|  |  |  |
| --- | --- | --- |
| Welter Mompean Sozin | \*\*\*\*\*\* | Turma U |
| Guilherme Augusto Amorim Terrell | 168899 | Turma U |
|  |  |  |
| EA075 – Laboratório de Sistemas Embarcados | | |
| Prof. Eric Rohmer  Data de entrega: ??/0?/2022 | | |

Uma imagem contendo Diagrama

Descrição gerada automaticamente

# Projeto 2 (parte 1) – Comunicação serial assíncrona

**Fluxograma máquina de estados**

A primeira parte do projeto consiste basicamente em duas funções: LerComando(), que tem a função de ler caracteres digitados pelo usuário no terminal através de comunicação serial assíncrona (UART) e concatenar os caracteres de modo a formar uma única string, e a função decodComando(), que deve analisar essa string e dar o direcionamento: se string corresponder a uma função válida com os devidos parâmetros (se houver) será exibido no terminal uma mensagem (cada uma das funções válidas tem um padrão de mensagem), caso a string não corresponda a uma função válida ou faltar algum parâmetro será exibido no terminal uma mensagem com o tipo de erro. Não foi utilizado nenhum tipo de espera ativa, dessa forma a condição para concatenação dos caracteres é que o usuário digite algo no terminal.

Diagrama

Descrição gerada automaticamente

* Pseudo-código

**Pseudo-código da função LerComando();**

Void LerComando(){

//variável comando armazena os caracteres

//inicialmente comando = “” -> string vazia

Ler caracter e armazenar na variável carac;

Se o usuário digitou algo no terminal: //carac != -1

Concatenar as variáveis comando e carac; //comando.concat(carac)

}

**Pseudo-código da função decodComando();**

Void decodComando(){

Se houver na string comando o caracter “\*” que indica fim de string:

//Início do tratamento das funções

Se comando == “RETVEL\*”:

Printar no terminal serial a mensagem “VEL: X RPM”;

Se comando == “PARA\*”:

Printar no terminal serial a mensagem “OK PARA”;

Se comando == “EXAUST\*”:

Printar no terminal serial a mensagem “OK EXAUST”;

Se comando == “VENT\*”:

Printar no terminal serial a mensagem “OK VENT”;

//Tratamento da função VEL XXX\*

//A string “VEL XXX\*” deve ter 4 caracteres antes do parâmetro: “V”, “E”, “L” e “ ”

//A string “VEL XXX\*” deve obrigatoriamente possuir após o espaço 3 números

//A string “VEL XXX\*” deve obrigatoriamente possuir após o espaço 4 caracteres

Se o próximo caractere depois do espaço for “\*”:

//”VEL \*”

//Falta o parâmetro de velocidade

Printar no terminal serial a mensagem “ERRO: PARAMETRO AUSENTE”;

Se a partir do caractere espaço houver um número de caracteres diferente de 4:

Printar no terminal serial a mensagem “ERRO: COMANDO INEXISTENTE”;

Senão:

Se o parâmetro numérico XXX >= e XXX <= 100

Converter para inteiro os 3 caracteres correspondentes a velocidade na função “VEL XXX\*”;

Printar no terminal serial a mensagem “OK VEL XXX%”;

Senão:

Printar no terminal serial a mensagem “ERRO: PARAMETRO INCORRETO”;

Se não houver na string comando o caracter “\*” que indica fim de string:

Printar no terminal serial a mensagem “ERRO: COMANDO INEXISTENTE”;

Resetar a variável comando;

}