Edwin Hall em 1889, o efeito Hall. Esse efeito relaciona a influência de um campo magnético, proveniente de um ímã, por exemplo, com a geração de uma diferença de potencial perpendicular ao fluxo de corrente elétrica [9], conforme a Figura 1 exemplifica.

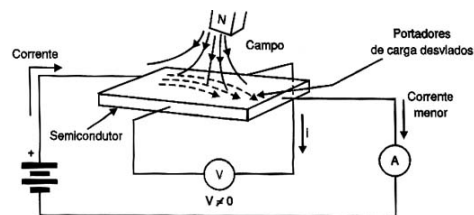


Fig. 1 – Circuito demonstrando o efeito Hall em um semicondutor [10].

Como a tensão que se estabelece é na ordem de milivolts, um circuito amplificador é usualmente utilizado para enviar esse sinal para outros dispositivos. Esse sinal pode ser na forma digital ou analógica, dependendo da aplicação.

B. Módulo RTC

RTC é uma sigla originada do termo em inglês *Real Time Clock* e trata-se de um componente capaz de informar hora, minuto, segundo, dia da semana, mês e ano para qualquer dispositivo externo via comunicação I2C [11]. Quando a alimentação é interrompida, uma bateria provê a energia necessária para sua operação, mantendo assim seus dados atualizados. Caso não houvesse a alimentação ininterrupta, a contagem de tempo seria interrompida. Circuitos integrados como o DS1307 são capazes de manter a informação atualizada por um período muito mais longo (Cerca de 80.000 vezes maior), se comparado com um microprocessador, por exemplo. O motivo para isso está no fato da potência consumida ser muito baixa, operando com uma corrente elétrica na ordem dos 500 nA [11], enquanto um microcontrolador como o ATmega328 opera com um consumo de 20 mA aproximadamente [12].

C. Display Nextion

Os *displays* são uma das formas mais completas existentes para interfacear o homem e máquina [6]. Apresentam saídas de dados de forma visual e permitem a entrada de dados via toque na tela. Os displays da linha Nextion apresentados pela Itead utilizam um *software* próprio para programação e para criação de telas [8], o Nextion Editor, apresentado na Figura 2.

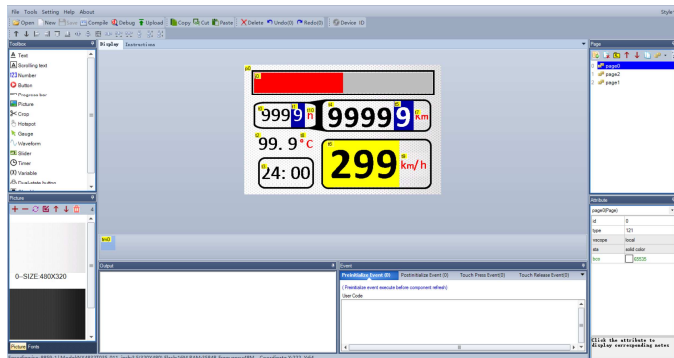


Fig. 2 – Ambiente de programação *Nextion Editor*.

A ferramenta busca facilitar a visualização, permite simulação e gravação, num mesmo ambiente de programação. A troca de informação com a unidade de processamento é via comunicação serial [6].

D. Comunicação via protocolo RS232

O protocolo RS232 é uma padronização para comunicação entre equipamentos, para troca de dados, largamente utilizada. Este padrão de normas, conhecido inclusive como EIA232, foi desenvolvido pela *Electronic Industries Association* nos anos 60 e especifica as tensões, temporizações e funções dos sinais [13].

É enviado um quadro de dados composto por: 1 *start bit*, 8 bits de dados e 1 *stop bit* [14]. A interpretação dos dados contidos nos quadros enviados e recebidos entre os dispositivos permitem a troca de informação. A principal limitação desse protocolo é que se permite a troca de dados entre 2 equipamentos somente através de, ao menos, 2 fios [14].

E. Comunicação via protocolo One Wire

Similarmente ao RS232, o protocolo *One Wire* trata-se de um conjunto de normas para troca de informação entre dispositivos eletrônicos. Esse protocolo baseia seu funcionamento em 3 fases. (1) A primeira fase habilita a comunicação e identifica os dispositivos conectados na rede [14], cada escravo possui um endereço fixo gravado no dispositivo durante a manufatura do componente. (2) A segunda fase seleciona o dispositivo com o qual a informação é trocada. (3) Por fim, na terceira fase ocorre a escrita ou leitura dos dados. Esse protocolo permite que até 2^{26} dispositivos sejam conectados simultaneamente no mesmo barramento de 1 fio [14].

F. Comunicação via protocolo I2C

Esse protocolo permite a comunicação com diversos dispositivos utilizando apenas dois fios. Entretanto, taxa de comunicação é até 4 vezes maior do que a *One Wire* [14].

O dispositivo principal deve informar aos outros dispositivos do início da comunicação, ou "*Start condition*". Quando isso ocorrer, os componentes estarão prontos para receber as informações seguintes que são: o endereço e a operação que este deve desempenhar. Após, quem tiver o endereço correspondente realiza a operação requisitada até haja um "*Stop condition*" para interromper a comunicação [14].

IV. PROPOSTA

Partindo do problema levantado, a seguir, são listados em categorias os principais pontos do desenvolvimento de uma solução viável. Inclui-se informações detalhadas sobre os componentes selecionados ou projetados.

A. Funcionalidades

Após definidos os objetivos do equipamento, foram identificadas as principais funcionalidades que o projeto deveria prover. Baseados nos sistemas de aquisição implementáveis sem a necessidade de ser invasivo, no sentido de modificar características da motocicleta, são listadas 6 funções propostas: apresentar o RPM do motor, o hodômetro, o hodômetro parcial, a temperatura ambiente, a hora e o velocímetro. Na Figura 3 é apresentada a disposição gráfica das informações no display. Buscou-se um grafismo moderno inspirado em motocicletas de outros segmentos e a máxima clareza possível na apresentação das informações para o piloto.

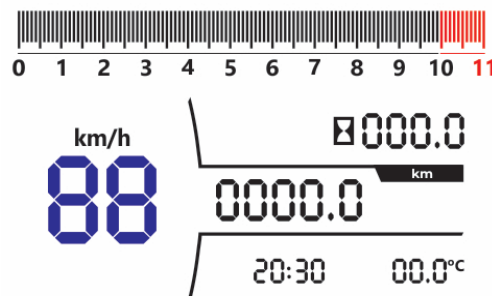


Fig. 3 – Disposição da informações no display.

Na Figura 4 é apresentada a tela de cronometragem. Trata-se de uma funcionalidade extra que busca agregar valor ao equipamento proposto. Nessa tela, o piloto pode cronometrar seu tempo por trajeto, visualizar o tempo das suas duas últimas cronometragens e seu melhor tempo. Inclui ainda um hodômetro que é zerado no início de cada medição. Através das teclas de função, o piloto informará o início e o término de cada volta.

