

EA076 – LABORATÓRIO DE SISTEMAS EMBARCADOS

Tópico: *Internet* das Coisas

ROTEIRO 3: Armazenamento de Dados e Interface com uma Estação

Prof. Antonio Quevedo e Profa. Wu Shin Ting

Embora os objetos inteligentes e a *internet* sejam integrantes típicos de uma *internet* das coisas (IoT), alguns acessórios são fundamentais para que uma *internet* das coisas possa atingir de forma plena o seu principal objetivo: aprimorar o nosso conhecimento acerca do meio físico que nos rodeia para tomadas de decisão mais seguras. Um destes acessórios é a memória para armazenar os dados que possam ser usados tanto para análise dos problemas como para predição dos problemas [1]. Num sistema real é usualmente disponível um banco de dados para gerenciamento desses dados. Outro complemento relevante são acessos alternativos aos objetos inteligentes quando a rede estiver inoperante, assegurando a manutenção dos serviços básicos localmente.

Nesta terceira parte do curso, vamos ver como adicionar à cada estação um *log* de dados acessível pelos outros objetos da rede e um conjunto de teclado e *display* para construir uma interface simples de comunicação direta entre os objetos inteligentes e um usuário. Na disciplina EA871 vocês já viram como programar um teclado matricial. Neste curso faremos o projeto de conexão das linhas e das colunas de um teclado com o nosso microcontrolador para que este responda de forma satisfatória às instruções dos programas que vocês desenvolveram anteriormente. A unidade de memória e o *display* adotam, respectivamente, interfaces de comunicação seriais I2C e SPI. Veremos como conectar estes periféricos ao nosso microcontrolador e como programar via *Processor Expert* o nosso microcontrolador para comunicar com eles.

A distribuição das aulas para cada tarefa é uma sugestão, para que os alunos tenham uma noção do andamento do trabalho.

Estrutura:

Primeira aula

- Conexão física do *display* gráfico Nokia 51-10 (SPI)
- Programação do módulo SPI do FRDM-KL25Z
- Visualização de temperatura, velocidade e sentido do motor CC 5V

Segunda aula

- Conexão física do teclado matricial (GPIO)
- Técnica de *debouncing*.

Terceira aula

- Controle in loco da velocidade e do sentido do motor CC 5V via teclado
- Conexão física do módulo de memória EPROM 24C (I2C)
- Programação do módulo I2C do FRDM-KL25Z

Quarta aula

- Armazenamento dos dados IoT (periodicamente a temperatura e alterações na velocidade e no sentido do motor, rotulados com os instantes de tempo de registro)
- Acesso aos dados estatísticos dos objetos inteligentes

Tarefas (a constarem no relatório):

Display gráfico Nokia 5110

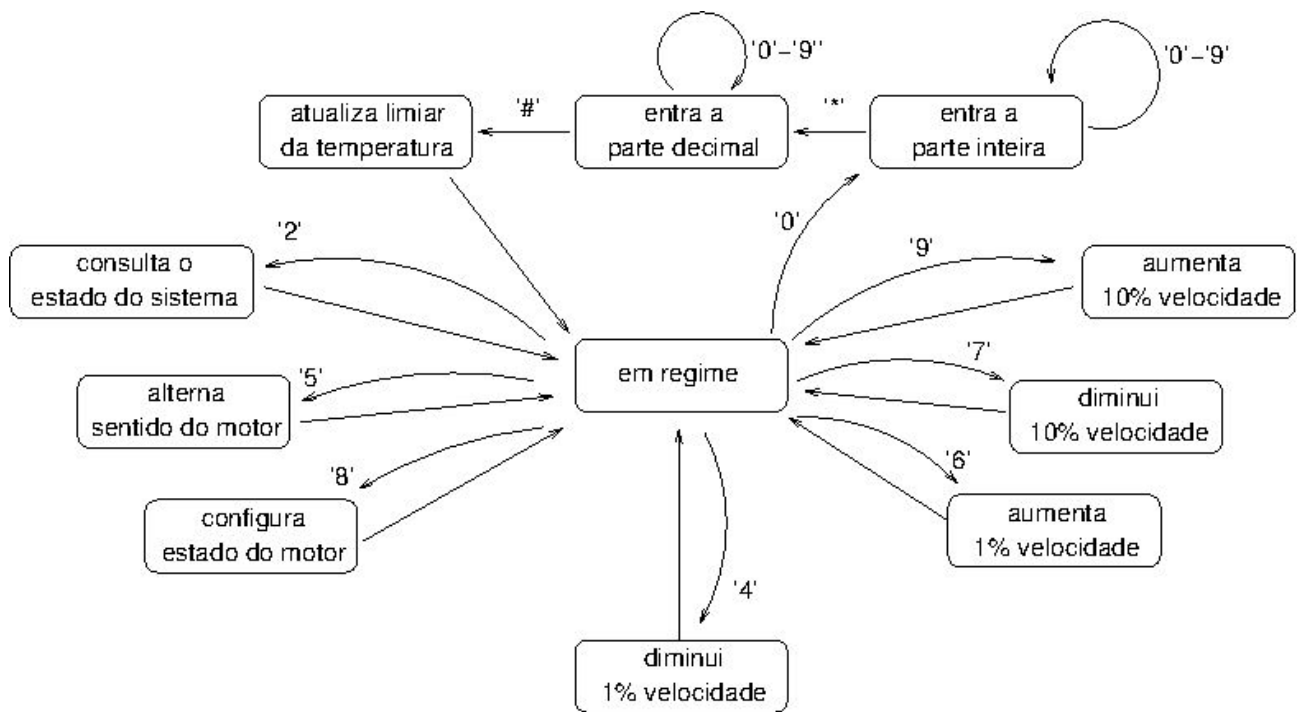
- 1) Projete a interface entre o *display* gráfico Nokia 5110 [2] e o nosso microcontrolador. Este *display* e o teclado 4x3 são ligados à nossa placa de circuito impresso via uma caixa *header* com trava para cabo fita de 16 vias, como mostra em [3]. Utilize Eagle [4] para desenhar o esquemático da interface entre o microcontrolador e a caixa. Use os pinos PTC5 e PTC6 para a interface SPI (*CLK* e *DIN*), e os pinos PTE2 (*LIGHT*), PTE3 (*RST*), PTE4 (*CE*) e PTE5 (*DC*). Monte o circuito com *wire-wrap* na sua placa de circuito impresso.
- 2) Escreva um programa que envie ao *display* gráfico em três linhas, a data, o horário e o último valor da temperatura amostrada. O programa deve se conectar ao MQTT e assinar o tópico “EA076/grupoXn/adjust”. Ao receber uma mensagem por este tópico, o programa deve acertar o relógio. Use no celular o app *IoT MQTT Panel*, com um item “Time Picker” para enviar a hora certa para o tópico acima.

