UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Documentação de Projeto

CONOPS, Domínio do Problema, Especificação e Design da Solução

---

Esta documentação vale-se dos modelos elaborado pelo Prof. Douglas Renaux para o uso no Projeto Final da disciplina de Sistemas Embarcados, com adaptações realizadas pelo estudante.

Projeto: Veículo Autoguiado

Autor: João Pedro Zanlorensi Cardoso

Versão: 28-ago-21

Parte 1 – CONOPS

1 Introdução

O sistema desenvolvido consiste em um sistema embarcado para controle de um veículo autoguiado, construído com propósito didático.

O sistema embarcado deverá ser conectado a um computador e irá interagir diretamente com um simulador de corrida, recebendo e enviando informações para ele, e indiretamente com o usuário do computador, que enviará instruções ao simulador e ao veículo.

O simulador possui uma interface gráfica que o permite exibir o veículo em movimento, bem como permite ao usuário do computador definir os parâmetros de operação do veículo, como as definições de como deverá ser a pista de corrida.

As informações transmitidas pelo o simulador consistem na leitura dos sensores do veículo simulado e em alertas de uso- tais como um sinal ao iniciar a corrida simulada ou uma notificação caso o veículo esteja próximo de uma colisão.

O usuário do computador poderá, ainda, enviar ao simulador uma instrução para reiniciar a corrida.

O objetivo do veículo, como sendo autoguiado, é o de não colidir com os obstáculos na pista e de terminar uma volta completa na pista no menor tempo possível.

2 Descrição do Sistema

O projeto utiliza um Kit EK-TM4C1294XL, da Texas Instruments, conectado a um computador através da interface serial UART, para realizar o controle de um veículo simulado por meio do simulador SimSE2 – Veículo Autoguiado.

O simulador tem o propósito de mostrar visualmente a operação do veículo, bem como de enviar dados referentes à sua operação através de uma porta serial para o Sistema Embarcado, e também permite a um usuário a definir os parâmetros de operação do veículo e enviar uma instrução de reset, que reinicia uma corrida.

O sistema embarcado é capaz de processar as informações enviadas através do simulador para a interface serial e de enviar comandos aos acionadores do veículo no simulador.

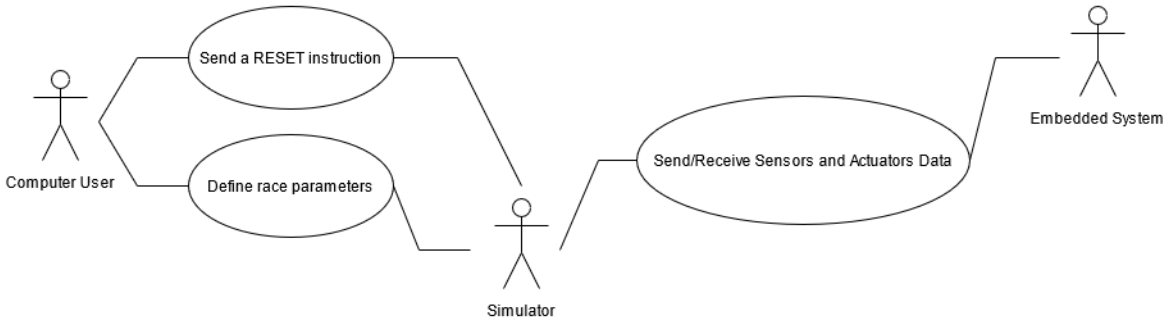

Imagem 1 – Diagrama de casos de uso

3 Interface com o Usuário

O usuário poderá definir os parâmetros de corrida através da interface gráfica do simulador SimSE2 e também poderá enviar uma instrução de reset ao simulador através de uma porta de comunicação serial utilizando o protocolo UART.