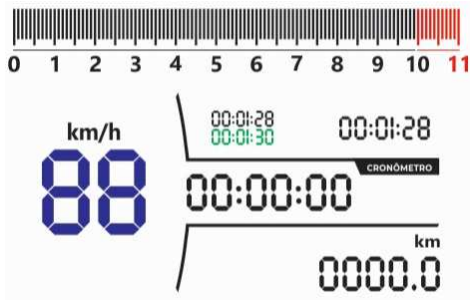


Fig. 4 – Disposição da informações no display.

B. Entradas do sistema

Para aquisição dos dados base para obtenção de forma direta ou indireta das grandezas necessárias na implementação das funcionalidades listadas, são necessárias 4 entradas, sendo elas: velocidade da roda dianteira, sinal do alternador (RPM do motor), botão F1 e botão F2. Essas entradas são conectadas à placa principal de processamento em sua maioria utilizando circuitos de condicionamento ou acoplamento de sinais.

Após definidas as entradas do sistema, foi feita uma pesquisa para se obter o maior número de componentes prontos para uso que atendessem os requisitos do projeto. Os componentes que não puderam ser encontrados foram confeccionados. Conforme mostra a Figura 5, foram utilizados um Arduino NANO e um display Nextion de 480x320 pixels e 3,5”, que se comunica via protocolo RS232 com o Arduino. Além desses componentes, foram adquiridos: Módulo RTC baseado no componente DS1307, sensor de temperatura DS18B20, sensor de efeito hall baseado no componente 33E938 e conversor rebaixador de tensão baseado no chip LM2596. O módulo RTC se comunica via protocolo I2C com o Arduino, já o sensor de temperatura escolhido utiliza protocolo OneWire para se comunicar com o Arduino. Foi necessário desenvolver uma placa eletrônica que oferecesse o suporte para as teclas F1 e F2, além de um divisor de tensão para aquisição do valor de velocidade do motor (Saída do alternador) em uma entrada. As teclas serão utilizadas para o ajuste da hora e para o reset do hodômetro parcial.

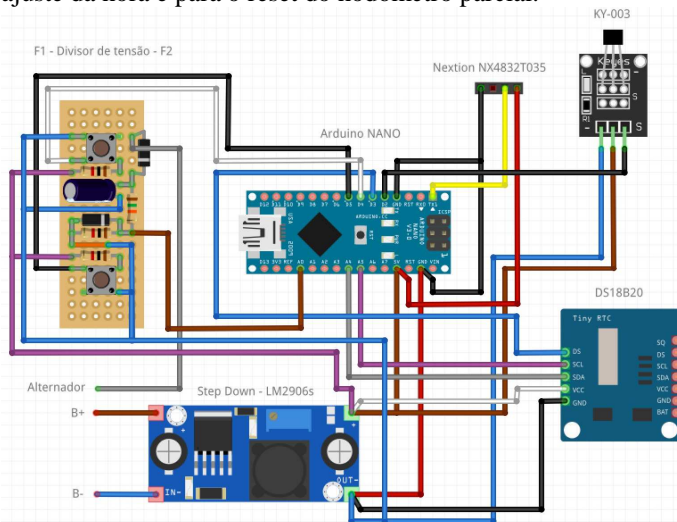


Fig. 5 – Componentes utilizados no projeto.

A Figura 6 apresenta o diagrama elétrico equivalente ao da Figura 4. Destaca-se o conector ‘J3’ presente no circuito, que é a derivação para o display Nextion onde: o pino 1 é conectado no GND do display, o 2 não é conectado, o 3 no RX do display e o 4 no +5V do display.

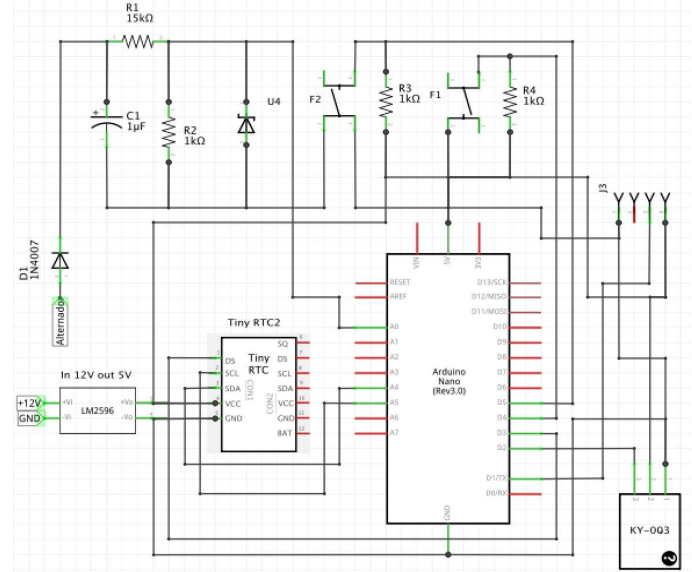


Fig. 6 – Circuito proposto para o projeto.

Ainda, baseando-se no circuito elétrico parcial da Figura 7, verifica-se que o sinal do Alternador pode ser conectado no fio ‘W’ do Regulador/Retificado da motocicleta, sinal GND pode ser conectado no ‘G’ do ICM (Módulo de controle da ignição) e o +12V pode ser conectado no ‘BI’ do interruptor de partida.

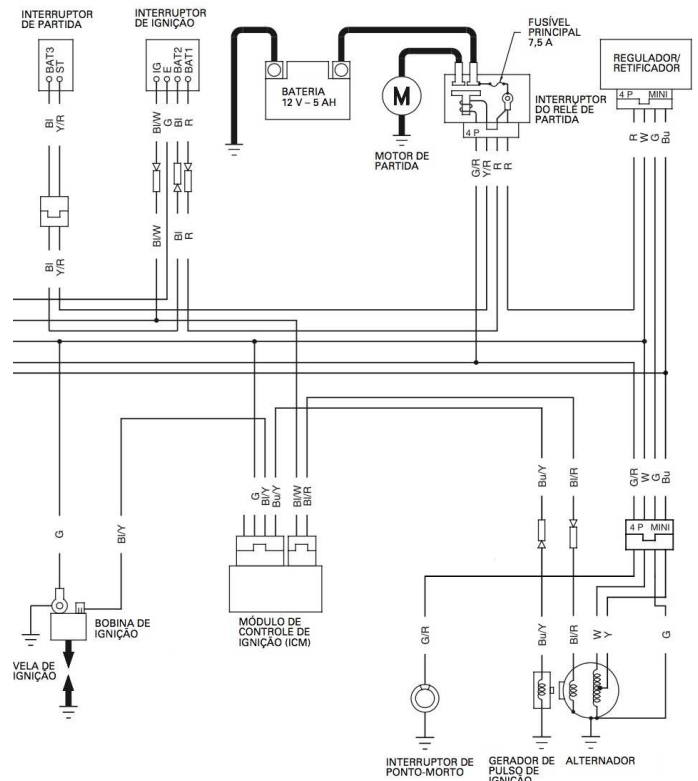


Fig. 7 – Diagrama elétrico original da motocicleta [15].

