

# EA076 – LABORATÓRIO DE SISTEMAS EMBARCADOS

## Tópico: *Internet das Coisas*

### ROTEIRO 2: Comunicação dos Objetos Inteligentes

*Prof. Antonio Quevedo e Profa. Wu Shin Ting*

Dispondo de uma rede de comunicação, no nosso caso baseado no protocolo **MQTT** [1], veremos nesta segunda parte do curso como ela pode ser aplicada na implementação de uma rede de objetos (inteligentes), ou *internet of things* (IoT) [2], em que os objetos possam responder aos comandos remotos ou proverem serviços à distância. Estas novas capacidades dos objetos impulsionaram uma nova geração de tecnologia, como casas inteligentes, carros inteligentes, objetos vestíveis, e monitoramento do estado de trânsito e desastres naturais em tempo real [3].

Com uso de microcontroladores, podemos transformar objetos estáticos, como sensores e atuadores, em objetos inteligentes capazes de se **auto-identificarem**, **interagirem com o meio físico** em que estão inseridos e **com outros objetos inteligentes** conectados à rede. Os microcontroladores dispõem de uma grande variedade de interfaces de comunicação com os periféricos, o que facilita a sua conexão com os sensores e atuadores. Os microcontroladores possuem funções de conversão entre os sinais analógicos (meio físico) e sinais digitais (processamento digital), o que simplifica o circuito de aquisição dos sinais oriundos de sensores e o circuito de geração dos sinais para atuadores. Os microcontroladores são providos de processadores e unidades de memória, permitindo que sejam programados os “comportamentos” dos objetos. E, mais importante, alguns microcontroladores já tem as facilidades de conexão às redes sem fio integradas. Mesmo quando não as tenham, como é o caso do nosso microcontrolador, encontram-se periféricos de conexão compatíveis.

Nesta segunda parte do curso, vamos transformar dois objetos estáticos (um sensor de temperatura LM-61 e um motor de corrente contínua 5V) em dois objetos inteligentes numa rede de comunicação. Na disciplina EA871 vocês já tiveram oportunidades para ver como se aumenta a “autonomia” dos sensores e dos atuadores com uso de um microcontrolador (interações com o meio físico) . Podemos programar, por exemplo, a periodicidade de amostragem dos sensores e as condições sob as quais um atuador deve atuar e como atuar. Aqui vamos focar na auto-identificação, associando a cada objeto um tópico (canal de comunicação) do protocolo **MQTT**, e na conectividade entre os objetos inteligentes, inclusive um ser humano via celular. Notem que vamos aqui usar indistintamente os termos **tópico** e **canal** como sinônimos.

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 sensor de temperatura LM-61
- Programação do conversor A/D do FRDM-KL25Z
- Conversão dos valores brutos em temperatura
- Programação do módulo RTC (*Real Time Clock*) do microcontrolador
- Validação dos valores capturados e dos instantes da captura através do Terminal USB

## Segunda aula

- Conexão física do motor CC 5V, através de uma ponte H.
- Programação do PWM para controle de velocidade e do sentido do motor CC 5V
- Controle de velocidade e sentido do motor CC 5V através do Terminal USB

## Terceira aula

- Acesso à temperatura e ao sentido de rotação do motor via celular pessoal
- Controle da velocidade e do sentido de rotação do motor via celular pessoal

## Tarefas:

### RTC e sensor de temperatura

- 1) Projete a interface entre o sensor de temperatura LM61 e o nosso microcontrolador. Utilizando os componentes de Eagle [4] mostrados no esquemático no Anexo, desenhe o esquemático desta interface. Monte o circuito com *wire-wrap* na sua placa de circuito impresso.
- 2) Projete a interface entre um módulo RTC (*real time clock*) e o nosso microcontrolador, lembrando que já existe um módulo RTC integrado nele [5]. Consulte uma forma de utilizá-lo na implementação de um relógio em [6]. Desenhe o esquemático e monte o conexão do sensor de temperatura (LM-61) com o microcontrolador usando o pino **PTB0**, e também a ligação entre pinos do microcontrolador para que o módulo RTC do nosso microcontrolador opere corretamente [6]. Monte o circuito com *wire-wrap* na sua placa de circuito impresso.
- 3) Veja em [7] a relação entre a tensão e a temperatura do sensor de temperatura LM-61. Escreva a equação que relaciona um dado bruto em 16 *bits* gerado pelo conversor AD do nosso microcontrolador no valor de temperatura em graus Celsius com uma casa decimal. Utilize somente tipo de dado inteiro.
- 4) Escreva um programa que faz amostragem das temperaturas numa periodicidade de 1 s, e envie para o Terminal USB a hora do RTC, no formato HH:MM:SS, seguida de um espaço e do valor de temperatura amostrado com resolução de 0.1°C. Documente os códigos com a sintaxe de Doxygen [10].

### Motor CC 5V

- 5) Projete a interface entre um motor CC 5V com velocidade e sentido de rotação configuráveis e o nosso microcontrolador. A interface deve contemplar o uso de um módulo TPM em modo PWM do nosso microcontrolador [5] para controlar a sua velocidade (**PTE20**) e duas saídas digitais (**PTE29** e **PTE30**) para controlar o sentido de rotação. Ela deve conter o *driver* de ponte H L293D [8]. Utilizando os componentes de Eagle [4] mostrados no esquemático no Anexo, desenhe o esquemático desta interface. Monte o circuito com *wire-wrap* na sua placa de circuito impresso.

