

EA076 – LABORATÓRIO DE SISTEMAS EMBARCADOS

PROJETO 2 – CONTROLADOR DE VENTILADOR DE TETO

INTRODUÇÃO

Neste experimento, realizaremos o projeto de um sistema para acionamento de um ventilador de teto através de um aplicativo de smartphone e de uma interface *Bluetooth*. O sistema ainda conta com *displays* para indicar o *status* atual do sistema e a velocidade de rotação do ventilador.

OBJETIVOS

- 1- Acionar um motor CC usando PWM e uma ponte H.
- 2- Projetar e construir um *encoder* óptico e usá-lo para estimar a velocidade de um motor.
- 3- Implementar protocolos de comunicação no nível de aplicação usando *parsing*.
- 4- Acionar dispositivos através de uma interface I2C.
- 5- Usar um módulo *Bluetooth* para transmissão de comandos para o sistema via *smartphone*.

ESPECIFICAÇÕES DO SISTEMA

O protótipo irá controlar a velocidade e o sentido de rotação de um pequeno motor CC de 5V, simulando um ventilador de teto. O acionamento do motor será realizado através de uma ponte H, utilizando o CI L293D que, além de servir de amplificador, permitirá o controle do sentido de rotação do motor. A velocidade será ajustada por meio de um sinal PWM.

Um *display* LCD 16x2 (16 caracteres por linha, 2 linhas) deve apresentar, o tempo todo e simultaneamente o status do ventilador, isto é, o **estado atual do sistema** (“VENTILADOR”, “EXAUSTOR” OU “PARADO”), bem como o **valor percentual da velocidade máxima** (0 a 100%).

A velocidade de rotação do ventilador deve ser estimada através de um *encoder* óptico acoplado ao motor. Para construir o *encoder*, pode ser usado um interruptor óptico PHCT203 (ou equivalente) posicionado de maneira que a hélice do ventilador obstrua o feixe de luz IR do interruptor ao passar por ele. A cada obstrução do feixe, um pulso é gerado pelo interruptor. A velocidade de rotação do ventilador pode ser estimada através da contagem do número de pulsos gerados durante certo intervalo de tempo.

A estimativa da velocidade atual do motor em RPM deve ser exibida **o tempo todo** em um conjunto de 4 displays de 7 segmentos. Como não há portas suficientes para acionar todos os displays simultaneamente, utilizaremos a estratégia de multiplexação no tempo combinada com o decodificador CD4511 adotada no projeto anterior. Além disso, utilizaremos o CI PCF8574 para conectar os displays usando a interface I2C do microcontrolador. Desse modo, serão necessários apenas dois pinos do microcontrolador para acionar os 4 displays.

O usuário irá controlar remotamente o sistema através de um *smartphone*. A comunicação entre o sistema e o *smartphone* se dará através de um protocolo serial sem fio via *Bluetooth*. Para isso, será usado um módulo *Bluetooth* HC-06 conectado à porta serial da placa de desenvolvimento Arduino UNO. Usaremos no *smartphone* um aplicativo simples de terminal serial via *Bluetooth*, como o *Serial Bluetooth Terminal*, por exemplo, com os comandos definidos de acordo com a tabela 1 (o caractere “*” termina o comando). A tabela 2 mostra as mensagens de erro que devem ser implementadas no sistema e enviadas ao *smartphone* (em caso de erro, o estado do ventilador não se altera).

Função	Comando	Resposta (serial)
Ajuste de velocidade (%)	VEL xxx* (xxx entre 000 e 100)	OK VEL xxx%
Função ventilador	VENT*	OK VENT
Função exaustor	EXAUST*	OK EXAUST
Parar	PARA*	OK PARA
Retorna a estimativa (X) da velocidade atual do motor em RPM	RETVEL*	VEL: X RPM

Tabela 1 – Lista de comandos via *Bluetooth*

Tipo de erro	Mensagem de resposta (serial)
A string recebida não corresponde a nenhum comando válido.	ERRO: COMANDO INEXISTENTE
O comando foi recebido sem parâmetro (para comandos que exigem parâmetros).	ERRO: PARÂMETRO AUSENTE
O parâmetro do comando não está na faixa correta de valores (para comandos que exigem parâmetros).	ERRO: PARÂMETRO INCORRETO

Tabela 2 – Lista de mensagens de erro.

É importante que o ventilador pare completamente (função parar) antes de trocar o sentido de rotação. Utilize a estimativa da velocidade para garantir que o motor esteja parado.

A figura abaixo mostra o diagrama de blocos do sistema.

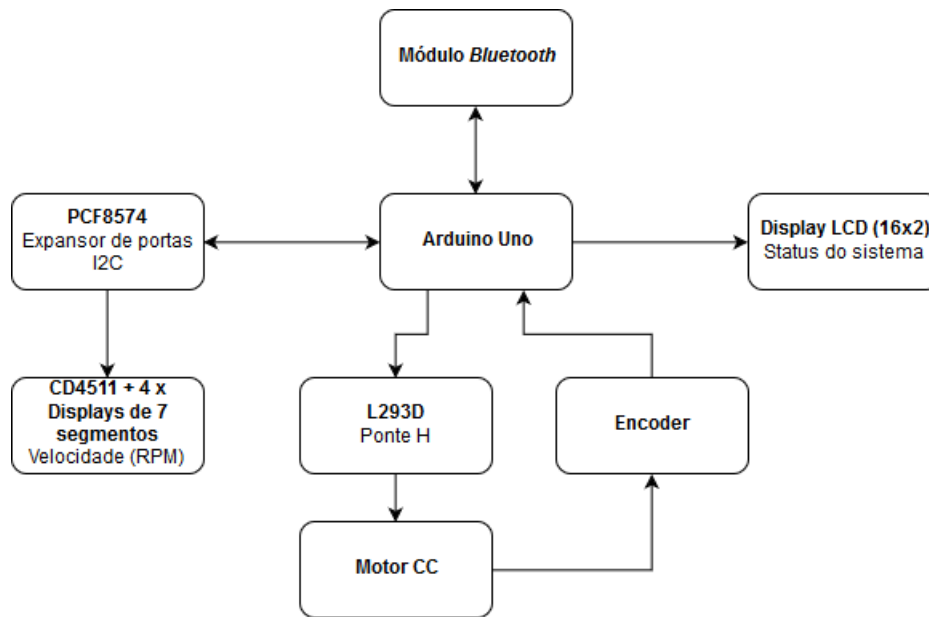


Diagrama de blocos do sistema.

ATENÇÃO!

O módulo *Bluetooth* é alimentado com tensão de 5V. Porém, suas entradas toleram tensões de no máximo 3,3V. Portanto, é necessário utilizar um divisor resistivo para compatibilizar as saídas do Arduino UNO, que assumem tensões de 5V para o nível lógico alto, com os níveis de tensão tolerados pelo módulo.

Não é permitido utilizar espera ativa no programa. Usem o código fornecido no projeto anterior para gerar uma base de tempo através de interrupções periódicas do temporizador.

Guia de referência das funções da biblioteca *LiquidCrystal* usada para o acionamento do display LCD:

<https://www.arduino.cc/en/Reference/LiquidCrystal>

Guia de referência das funções da biblioteca *Wire* usada para comunicação através da interface I2C:

<https://www.arduino.cc/en/Reference/Wire>