V. RESULTADOS

Foram desenvolvidos os respectivos suportes e invólucros descritos anteriormente. Nas figuras 18 – 20, são apresentados os resultados da instalação na motocicleta. Todos os sistemas elétricos foram testados e funcionam conforme o esperado.



Fig. 18 – Sistema proposto instalado (Imagem no display ilustrativa).



Fig. 20 – Vista ampla lateral do sistema proposto instalado.



Fig. 19 – Vista lateral do sistema proposto.



Fig. 21 – Detalhe do suporte do sensor de velocidade.



Fig. 22 – Detalhe do invólucro para o sensor de velocidade.



Fig. 23 – Imã inserido num parafuso M6 que suporta o disco de freio dianteiro.

A. Oportunidades para trabalhos futuros

O sistema de cronometragem, para os casos onde o piloto percorre um circuito ao invés de uma trilha, poderia vir a ser incômodo, devido a utilização de uma tecla para início e término do trajeto. Foi proposta a utilização de um sistema de GPS para automatizar esse processo e vir a oferecer mais comodidade ao piloto. Este módulo caso implementado, eliminaria a necessidade do RTC e poderia servir de auxílio para a leitura da velocidade, como descrito no relatório anterior.

Após a construção do protótipo real, foi optado por desenvolver futuramente um botão remoto próximo à mão esquerda do piloto, pois, os botões utilizados ficam posicionados demasiadamente longe das mãos do piloto. Os botões F1 e F2 previstos atualmente seriam eliminados no projeto futuro, diminuindo a complexidade do invólucro do equipamento e consequente os custos com matéria prima e usinagem, entretanto, para critérios de simulação, foi utilizado somente o botão F1 para os testes. A posterior migração desse

botão F1 visa inclusive eliminar a necessidade da utilização do módulo de GPS pois o botão ficará mais acessível, muito mais simples e confiável de operar se comparado as incertezas que o módulo de GPS poderia vir a oferecer (como a medição da velocidade apresentar um erro em baixas velocidades e agravar o erro de posicionamento conforme o número de satélites encontrados [5]).

Nesta previsão, percebeu-se que o programa deveria ser repensado para operar somente com uma tecla. O fluxo do programa se tornou complexo devido essa implementação, sendo assim, para descrever como as funcionalidades propostas foram concebidas, apresenta-se a seguir uma série de fluxogramas com explicações detalhadas a respeito de seu funcionamento. Acrescenta-se inclusive, como uma oportunidade de melhoria, a utilização de um processador mais poderoso para a aplicação. Conforme detalhado a seguir, juntamente com os fluxogramas, uma das maiores limitações encontradas durante o desenvolvimento da aplicação foi a capacidade de processamento limitada.

B. Funcionalidades

Antes de descrever o funcionamento detalhado de cada estrutura dedicada, precisa-se deixar claro algumas questões. A primeira é que devido a baixa capacidade de processamento do processador escolhido, que demandou algumas ações para contorno, sendo que, uma delas, foi a criação de uma tela de inicialização. Essa tela nada mais é (para o operador) do que o logo da marca Honda que aparece gradualmente e desaparece (como uma espécie de animação). Entretanto, durante esse processo, inicializa-se o display, requisita-se a hora ao RTC, lê-se a memória EEPROM para obtenção de informações pertinentes ao piloto posteriormente descritas nesse relatório e, por fim, a leitura da temperatura. Esse processo, principalmente devido à comunicação com o sensor de temperatura, demandava muito processamento e se tornava desconfortavelmente lenta a aparição das informações na tela principal na inicialização. Foi considerada a substituição do sensor por outro com diferente protocolo de comunicação, entretanto, entende-se que o desempenho atual passou a ser satisfatório após a utilização desse recurso. Esse recurso tornou desnecessária a intervenção no *hardware* que previa a utilização de um sensor de temperatura com comunicação I2C. Outras ações para melhoria de desempenho também serão descritas no tópico “performance” a seguir.

A segunda informação pertinente é que o fluxo de processamento torna-se um *loop* infinito, ou seja, algo se repete, após a inicialização. Essa questão torna muito mais complexa a maneira de se programar, necessitando de variáveis comumente presente no glossário da programação como *flags*. Essas memórias servem para sinalizar eventos sem interromper o fluxo do processamento, e foi amplamente utilizada nesse projeto. A aparição delas foi suprimida na maioria dos fluxogramas para facilitar o entendimento.

Sabendo disso, apresenta-se o primeiro fluxograma abaixo (Figura 24), que esclarece o processo de inicialização no que diz respeito às funcionalidades (não aos processos internos pertinentes ao funcionamento dos sistemas utilizados). Durante essa fase, caso o operador tenha pressionado a tecla F1 antes

mesmo de ligar o sistema, ocorrerá a reinicialização de todos os contadores como o hodômetro, o horímetro e todos os valores da tela de cronometragem. Caso o operador pressione a tecla F1 durante a exibição do logo Honda, será acessado o procedimento de ajuste de hora descrito na sequência. Ambos os processos são executados somente caso a última tela acessada tenha sido a principal. Isso se deve ao fato que a tela requisitada é mantida na memória EEPROM. Caso a tecla F1 não seja pressionada em algum momento, a inicialização conclui-se e a execução do programa se segue.

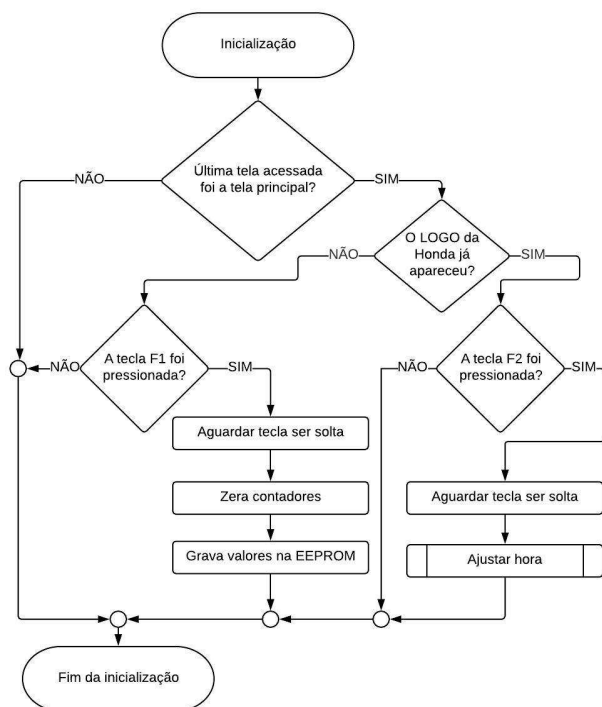


Figura 24 – Fluxograma do processo de inicialização.

Esses recusos implementados visam liberar total acesso ao equipamento por parte do operador, tendo em mente que o projeto se trata de um protótipo de produto.

O próximo fluxograma apresentado (Figura 25) representa o processo de ajuste da hora do equipamento. Basicamente, inicia-se o processo de ajuste pelo valor dos minutos, de forma que breves pulsos (menores que 1 segundo) incrementam o valor e um pulso mais prolongado (maior que 1 segundo) alterna para o ajuste da hora. Durante o ajuste da hora, da mesma forma, breves pulsos incrementam o valor e caso haja um pulso mais prolongado o processo é finalizado e o fluxo do programa se segue.