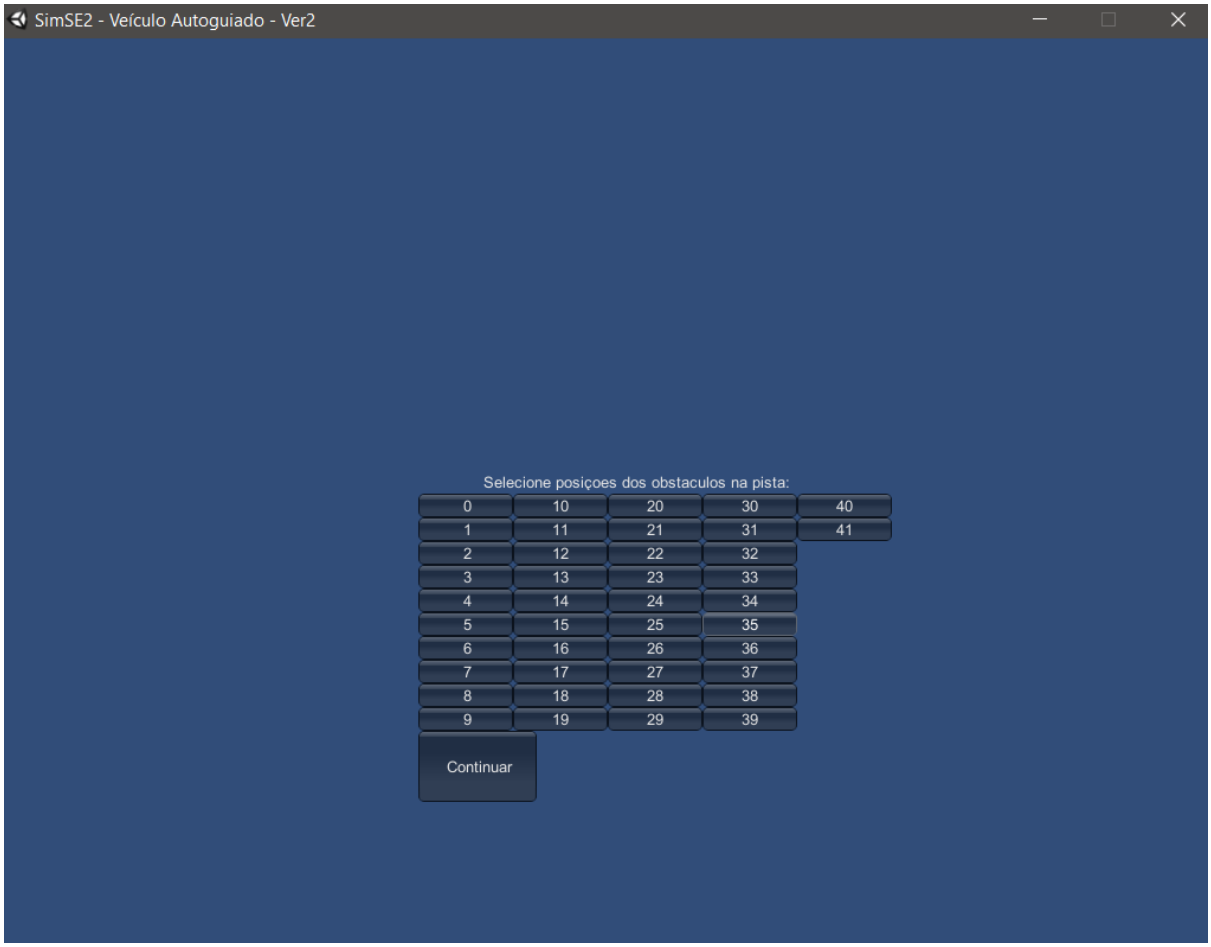

Imagem 2 – Interface do simulador ao usuário

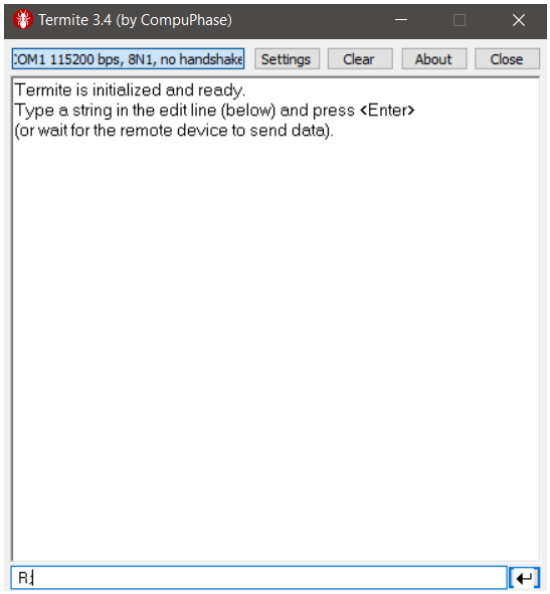

Imagem 3 – Comunicação Serial

4 Identificação dos Stakeholders

Os envolvidos no desenvolvimento e utilização deste sistema são: o estudante matriculado na disciplina de Sistema Embarcados da UTFPR, responsável pelo desenvolvimento do projeto; a banca avaliadora, composta pelos professores da disciplina de Sistemas Embarcados da UTFPR, que farão avaliação do conjunto concebido.

