

# Trabalho final

## 1 Visão geral

O trabalho final é uma oportunidade para vocês aplicarem os conhecimentos estudados neste semestre em uma aplicação IoT – *Internet of Things* criando sensores inteligentes. Este projeto deve ser executado **em duplas**.

Os sensores inteligentes criados serão compostos pela interação de três componentes principais:

1. Sensor(es).
2. Microcontrolador ATmega328P.
3. Módulo WiFi ESP-01 ou NodeMCU.

O ATmega328P deverá ser programado por vocês e fará a leitura e interpretação dos dados não tratados de um ou mais sensores. Esses dados **devem** ser filtrados, caso necessário, e **podem** ser exibidos em um *display* localmente. Após o condicionamento dos dados do sensor, estes valores devem ser enviados ao módulo WiFi ESP-01 ou NodeMCU. O módulo WiFi funcionará como uma espécie de *gateway* WiFi que enviará os dados tratados do sensor à Internet através do protocolo MQTT (*Message Queuing Telemetry Transport*). Visão geral do sistema é ilustrada pela Figura 1.

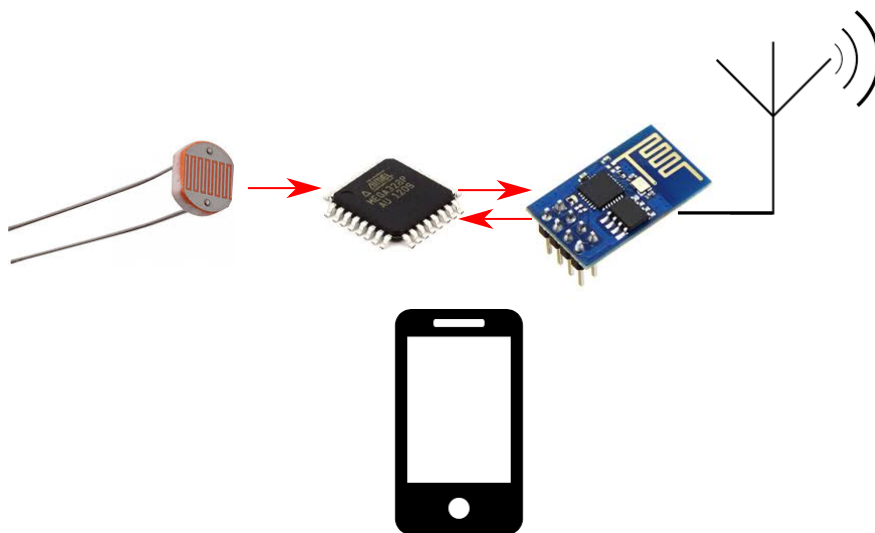


Figura 1: Visão geral do “sensor inteligente”

## 2 Requisitos

Para o cumprimento deste projeto, deve-se selecionar um dos sensores abaixo. Cada dupla deve escolher apenas um. Sensores mais simples, como LDRs, deverão ter o sinal filtrado e o valor enviado deve ser composto pela combinação deles. Atuadores são opcionais e os modelos disponíveis estão listados abaixo. É possível que a dupla utilize outro sensor, desde que aprovado pelo professor previamente.

O módulo WiFi ESP-01 ou NodeMCU já está previamente programado com um *firmware* customizado. A comunicação entre o ATmega328P e o módulo utilizará uma simplificação do protocolo Modbus RTU sobre UART com *baudrate* de 9600.