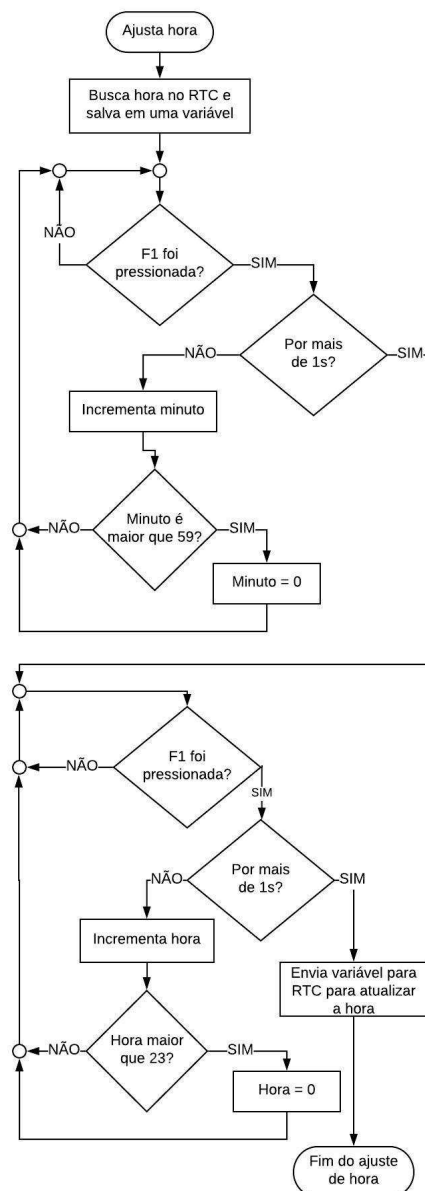


Figura 25 – Fluxograma do processo de ajuste da hora.

Na sequência (Figura 26), o programa entra no *loop* infinito, aqui diversos processos são requisitados de forma ordenada, sem interromper ou comprometer o processamento e o fluxo do programa. Cada um desses processos será explicado a seguir.

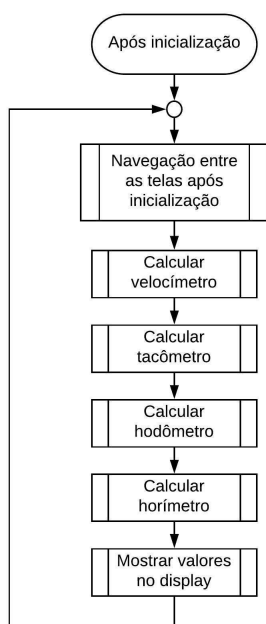


Figura 26 – Fluxograma do *loop* principal.

O fluxograma descrito a seguir (Figura 27) é responsável por exemplificar a maneira determinada para o operador interagir com o equipamento durante o uso. Trata-se da navegação entre as telas (principal e de cronometragem) e, de forma resumida, o controle da cronometragem como o início, o término e o reinício do cronômetro. Antes de descrever como se realiza esse processo, precisa-se esclarecer algumas definições adotadas no projeto. A primeira é que a cronometragem ocorre no formato “hh:mm:ss” ou seja, não há a cronometragem com precisão na ordem de submúltiplos do segundo. Essa definição ocorre devido a limitação da capacidade de processamento que impossibilita a comunicação com o display de forma suficientemente rápida. Entende-se que essa precisão é suficiente para a aplicação, pois, a maneira manual adotada para o controle do início e término já oferece imprecisões elevadas, bem como o critério de tempo não ter utilização em competições geralmente. Em segundo lugar, salienta-se a utilização da memória EEPROM no processo, que a cada nova iteração grava valores em endereços especificados e os restaura durante a reenergização do equipamento. Esse procedimento é extremamente importante pois permite que recuperar a última tela acessada, além dos valores das duas voltas anteriores e a melhor volta. Além disso, permite que a cronometragem não seja interrompida enquanto o equipamento estiver desligado. Dessa forma, se o motor “morrer” numa trilha (algo que geralmente acontece) a cronometragem não é interrompida e, ao religar o motor, a última tela visualizada já é acessada e os históricos são mantidos, permitindo assim mais comodidade na utilização do equipamento.

Agora, dando sequência à descrição do fluxograma, ao pressionar a tecla F1 durante um tempo de aproximadamente 2 segundos alterna-se entre a tela de cronometragem e a tela principal. Na tela principal não há outras funcionalidades para a tecla F1, porém, na tela de cronometragem existem diversas

funções acessadas por breves pressionadas simples (uma vez) ou dupla (duas vezes). Caso F1 seja uma vez brevemente pressionada (por um tempo menor que 200 ms) inicia-se o cronômetro. Entretanto, caso a cronometragem estiver em andamento, ocorre o incremento da volta (pelo procedimento “Nova Volta” posteriormente descrito). Caso a tecla F1 seja pressionada duas vezes a cronometragem é paralizada, porém, caso já esteja, a contagem de tempo, os históricos das voltas anteriores, a melhor volta e a distância percorrida são zerados.

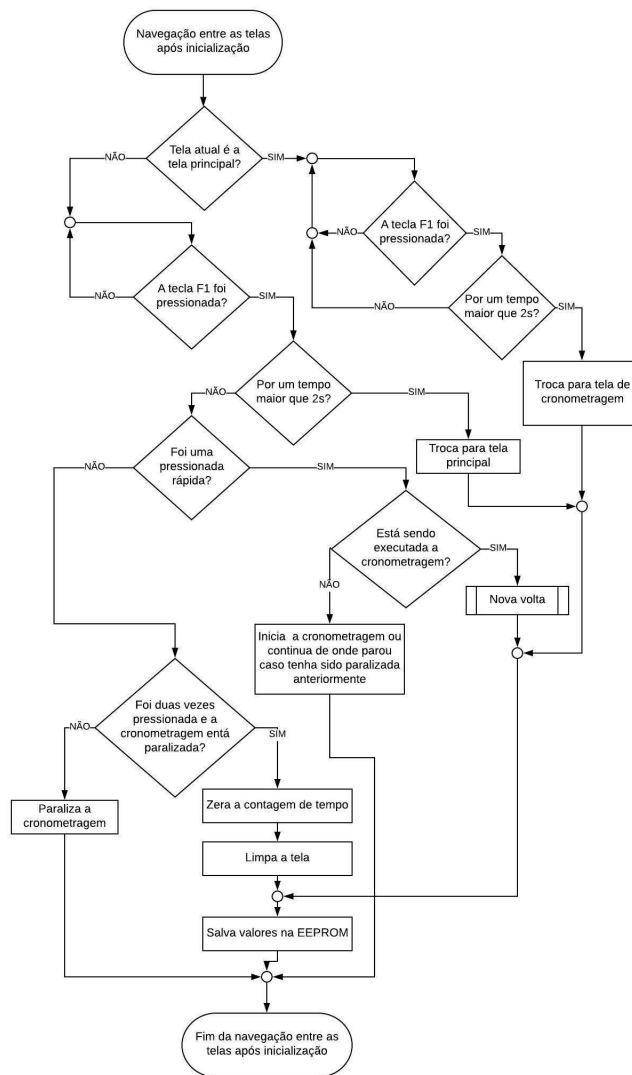


Figura 27– Fluxograma da navegação entre telas e controle da cronometragem.