5 Requisitos de Stakeholders

O estudante tem como objetivo desenvolver o projeto empregando de forma acurada o máximo de assuntos dentre os adquiridos durante a disciplina de Sistemas Embarcados da UTFPR.

A banca avaliadora tem a função de avaliar a atividade desenvolvida, de forma a garantir a sua qualidade.

6 Cenários de Operação

A operação do sistema se dará na forma de um teste, realizado da seguinte forma:

- O estudante realizará a simulação do sistema enunciado, comentando os principais aspectos de implementação e os seus desafios de desenvolvimento, e fornecerá ao professor informações acerca dos resultados da simulação e os materiais de apoio ao projeto (como a documentação e a especificação), bem como o código desenvolvido para o Sistema Embarcado;
- O professor avaliará os materiais fornecidos pelo estudante.

Parte 2 – Domínio do Problema

1 Sistemas Computacionais Embarcados

Sistemas Computacionais Embarcados são sistemas computacionais- com capacidade de processamento, memória e periféricos- que integram outro(s) sistema(s), reagindo a estímulos internos e externos a ele.

Sistemas Computacionais Embarcados possuem cenários de uso específicos; raramente exigem modificações de software embarcado (firmware); e atendem a diversas restrições, tais como custo, tamanho, robustez, restrições de consumo e de operação.

2 Operação em Tempo Real

Uma operação em Tempo Real é definida como uma operação que é executada sob um regime de período pré estabelecido.

Dentre as operações em Tempo Real, destacam-se as:

- **Operações em hard real-time:** consistem em operações em cujo a falta de rigor ao respeitar o período pré-estabelecido de execução pode ocasionar em situações críticas ou fatais ao sistema ou às partes que o compõe, bem como aos seus usuários;
- **Operações em soft real-time:** consistem em operações em cujo a falta de rigor ao respeitar o período pré-estabelecido de execução acarreta degradação do desempenho do sistema ao ponto de vista de seu(s) usuário(s).

3 Programação Concorrente

A programação concorrente está fundamentada no princípio de criar múltiplas linhas de atividades (tarefas) que têm a possibilidade de serem executadas em paralelo.

A ordem de execução destas tarefas é determinada por políticas de escalonamento, e em sistemas mononúcleo não há a possibilidade de execução simultânea de duas ou mais tarefas. Ao mesmo tempo, há o desafio de estabelecer políticas de comunicação entre as tarefas eficientes, como com a utilização de filas de mensagens, que permitem o envio e o recebimento de tarefas entre duas ou mais tarefas através de um canal do tipo fila de mensagens.

Parte 3 – Especificação e Design da Solução

1 Especificação Funcional

Os itens abaixo consistem nos Requisitos Funcionais (RFs) do projeto:

- **RF1:** O veículo deverá realizar a leitura periódica dos dados coletados pelos sensores físicos que o integram
- **RF2:** O veículo deverá percorrer uma pista de corrida de forma autônoma
- **RF3:** O veículo não deverá colidir com os obstáculos da pista ou com as margens da pista que percorrer
- **RF4:** O veículo deverá acelerar para frente automaticamente ao início da corrida
- **RF5:** veículo deverá aumentar a sua aceleração se não detectar obstáculos a uma distância limite à sua frente, que deverá ser calculada dinamicamente de forma a garantir que ele não colida
- **RF6:** O veículo deverá desviar automaticamente se houver obstáculos à uma distância limite da sua frente, que deverá ser calculada dinamicamente conforme a velocidade em que o veículo estiver
    - **RF6.1:** Para desviar, o veículo deverá freiar, virar em um ângulo adequado para que não haja a sua colisão e acelerar na direção a que ele virou
    - **RF6.2:** O veículo deverá ser capaz de determinar o ângulo adequado para virar de forma que não haja colisão em nenhum ponto da sua superfície com os obstáculos
    - **RF6.3:** Se um obstáculo estiver a 5 metros da sua frente, o veículo deverá realizar uma parada emergencial, seguida de uma virada em um ângulo que o permita não colidir, e de uma aceleração
- **RF7:** Ao final de uma volta pela pista, o veículo deverá freiar