### 2.1 Sensores

Deve-se selecionar alguns dos seguintes sensores:

- Luminosidade (vários LDRs): filtro de média móvel e aquisição acionada por temporizador.
- Umidade (vários sensores): filtro de média móvel e aquisição acionada por temporizador.
- NTC: filtro de média móvel e aquisição acionada por temporizador.
- GPS (Skylab SKM53).
- Acelerômetro e giroscópio (Módulo GY-521 MPU6050).
- Pressão atmosférica e temperatura (BMP0805).
- Umidade e temperatura (DHT11).
- Acelerômetro, bússola, pressão e temperatura (módulo GY-80).
- Corrente, efeito hall (ACS712)

### 2.2 Atuadores

O uso de atuadores é um requisitos de projeto e o atuador deve responder comandos recebidos por MQTT. Os seguintes atuadores estão disponíveis mas outros podem ser escolhidos desde que aprovado previamente pelo professor.

- Servo motor.
- Leds.
- Motores CC.
- Displays (gráficos e caractere)

## 2.3 Comunicação (Modbus RTU)

O módulo WiFi ESP-01 ou NodeMCU possui um *firmware* customizado. A comunicação entre o módulo WiFi e o ATmega328P deverá ser feita utilizando a porta serial (UART) de ambos os microcontrolador e com uma simplificação do protocolo Modbus RTU.

O Modbus é um protocolo de comunicação serial originalmente desenvolvido pela Schneider Electric em 1979. É muito utilizado na comunicação entre dispositivos industriais como controladores lógicos programáveis (CLPs). Modbus é atualmente um padrão de comunicação serial e é amplamente utilizado na comunicação entre dispositivos no meio industrial. Existem várias versões do Modbus como o RTU que utilizaremos neste trabalho. A versão RTU permite conectar o sistema industrial a um computador conhecido como RTU (*Remote Terminal Unit*) que faz a integração do sistema industrial em um ambiente supervisório chamado SCADA (*Supervisory Control and Data Acquisition*).

A versão RTU permite comunicação serial em uma representação binária e compacta dos dados. O formato do pacote RTU é composto pelo endereço do dispositivo, um comando, os dados e a verificação cíclica de redundância. Uma mensagem RTU deve transmitir os dados de forma contínua sem tempos entre os caracteres. Os pacotes devem ser separados por momentos de silêncio.

Um exemplo de pacote RTU pode ser visualizado na Figura 2, onde:

- *addr*: endereço do dispositivo que se deseja enviar uma mensagem.
- *cmd*: comando que se deseja enviar, geralmente escrita ou leitura.
- *reg*: qual registrador do dispositivo deseja-se escrever ou ler.
- *data*: dado que se deseja escrever no registrador ou número de registradores que se deseja ler.
- *crc*: verificação cíclica de redundância.

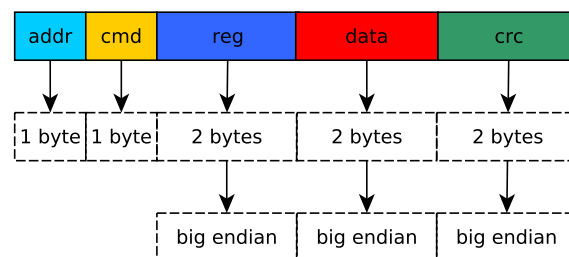


Figura 2: Exemplo de frame Modbus RTU

### 2.3.1 Características

As características do protocolo Modbus suportados pelo módulo WiFi ESP-01 são:

- endereço: 0x15
- comando de escrita: 0x01
- comando de leitura: 0x02

### 2.3.2 Registradores

Os registradores disponíveis no módulo ESP-01 são representados pela Figura 3, onde:

- estado: estado da conexão WiFi.
- atuador\_x: valores provenientes da “Internet” que podem acionar algum atuador conectado no ATmega328P. Deve ser lido explicitamente pelo ATmega328P.
- sensor\_x: valores escritos pelo ATmega328P que serão enviados pelo módulo à “Internet”.

	descrição	endereço
somente leitura	estado	0x00
	atuador_0	0x01
	atuador_1	0x02
	atuador_2	0x03
	atuador_3	0x04
escrita	sensor_0	0x05
	sensor_1	0x06
	sensor_2	0x07
	sensor_3	0x08
2 bytes		