Como parte do processo anterior, próximo fluxograma apresentado (Figura 28) trata do incremento da volta (Nova Volta) no processo de cronometragem. Nesse processo, os valores de tempo entre as voltas é transferido para a volta posterior. Além disso, o valor de tempo de cronometragem é convertido para segundos e comparado com o tempo da melhor volta. Caso a comparação resulte em um tempo menor para a última volta, o valor de melhor volta é atualizado. Para isso, o valor em segundos é novamente convertido para o formato “hh:mm:ss” e enviado para o display. Ocorre a mudança da cor do texto da volta em concordância com o teste de melhor volta,

ou seja, caso a volta tenha sido a melhor até o momento, a cor muda para verde, caso não, a cor muda para vermelho.

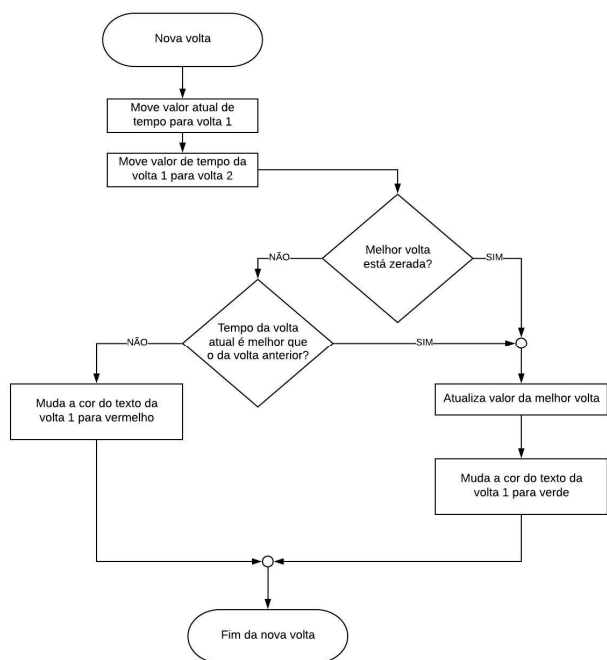


Figura 28 – Fluxograma de incremento de volta.

O fluxograma a seguir (Figura 29) apresenta a leitura do sensor da roda, o incremento da variável “voltas da roda”, o cálculo da velocidade e a apresentação da velocidade no display. A leitura do sensor utiliza uma *flag* para impedir que caso a roda pare acionando o sensor, o programa interprete de forma incorreta esse acionamento, incrementando a distância percorrida e apresentando velocidades de deslocamento absurdas. A cada acionamento do sensor, incrementa-se uma variável para que posteriormente seja calculada a velocidade e a distância percorrida. A velocidade é apresentada a cada 2 s no display, dessa forma, calcula-se a diferença entre o valor da variável “voltas da roda” atual e o valor da última vez apresentada a velocidade. Ao multiplicar esse valor pelo perímetro da roda, que é de 2,2 m, obtém-se a distância percorrida nesse intervalo de tempo de 2 s. Para converter para km/h basta multiplicar por 1,8 esse valor. Posteriormente, envia-se esse valor obtido para o display na forma de uma *string*.

Devido o perímetro da roda ser tão grande, o valor da velocidade limita-se a uma resolução de 4 km/h nesse caso. O tempo de amostragem no display, que geralmente é de 1 s para aplicações similares, passou a ser de 2 s para que a resolução não fosse de 8 km/h, aproximadamente. Para melhorar esse resultado optou-se por aumentar o número de leituras para cada volta, fracionando o perímetro de 2,2 m. Devido algumas limitações mecânicas, como o tipo de suporte desenvolvido e a disposição dos componentes de fixação existentes na motocicleta, foi possível obter 4 acionamentos por volta da roda. Isso fez com que o cálculo passa-se a ser: a diferença entre o valor da variável “voltas da roda” atual e o valor da

última vez apresentada a velocidade, multiplicado pelo perímetro da roda dividido por 4 (que resulta em 0,55 m) e por fim, multiplicado por 3,6 (ao invés de 1,8) pois o tempo entre as amostragens passou a ser de 1 s. Esse novo cálculo resultou em uma resolução de 2 km/h. Essa melhoria no número de leituras contribuiu muito para a melhor percepção da velocidade.

Optou-se por substituir o sensor de velocidade magnético de efeito *hall* por um sensor indutivo industrial. Esse sensor é capaz de detectar quando um metal se aproxima pelo menos 2 mm da face detectora, opera entre 8 e 24 Vcc e foi facilmente adaptado no sistema oferecendo uma medição mais confiável. O motivo principal para a substituição do sensor foi a necessidade de substituir todos os 4 parafusos que suportam o disco de freio. Suspeitava-se que esses parafusos, após serem perfurados para suportar o imã, pudessem ser fragilizados. Para que não ocorresse quaisquer problemas de segurança, utilizou-se os parafusos originais e substituiu-se o sensor. Além disso, o sensor industrial é mais confiável do que o sensor magnético, devido a detecção não ocorrer, dependendo do ângulo dos polos do imã (aperto do parafuso) em relação ao sensor magnético. Já o sensor indutivo, verifica a proximidade dos parafusos originais de maneira confiável e nenhum problema foi encontrado na utilização desse componente.

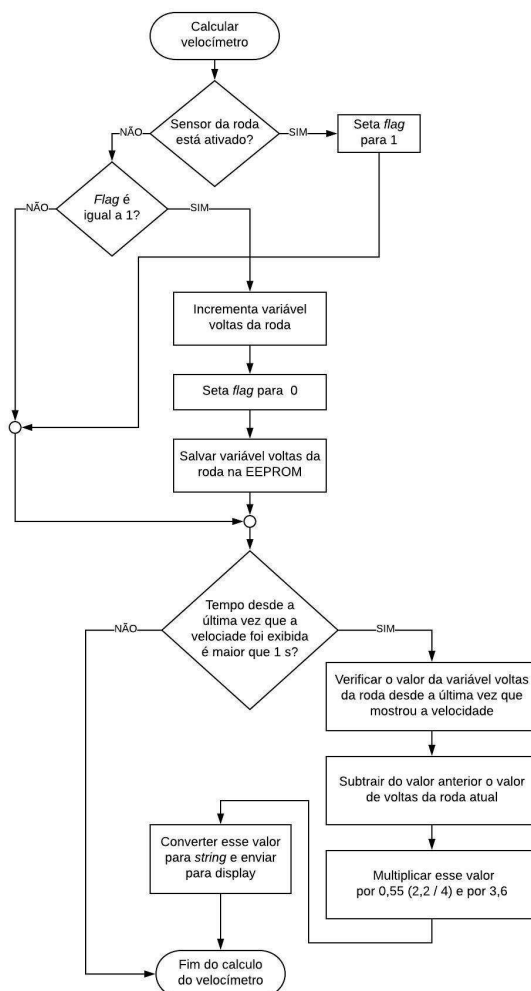


Figura 29– Fluxograma da leitura do sensor da roda e do cálculo da velocidade.