2 Especificação não-Funcional

Os itens abaixo consistem nos Requisitos Não Funcionais (RNFs) do projeto:

- **RNF1:** O período de leitura dos dados coletados pelos sensores do veículo deverá ser de 200 milisegundos
- **RNF2:** O veículo deverá piscar um LED quando o simulador receber uma instrução de reset
- **RNF3:** O veículo deverá ser capaz de coletar estatísticas da sua operação e de transmiti-las a um computador ao final de cada volta na pista
    - **RNF3.1:** O veículo deverá transmitir qual foi a sua máxima velocidade atingida
    - **RNF3.2:** O veículo deverá transmitir qual foi a quantidade de obstáculos de que ele desviou
    - **RNF3.3:** O veículo deverá transmitir qual foi o tempo total que ele levou para completar a volta

- **RNF3.4:** O veículo deverá transmitir qual foi a sua velocidade média ao longo do trajeto
- **RNF3.5:** As estatísticas de operação deverão ser transmitidas ao usuário externo através da interface serial UART na forma de texto

- **RNF4:** O veículo deverá evitar cálculos redundantes

3 Restrições

Os itens abaixo consistem nas Restrições (Rs) do projeto:

- **R1:** O sistema eletrônico do veículo deverá utilizar funções do CMSIS-RTOS
- **R2:** O protocolo de comunicação entre o veículo e o usuário externo capaz de intermediar a corrida e visualizar suas estatísticas deverá ser o protocolo UART
- **R3:** O veículo deverá utilizar o Kit EK-TM4C1294XL

4 Arquitetura Estática do Sistema

A figura abaixo consiste no diagrama Arquitetura Estática para o sistema concebido. Ele almeja descrever os elementos programáticos que compõe o sistema e a forma como eles interagem entre si:

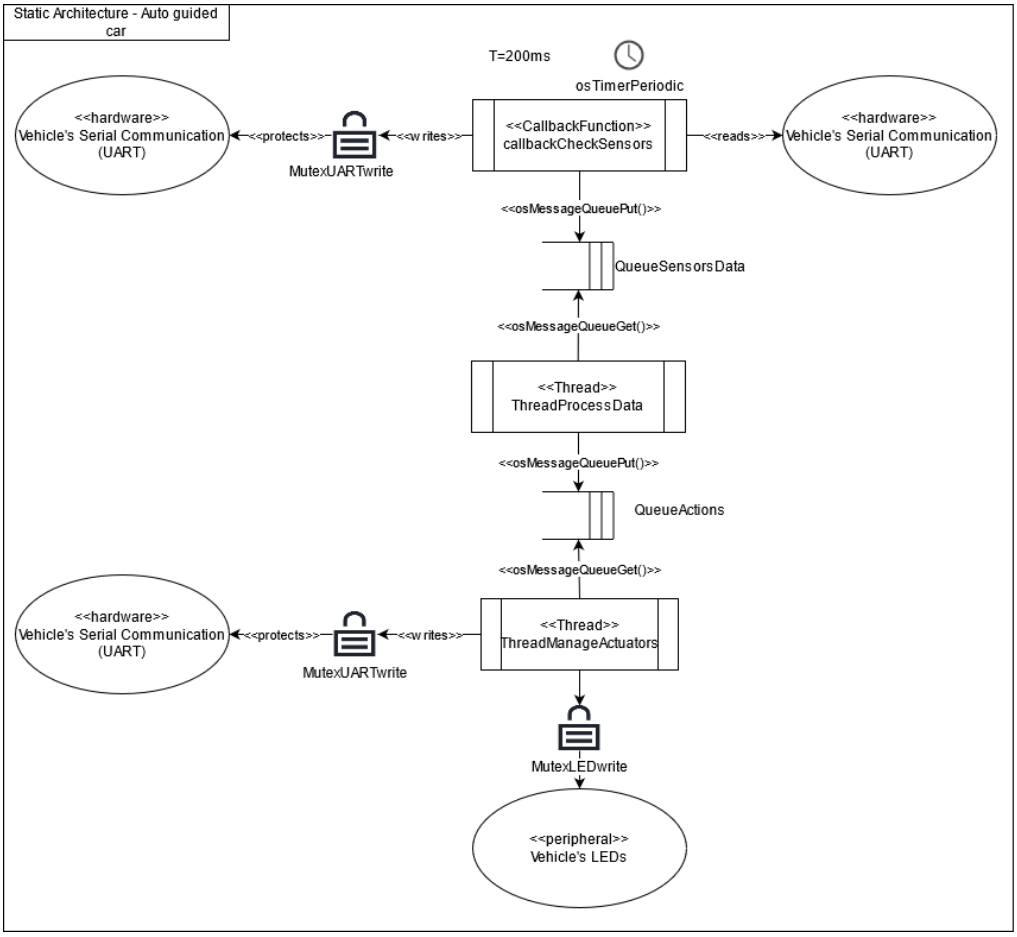

Imagem 4 – Arquitetura Estática do Sistema

5 Comportamento Dinâmico do Sistema

A figura abaixo consiste no diagrama de Máquina de Estados para o sistema concebido. Ele almeja descrever o comportamento dinâmica do sistema:

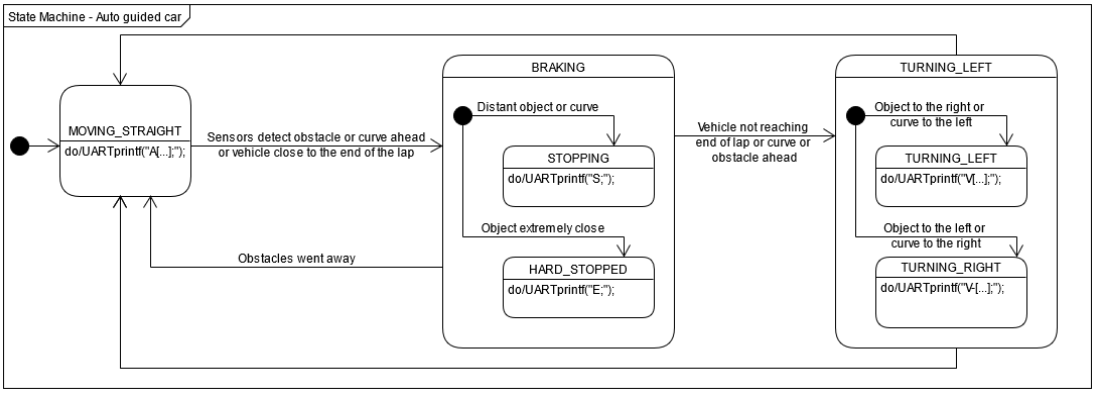

Imagem 5 – Diagrama de Máquina de Estados

Parte 4 – Estudo da Plataforma

1 Comportamento das tarefas

Os diagramas abaixo descrevem o comportamento das tarefas do sistema através de fluxogramas.