Figura 3: Exemplo de frame Modbus RTU

### 2.3.3 Exemplos

Exemplos de pacotes Modbus RTU especificados para esse trabalho podem ser visualizados nas Figuras 4 e 5

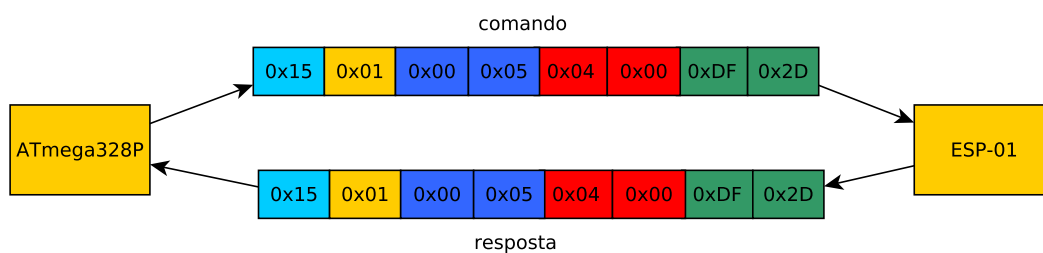


Figura 4: Enviar número 0x0400 para o endereço 0x05.

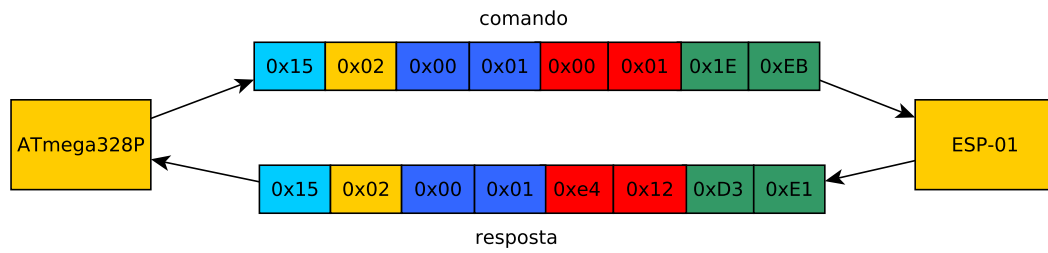


Figura 5: Solicitar o valor do registrador 0x0001.

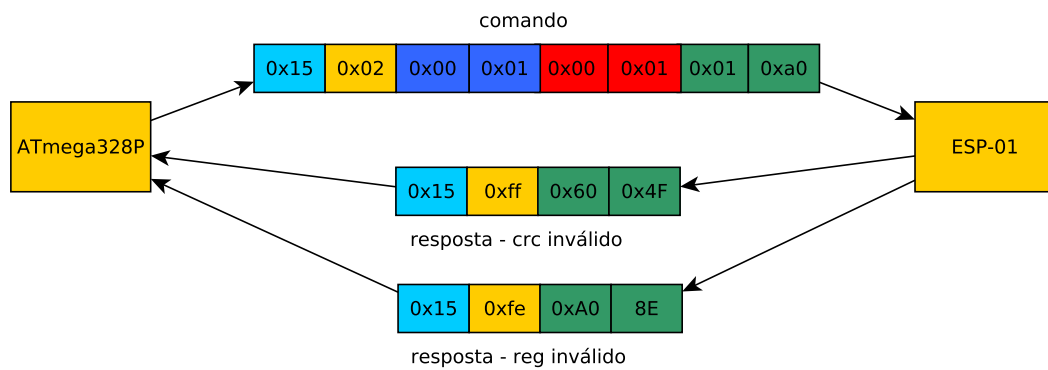


Figura 6: Códigos de erros.

### 2.3.4 Hardware

A conexão entre do ATmega328P e o módulo WiFi é ilustrada pela Figura 7.