Quanto ao resultado da velocidade medida, percebeu-se um problema que não foi possível contornar durante o tempo delimitado para a execução desse trabalho. A capacidade limitada de processamento impossibilita a leitura de velocidades maiores do que 12 km/h, fato que foi verificado por um experimento: eliminou-se todas as tarefas executadas pelo programa que não fossem relacionadas com a leitura da velocidade e, utilizando um aplicativo de GPS que apresenta a velocidade de deslocamento, realizou-se o mesmo percurso onde, anteriormente, a velocidade apresentava erros de medição comparadas com o mesmo aplicativo. Com o programa “focado” em somente apresentar a velocidade, verificou-se que a velocidade corresponde com a do GPS, apresentando pequenas variações devido a resolução ser de 2 km/h enquanto o GPS possui resolução de 1 km/h. O ensaio foi limitado à 40 km/h devido o limite de velocidade da via e por questões de segurança. A solução para esse problema requer a substituição do processador.

A seguir no próximo fluxograma (Figura 30) depreve-se como é realizada a obtenção do valor do RPM do motor. A cada 300 ms lê-se a entrada analógica que recebe o valor de tensão do alternador, converte-se para um valor entre 0 e 100 e envia-se esse valor para o display.

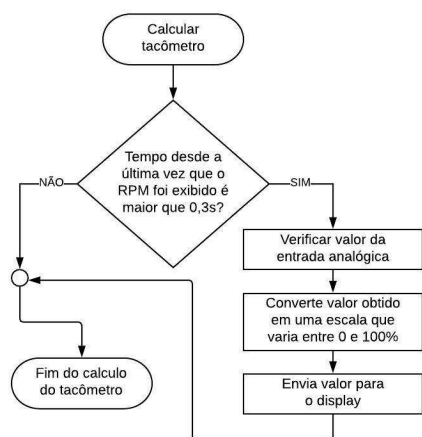


Figura 30 – Fluxograma da leitura do RPM do motor (tacômetro).

A leitura da entrada analógica é realizada 30 vezes e o valor médio dessas amostragens é utilizado para os cálculos de conversão. Esse procedimento visa “suavizar” a leitura, reduzindo os picos de tensão do alternador e melhorando a visualização. Esse valor médio obtido, necessita passar por um sistema de linearização para ser apresentado. Esse sistema compara o valor recebido e, dentro de 22 faixas preestabelecidas, converte o valor a ser entregue ao display. O sistema de linearização permite de maneira simples realizar essa aferição, basta verificar o RPM com um equipamento externo e para cada faixa ajustar o valor valor exibido. Entretanto, não foi possível aferir o tacômetro até o momento da elaboração desse relatório devido a falta de equipamento e aos prazos de entrega estabelecidos.

Apresenta-se a seguir o fluxograma (Figura 31) que trata do cálculo do hodômetro e o envio desse valor para o display. O

valor foi segmentado entre o valor inteiro e o valor em décimos de quilômetros. Para a tela principal, verifica-se o valor de “voltas da roda”, converte-se esse valor para a distância em km e em décimos de quilômetros e envia-se esse valor para o display. Caso esteja na tela de cronometragem (e esteja em andamento a cronometragem), o valor apresentado de distância apresentado é somente o percorrido no trajeto analisado. Sendo assim, o valor base da distância considera somente a diferença entre o valor da variável “voltas da roda” desde o início da cronometragem e o atual (toda vez que iniciada a cronometragem armazena-se o valor da variável para o cálculo). Após o valor é convertido da mesma forma descrita e enviado para o display. O processo ocorre a cada 1,5 segundos e salienta-se que o valor do hodômetro é constantemente gravado na memória EEPROM, não se perdendo com a desalimentação do sistema.

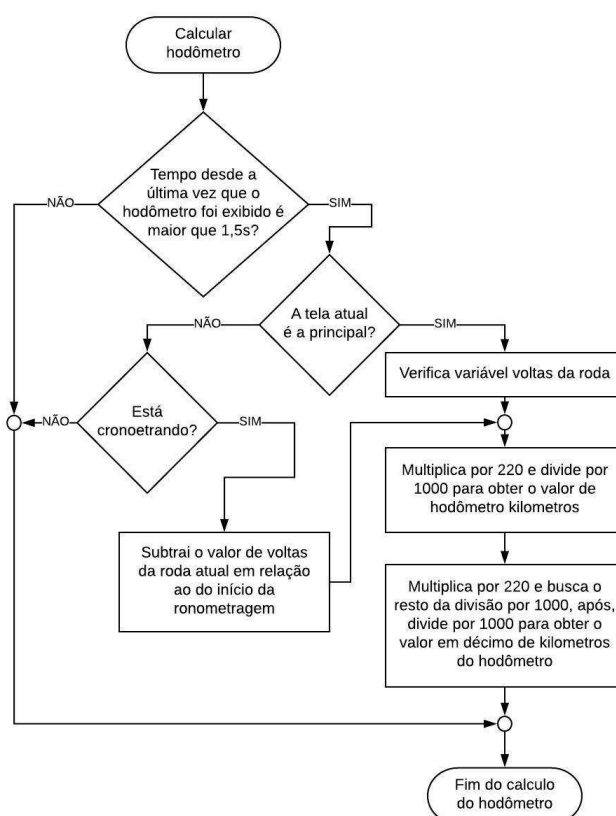


Figura 31– Calcular o hodômetro.

O processo descrito no fluxograma a seguir (Figura 32) depreve a maneira pela qual o horímetro proposto funciona. A cada 6 min, caso o motor estiver ligado, ou seja, o valor do RPM do motor for próximo de 1000, o valor de décimos de hora do horímetro é incrementado. Quando o valor for maior do que 9 significa que uma hora completa se deu, então, o valor de hora é incrementado e o valor de décimos é zerado. O processo se repete até que se atinja o valor limite de 999 horas, que é quando ocorre o processo de zeramento automático do valor, ou, caso desejado, pelo processo descrito durante a inicialização. Lembrando inclusive, que esse valor de horas não se perde com a desenergização devido a utilização da memória

EEPROM da mesma forma como que é feito para outras funcionalidades.