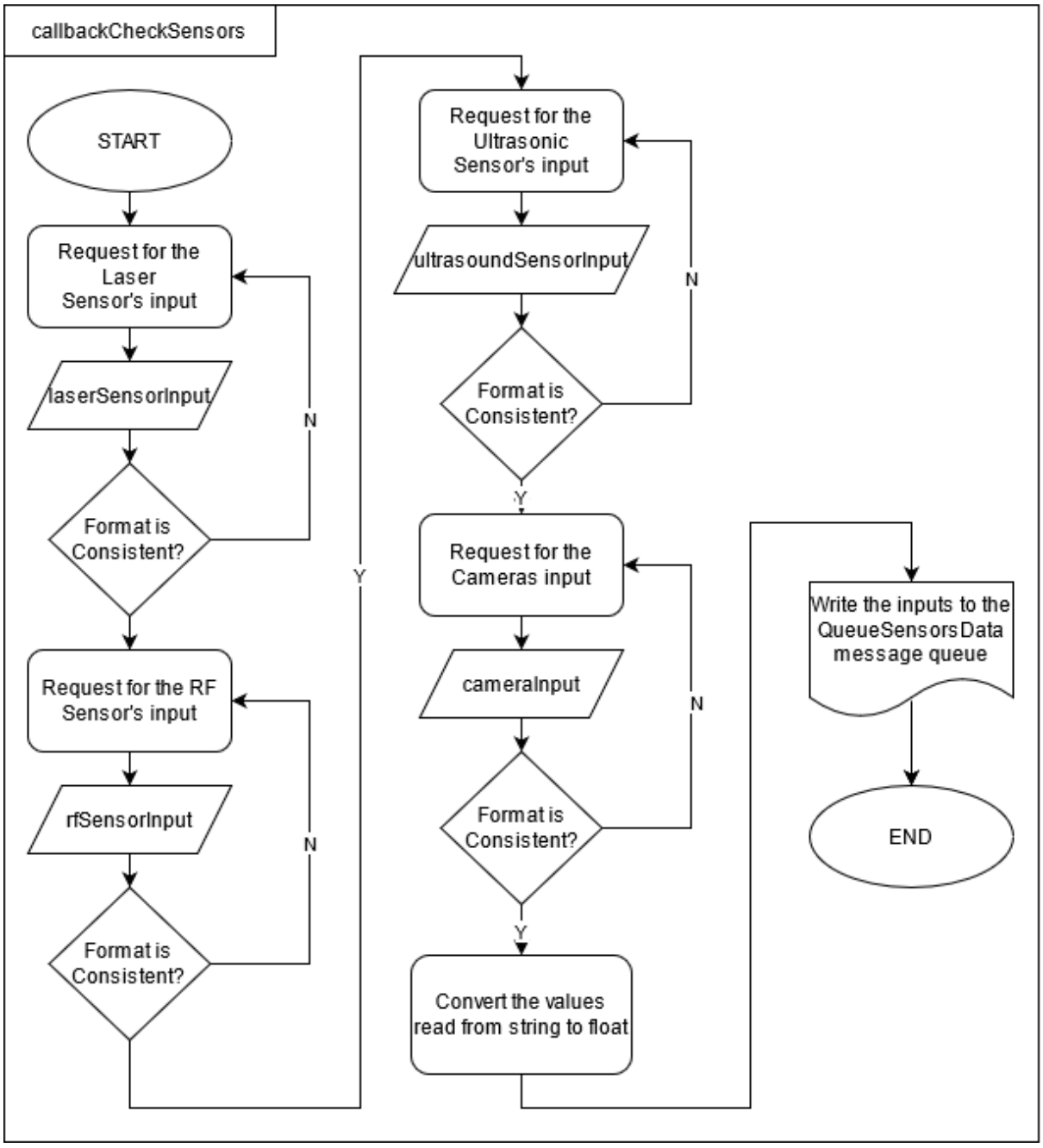

Imagem 6 – Fluxograma da função de callback callbackCheckSensors

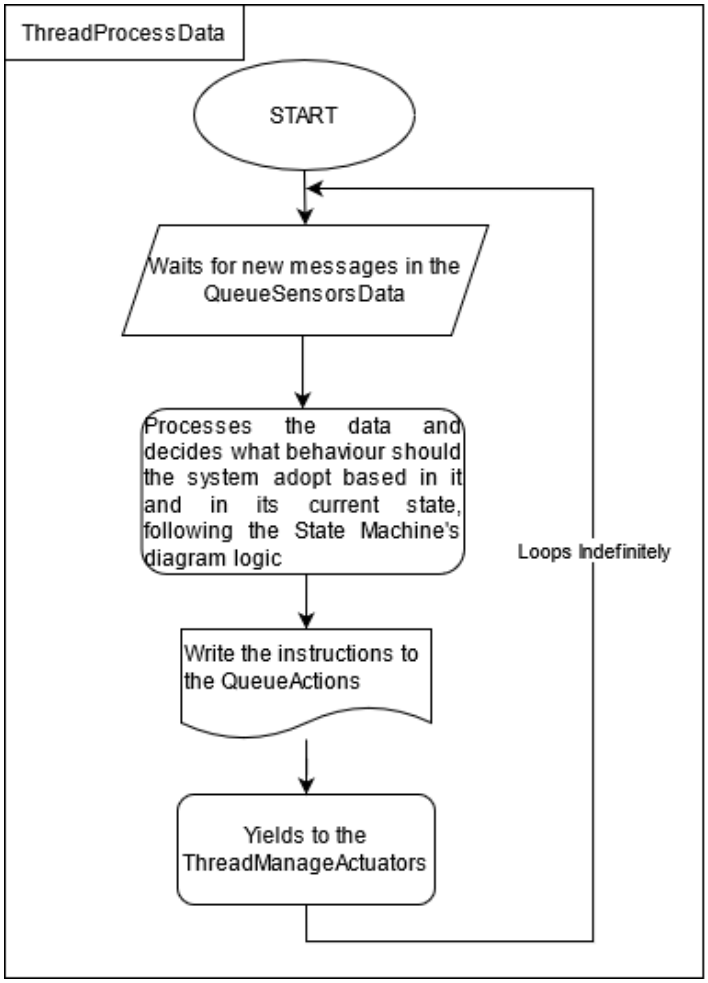

Imagem 7 – Fluxograma da Thread ThreadProcessData

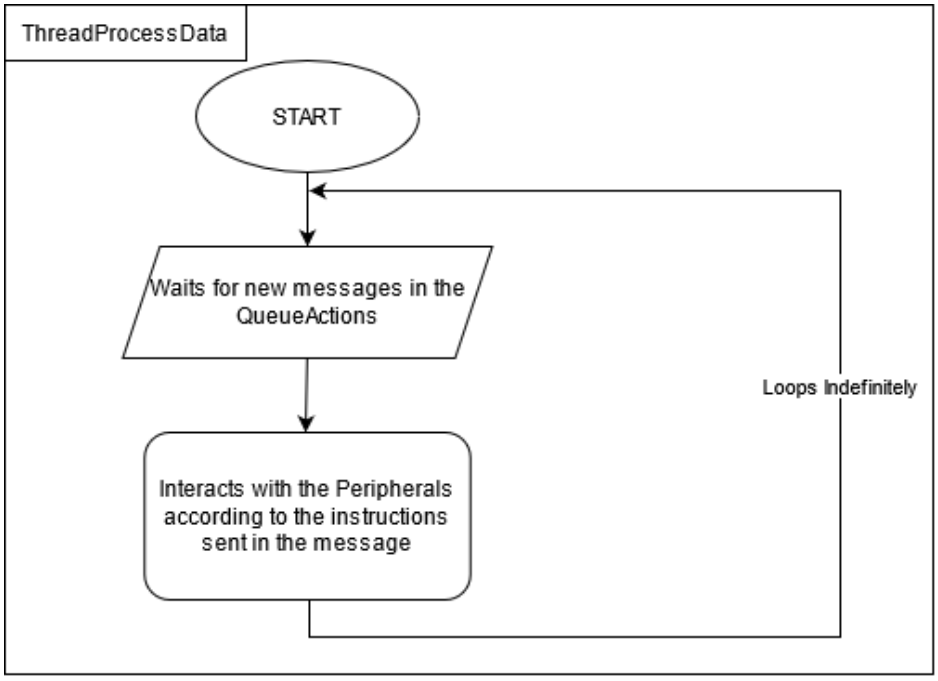

Imagem 8 – Fluxograma da Thread ThreadProcessData

2 Leitura dos dados enviados na UART pelo simulador

Dentre os principais desafios enfrentados no desenvolvimento de uma solução para este projeto, está a leitura dos dados enviados na UART pelo simulador do veículo.

A transmissão de dados do Kit Tiva para o simulador através da porta serial COM1 dá-se da forma usual, no entanto no recebimento há desafios adicionais.

O principal desafio ao ler os dados enviados pelo simulador para o Kit deve-se ao fato de que os textos enviados pelo simulador não utilizam do caractere "\r" para finalização de strings, e, portanto, as funções das bibliotecas de leitura de dados da UART através do arquivo uartstdio.c, fornecido para a utilização nesta disciplina, não funcionaram.

Para contornar este problema, três possibilidades podem ser seguidas:

- A alteração do código do simulador, para que o caractere "\r" seja enviado nativamente;
- A alteração do código da biblioteca para leitura de dados da UART fonecido;
- Uma função de callback que realize a leitura dos dados enviados na porta COM utilizando as funções fonecidas da uartstdio.c em tempo delimitado, desta forma interrompendo o execução assim que a leitura estiver esperando pelo caractere de terminação "\r"