## 2.4 Cálculo do CRC

O CRC pode ser calculado pela função abaixo (<https://www.lammertbries.nl/comm/info/crc-calculation.html>).

```
uint16_t CRC16_2(uint8_t *buf, int len)
{
    uint32_t crc = 0xFFFF;
    int i;

    for (i = 0; i < len; i++)
    {
        crc ^= (uint16_t)buf[i];    // XOR byte into least sig. byte of crc

        for (int i = 8; i != 0; i--) {    // Loop over each bit
            if ((crc & 0x0001) != 0) {    // If the LSB is set
                crc >>= 1;                // Shift right and XOR 0xA001
                crc ^= 0xA001;
            }
            else                            // Else LSB is not set
                crc >>= 1;                // Just shift right
        }
    }
    return crc;
}
```

## 2.5 Comunicação MQTT

Toda a comunicação com a Internet utilizando o protocolo MQTT é realizada pelo módulo ESP-01. O protocolo MQTT utiliza-se do modelo de publicação/subscrição para a comunicação, por

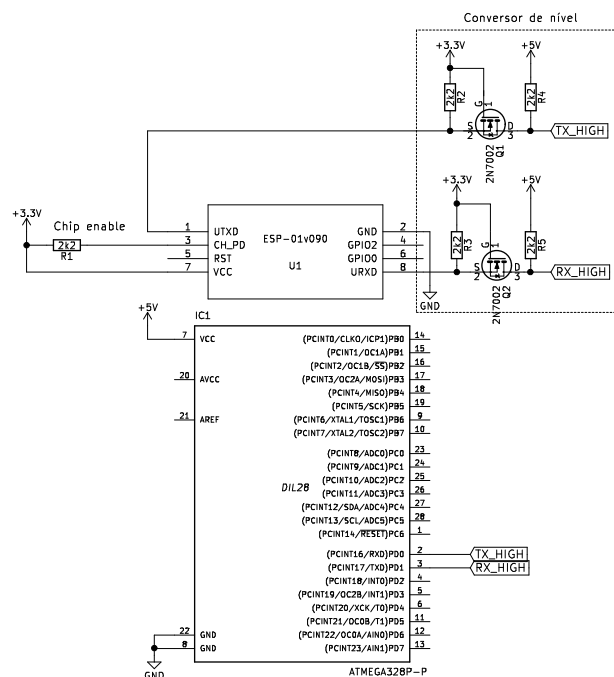


Figura 7: Conexão elétrica entre o ATmega328P e o módulo WiFi.

meio de um *Broker*. O *Broker* é o responsável por receber, enfileirar e enviar as mensagens dos clientes que publicam em determinado tópico  $x$  para os clientes subscritos neste tópico  $x$ . O cliente *publisher* é o responsável por se conectar ao *broker* e publicar as mensagens para um determinado tópico de interesse. E o cliente *subscriber* é responsável por se conectar ao *broker* e se “inscrever” em determinado tópico, para assim, receber todas as mensagens de interesse deste tópico publicadas por outros clientes.

Cada dupla receberá um módulo ESP-01 programado para conectar em uma rede WiFi e publicar os dados escritos nos registradores do protocolo Modbus (sensor\_x) nos tópicos respectivos. Além disso, o módulo WiFi subscrive-se nos tópicos relacionados com os atuadores (atuador\_x). A relação entre componentes do trabalho pode ser visualizado nas Figuras 8 e 9.

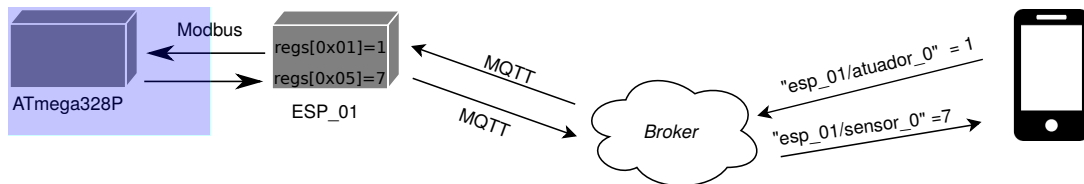


Figura 8: Interação entre os componentes dos sensores inteligentes.