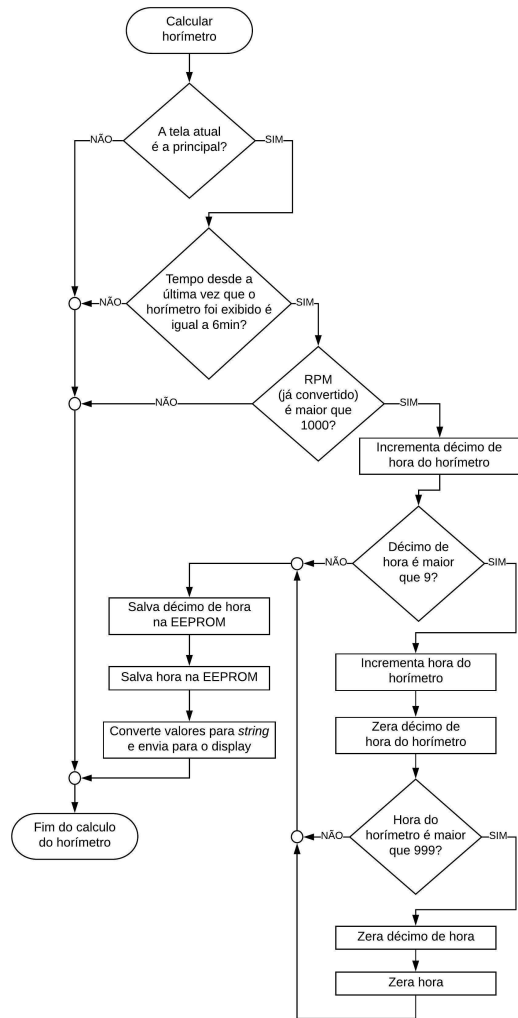


Figura 33 – Fluxograma do horímetro.

O último fluxograma apresentado (Figura 34) detalha como os demais valores são apresentados no display. A hora, a temperatura e o valor corrente (atual) da cronometragem são enviados de forma ordenada ao display em intervalos de tempo distintos devido a falta da necessidade da alta taxa de atualização desses valores, bem como para melhorar o desempenho do processamento. A hora e a temperatura estão somente presentes da tela principal, por isso, verifica-se a tela atual e o tempo desde a última atualização para que seja feita a requisição ao hardware responsável dos valores, a conversão necessária e o envio para o display. Na tela de cronometragem, caso executando a cronometragem, ocorre a execução dos cálculos e demais processos para a obtenção do tempo decorrido no cronômetro. Esse processo é executado a cada 1 s e basicamente verifica a diferença entre o tempo atual (lido do RTC) e o valor de tempo de quando foi iniciada a cronometragem (gravado em uma variável no início do cronometragem, assim como a distância descrita

anteriormente). O cálculo é realizado em segundos devido a maior facilidade na operação.

Ocorrem algumas peculiaridades nesse processo de obtenção do tempo decorrido no cronômetro. O primeiro deles é a verificação do tempo decorrido que, se for maior de 23 h 59 m e 59 s, realiza-se a paralização da cronometragem. O segundo, é o contorno de uma limitação do processo de subtração de tempo, que torna incoerente o valor caso ocorra a passagem pela meia-noite durante a cronometragem. Isso se deve ao fato de que o valor se torna negativo devido ao resultado da conversão para segundos do tempo atual ser menor do que o tempo desde o início da cronometragem. Para contorno, soma-se 86400 ao valor de tempo atual antes do cálculo da diferença de tempo. Esse valor e o equivalente em segundo contidos em um dia, e busca fornecer positivo o valor da diferença.

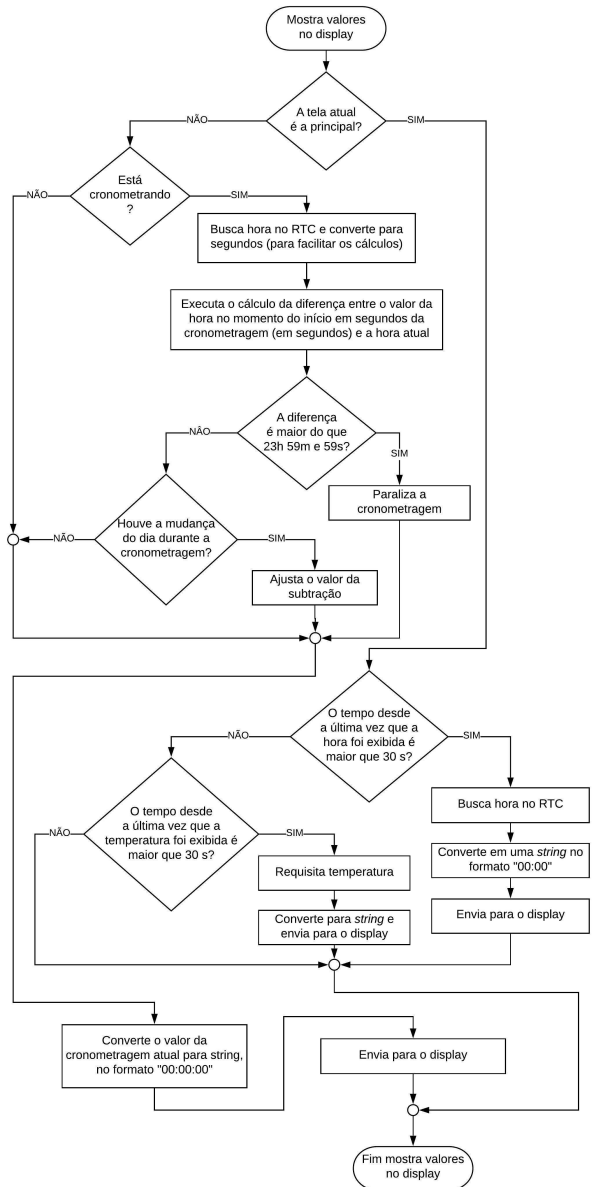


Figura 34– Fluxograma da apresentação dos valores no display.

C. Performance

Uma das principais dificuldade encontradas foi a necessidade do aumento da performance do programa. Algumas situações problemáticas já haviam sido relatadas anteriormente, bem como suas soluções, entretanto, as que não foram, serão relatadas a seguir. A primeira delas foi a sequência de instruções de teste de condições (em linguagem “C” conhecidos como *if*) utilizadas para verificar se o tempo entre cada amostragem já foi atingido. Havia sido utilizado inicialmente testes em sequência que prejudicavam a velocidade de processamento. Essas instruções foram complementadas com um artifício conhecido como *ifs* encadeados. Esse processo consiste em elencar cada novo *if* dentro de um *else* do anterior, assim, caso uma condição seja válida, os outros testes não são executados até um novo ciclo de rastreamento do programa, ou *scan*.

Outro problema encontrado foi que, cada vez que uma tela fosse selecionada, o programa aguardava o tempo determinado para apresentar os valores das variáveis no display. Esse problema foi contornado com a utilização de *flags* setadas no momento da troca de telas, que sinalizam para os comandos *ifs* que, mesmo que o tempo não tenha sido atingido, o valor deve ser enviado, forçando assim a atualização na transição de telas. Após uma vez enviado, a *flag* é resetada e o comando volta a verificar somente o tempo entre cada apresetação definido.

Ao iniciar a cronometragem, os valores de cada volta anterior, bem como o valor da melhor volta, são atualizados no display. Esse processo requisita muito processamento e tempo, tornando o início da cronometragem lento (cerca de 3 s atrasado). A maneira encontrada para amenizar esse problema foi realizar um teste com cada variável enviada, de forma que valores zerados não fossem enviados ao display. Embora essa tarefa seja uma sequência de 9 *ifs* em série (1 para cada ‘hh’, ‘mm’ e ‘ss’ de cada um dos 3 tempos), impossível de serem encadeados, o início da cronometragem inicia com somente 1 s de atraso. Essa redução no tempo é considerada satisfatória para o projeto.