Obs.: Caso ache necessário isolar fisicamente o circuito de controle de velocidade e o de potência, integre os optoacopladores [9] para isolar fisicamente o circuito do

microcontrolador do circuito de acionamento do motor. Determine os valores de resistência dos resistores limitadores de corrente e de *pull-up*.

- 6) Programe o módulo TPM do nosso microcontrolador para que ele gere sinais PWM cuja largura seja programável.
- 7) Escreva um programa que controla o sentido e a velocidade de rotação (0-100%) do motor conforme a entrada dada pelo Terminal USB, finalizada com ENTER. Documente os códigos com a sintaxe de Doxygen [10].

## Comunicação

- 8) Escreva um programa contendo os seguintes passos (Documente os códigos com a sintaxe de Doxygen [10]):
  - a) Conecte com a rede WiFi EA076 e com o *Broker* MQTT (item 7 do Roteiro 1).
  - b) Faça o *Subscribe* dos seguintes tópicos (*X* é a letra da turma e *n* é o número do grupo):
    - i) EA076/grupoXn/dir (**dir**)
    - ii) EA076/grupoXn/power (**power**)
    - iii) EA076/grupoXn/mode (**mode**)
    - iv) EA076/grupoXn/limiar (**threshold**)
  - c) A cada 1 segundo, publique uma *string* contendo o valor da temperatura em °C, com resolução de 0.1 no tópico EA076/grupoXn/temp (item 4 deste roteiro)
  - d) Faça repetidamente:
    - i) Verifique a recepção de uma mensagem do módulo ESP
    - ii) Caso receba uma, extraia as *strings* de tópico e de mensagem
    - iii) Caso a mensagem chegue pelo tópico
      - (1) **dir**, configure o sentido de rotação do motor, de acordo com a mensagem recebida (“Vent” ou “Exhaust” para modo ventilador ou modo exaustor) (item 7 deste roteiro)
      - (2) **power**, configure a velocidade do motor, de acordo com a mensagem recebida (*string* representando um número entre 0 e 100) (item 7 deste roteiro)
      - (3) **mode**, configure o modo de ativação do motor, de acordo com a mensagem recebida (“On”, “Off” ou “Auto” para motor ligado, desligado ou em modo de acionamento automático quando a temperatura for maior que o **threshold** definido.) (item 7 deste roteiro)
      - (4) **threshold**, convirta a mensagem (*string* representando um valor de temperatura com resolução de 0.1°C) em um inteiro representando décimos de grau, armazenado em uma variável que determina o limiar de temperatura para ativar o motor no modo automático (item 7 deste roteiro)
- 9) Instale o aplicativo IoT MQTT Panel [11] no seu celular e o configure para se conectar ao *Broker*. Depois, adicione os seguintes elementos no painel:

- a) Um *Vertical Meter* ou *Gauge* na faixa de 0 até 40, que recebe o tópico EA076/grupoXn/temp para apresentar a temperatura medida em nossa estação
  - b) Um *Radio Buttons* com dois itens: Ventilador e Exaustor, que publica no tópico EA076/grupoXn/dir a *string* “Vent” ou “Exhaust” respectivamente
  - c) Um *Slider* que publica no tópico EA076/grupoXn/power um valor entre 0 e 100
  - d) Um *Radio Buttons* com três itens: Ligado, Desligado e Automático, que publica no tópico EA076/grupoXn/mode a *string* “On”, “Off” ou “Auto” respectivamente
  - e) Um *Text Input* que publica no tópico EA076/grupoXn/threshold
- 10) Conecte o aplicativo ao *Broker* e verifique a atualização de temperatura no painel, além da atuação dos controles do motor.

## Apresentação:

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

## Bibliografia:

- [1] MQTT - Protocolos para IoT (<https://www.embarcados.com.br/mqtt-protocolos-para-iot/>)
- [2] Internet of Things (IoT) | What is IoT | How it Works | IoT Explained | Edureka  
(<https://www.youtube.com/watch?v=LlhmzVL5bm8>)
- [3] 10 Real World Applications of Internet of Things (IoT) – Explained in Videos  
(<https://www.analyticsvidhya.com/blog/2016/08/10-youtube-videos-explaining-the-real-world-applications-of-internet-of-things-iot/>)
- [4] Embarcados. Tutorial Eagle. <https://www.embarcados.com.br/serie/tutorial-eagle/>
- [5] Freescale. KL25 Sub-Family Reference Manual, Rev. 3, September 2012.  
<ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/ARM/KL25P80M48SF0RM.pdf>
- [6] Adrian Sanchez Cano. Using RTC module on FRDM-KL25Z  
<https://community.nxp.com/docs/DOC-94734>
- [7] *Datasheet* LM-61. <ftp://ftp.dca.fee.unicamp.br/pub/docs/ea871/datasheet/LM61.pdf>
- [8] *Datasheet* L293D. <ftp://ftp.dca.fee.unicamp.br/pub/docs/ea076/datasheet/l293d.pdf>
- [9] Electronic Tutorials. Optocoupler Tutorial.  
<https://www.electronics-tutorials.ws/blog/optocoupler.html>
- [10] Doxygen. <http://www.doxygen.nl/>
- [11] IoT MQTT Panel <https://play.google.com/store/apps/details?id=snr.lab.iotmqttpanel.prod>