Teclado 4x3

- 3) Projete a interface entre um teclado 4x3 [5], cujo componente no Eagle pode ser acessado em [6], e o nosso microcontrolador. Determine os valores de resistência dos resistores de *pull-up/pull-down*. Como já mencionado antes, este teclado e o *display* Nokia são conectados a nossa placa de circuito impresso através de um *header*. Ligue as colunas C1, C2 e C3 (com os resistores de *pull up*) respectivamente aos pinos PTA5, PTA12 e PTA13, e as linhas L1, L2, L3 e L4 respectivamente aos pinos PTD2, PTD3, PTD4 e PTD5. Utilize Eagle [4] para desenhar o esquemático da interface entre o microcontrolador e a caixa (PINH-2x8). Monte o circuito com *wire-wrap* na sua placa de circuito impresso.
- 4) Todas as chaves de contato podem apresentar *bouncing* [7], introduzindo bordas indesejáveis nos sinais que podem gerar interrupções espúrias. Projete uma solução que amenize este problema
- 5) Escreva um programa que ecoe no *display* gráfico o valor da tecla digitado no *keypad*.

Interface Local da Estação

- 6) Usando como base o item (8) do Roteiro 2, adicione uma interface amigável em que um usuário pode consultar o estado do sistema (velocidade, direção e modo que devem permanecer no *display* por 5 segundos) via *display* Nokia, ao digitar a tecla “2”, e controlar a velocidade do motor com as teclas “4” (diminui 1%), “6” (aumenta 1%), “7” (diminui 10%) e “9” (aumenta 10%). A tecla “5” altera a direção do motor (ventilador / exaustor) e a tecla “8” altera o modo (On, Off ou Auto). Como realimentação visual da consulta, a nova velocidade, nova direção ou novo modo é visualizada no *display*. Ao apertar a tecla “0”, o sistema entra no modo de entrada de temperatura-limiar, usando o “*” como ponto decimal e o “#” para finalizar a entrada. A figura ilustra os estados do sistema e as transições entre eles em resposta às teclas digitadas por um usuário.



Log de dados

- 7) Projete a interface entre uma unidade de memória EEPROM 24C6 [8] e o nosso microcontrolador através dos pinos PTC10 e PTC11. Desenhe com uso de Eagle [4] o esquemático da interface projetado. Monte o circuito com *wire-wrap* na sua placa de circuito impresso.
- 8) A partir do item 6, adicione ao programa a capacidade de armazenar periodicamente (a cada 5 segundos) os valores de temperatura amostrados na memória. Ao receber a mensagem “Dump” no tópico *command*, envie a sequência de temperaturas (um *publish* para cada valor de temperatura, com uma casa decimal) pelo tópico *log*. No aplicativo, inclua um gerador de gráfico para receber estes valores.

Apresentação:

Para cada programa escrito, apresente no relatório os projetos exportados com a documentação em *doxygen*, e um vídeo mostrando o funcionamento da estação implementada.

Bibliografia:

- [1] Sean Boyce. Review: IoT Data Logging Services with MQTT
<https://hackaday.com/2017/10/31/review-iot-data-logging-services-with-mqtt/>
- [2] ETT. User's Manual of Graphic LCD “ET-NOKIA LCD 5110”
ftp://ftp.dca.fee.unicamp.br/pub/docs/ea076/complementos/User_Manual_ET_LCD5110.pdf

[3] SATE. Pinagem da Conexão para LCD e Teclado.

ftp://ftp.dca.fee.unicamp.br/pub/docs/ea076/complementos/Pinagem_Key_LCD.pdf

[4] Embarcados. Tutorial Eagle. <https://www.embarcados.com.br/serie/tutorial-eagle/>

[5] Crameda. Datasheet Keypad.

https://www.crameda.com/pdf_files/eng/keypads/Monobloc-3x4-4x4-DS-E.pdf

[6] Componente Eagle Keypad.

<ftp://ftp.dca.fee.unicamp.br/pub/docs/ea076/complementos/KeyPad3x4.lbr>

[7] Jack Ganssler. A Guide to Debouncing, or, How to Debounce a Contact in Two Easy Pages.

<http://www.ganssle.com/debouncing.htm>

[8] Componente Memória Atmel 24C6.

<ftp://ftp.dca.fee.unicamp.br/pub/docs/ea076/complementos/atmel.lbr>

Anexo

ESP8266EX Standalone Device