D. Desempenho

Esse tópico visa descrever detalhadamente os resultados obtidos na medição de velocidade e leitura do RPM de forma prática, assim como o restante das funcionalidades. Apresenta problemas, precisão e demais informações para cada grandeza apresentada.

Iniciando pela tela principal, temos como resultado: a temperatura, a hora e o horímetro funcionando conforme o esperado, apresentando valores confiáveis e de maneira adequada. A velocidade (comum para a tela principal e para a tela de cronometragem), como descrito anteriormente, apresenta uma resolução de 2 km/h mas, apresenta valores distorcidos para velocidades acima dos 12 km/h, devido à performance do processador. O valor se torna mais acertivo quando a “carga” de processamento é reduzida. Essa distorção na leitura da velocidade interfere na medição da distância percorrida, que é menor do que o esperado. A resolução desse problema é possível somente com a substituição do processador. A escolha desse processador se deu ao fato do mesmo estar presente na revisão bibliográfica, e ser utilizado

no processamento de informações similares. Entretanto, não foi possível prever que todas as funcionalidades propostas demandariam tamanha capacidade de processamento.

Na tela de cronometragem, o sistema de cronometragem funciona como o esperado, entretanto, apresenta lentidão na iteração com a tecla F1. O comando para início e o término da cronometragem precisa, em alguns casos, ser repetido para que funcione, dependendo da velocidade do acionamento da tecla. Isso é reflexo do alto tempo de *scan* do programa cuja solução é possível somente com a substituição do processador. A leitura do RPM, comum para a tela de cronometragem e principal, apresentou um comportamento não-linear possível de ser contornado. Foi possível desenvolver um sistema de aferição, que não pode ser testado completamente durante o tempo limite para a entrega do projeto.

VI. CONCLUSÃO

Características como *layout*, brilho e tamanho de tela, suportes, sensores, materiais utilizados e botões funcionaram conforme o esperado dentro das requisições do projeto. Destaca-se no uso do display, sua programação e comunicação que são simples e intuitivas.

O projeto apresentou diversos desafios durante seu desenvolvimento, sendo o principal deles a capacidade de processamento do processador. Algumas funcionalidades como a medição da velocidade e da distância percorrida ficaram comprometidas devido à leitura do sensor da roda ser afetada pelo tempo demandado de processamento, bem como a iteração do piloto com o sistema de cronometragem. O RTC e o sensor de temperatura funcionam conforme o esperado e de maneira confiável de forma que nenhum problema foi constatado durante os testes. Esses problemas poderiam ser amenizados com a utilização de conversores de frequência em tensão como o CI 2907, entretanto, até o prazo de entrega estabelecido não foi possível desenvolver essa solução.

O projeto, como um todo, cumpre com os requisitos e ideais definidos. Trabalhos futuros poderiam transformar o protótipo em um produto funcional, sem grandes investimentos de tempo e dinheiro, devido os resultados estarem próximos dos objetivos propostos.

AGRADECIMENTOS

Aos professores, por seus ensinamentos e conselhos durante toda a graduação. Ao professor Me. Vítor em especial, orientador desse trabalho. Aos meus amigos e colegas de trabalho por todo o apoio prestado, pelo empréstimo da motocicleta utilizada, pelo auxílio prestado na elaboração e confecção dos protótipos e pelos ótimos conselhos.

REFERÊNCIAS

- [1] Honda CRF230F. Disponível em: <https://www.honda.com.br/motos/crf-230f>. Acesso em: 20 de março de 2018.
- [2] L. B. Salam *et al.*, “Metagenomic insights into effects of spent engine oil perturbation on the microbial community composition and function in a tropical agricultural soil” *Environ Sci Pollut Res*, vol. 24, no. 8, pp. 7139–7159, Jan. 2017.

- [3] Strava. Disponível em: <https://www.strava.com>. Acesso em: 20 de maio de 2018.
- [4] C. A. Machado, J. E. Baggio. Sistema de telemetria para veículos automotores. XXVIII Congresso de iniciação científica e tecnológica em engenharia. Disponível em: <https://www.publicacoeseventos.unijui.edu.br/index.php/cricte/article/download/8865/7572>.
- [5] Q. Tenesaca *et al.*, “Implementación De Un Sistema De Adquisición De Datos Para Monitoreo De Motos De Velocidad.”. Tesis de Pregrado, Universidad del Azuay, Cuenca, Equador, 2015.
- [6] J. Montesdeoca, *et al.*, “Diseño y construcción de un tablero electrónico-digital baseado en la normativa vigente de la fórmula SAE para contribuir con la construcción de un vehículo de competencia para la ESPOCH”. Tesis Ingeniero Automotriz, Escuela Superior Politécnica de Chimborazo, Facultad de Mecánica, Riobamba, Equador, 2016.
- [7] Nextion Editor. Disponível em: <https://nextion.itead.cc/resources/download/nextion-editor/>. Acesso em: 20 de maio de 2018.
- [8] A. C. Bento, “IoT of Nextion X TFT ILI9341: Experimental Results and Comparative Survey,” *International Research Journal of Engineering, IT & Scientific Research*, vol. 4, no. 4, pp. 14-23, Mar. 2018.
- [9] P. A. Tipler, G. Mosca, “Física para cientistas e engenheiros - Eletricidade e magnetismo, Ótica”, 6 ed, vol. 2, Brasil: LCT, 2009.
- [10] Sensor de efeito Hall. Disponível em: https://pt.wikipedia.org/wiki/Sensor_de_efeito_Hall. Acesso em: 20 de março de 2018.
- [11] DS1307 datasheet. Disponível em: http://www.alldatasheet.com/view.jsp?Searchword=Ds1307%20datasheet&gclid=Cj0KCQjw5qrXBRC3ARIsAJq3bwqKkLtMwODiXEt2LJ_N9fiVmofIJmgSxyen9McOPN0eoiewlgxQ-czgaAkxIEALw_wcB. Acesso em: 20 de março de 2018.
- [12] Arduino nano. Disponível em: <https://store.arduino.cc/usa/arduino-nano>. Acesso em 24 de fevereiro de 2018.
- [13] J. P. Marques, *et al.*, “Implementação de uma arquitetura de comunicação serial RS-232 com aplicação em uma unidade aritmética em ponto flutuante de 32 bits,” *não publicado*.
- [14] Comparação Entre Protocolos de Comunicação Serial. Disponível em: <https://www.robocore.net/tutoriais/comparacao-entre-protocolos-de-comunicacao-serial.html>. Acesso em: 20 de janeiro de 2018.
- [15] Honda Motorcycle Service Repair Workshop Manuals. Manual de Servicos Honda CRF230F. Disponível em: <https://bit.ly/2rjQhZb>. Acesso em 15 de janeiro de 2018.