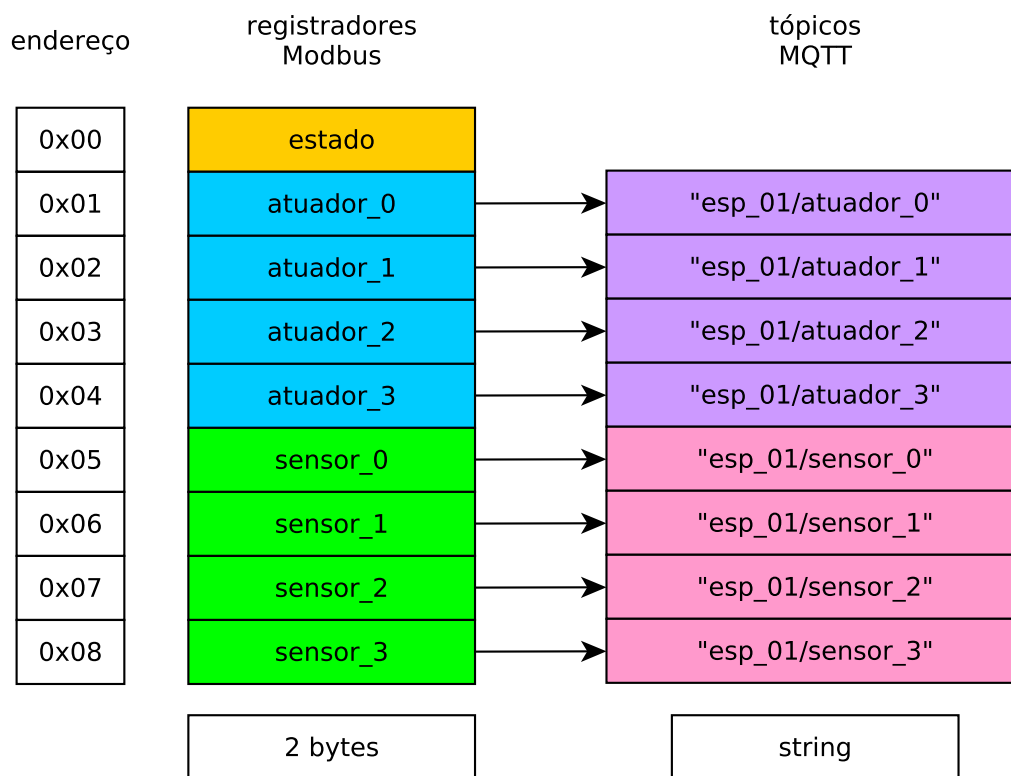


Figura 9: Relação entre os tópicos MQTT e os endereços Modbus.

### 3 Repositório de código

Para fins de organização, seu trabalho deve ser disponibilizado em repositório GiT, de preferência o GitHub ([www.github.com/](http://www.github.com/)). O uso do repositório deve cumprir os seguintes requi-

sitos:

- O uso do repositório não é apenas para a entrega. Deve haver um histórico razoável *commits* com descrição adequada.
- O projeto deve incluir o arquivo README.md que contempla a descrição do projeto, dependências, como instalar, executar e outras informações relevantes.
- O projeto deve incluir o arquivo CONTRIBUTORS.md que apresenta detalhes dos membros do projeto coma contribuição individual de cada membro.
- Sua implementação deve segui as melhores práticas de programação. Mantenha um código multiplataforma e tente ciar um arquivo *Makefile*.

## 4 Apresentação

No final do semestre, deve-se apresentar o projeto para o professor. Sua apresentação deve responder as seguintes perguntas:

- Background
  - O que é o seu projeto?
  - Como funciona sensor/atuador escolhido?
  - Quais estruturas / algoritmos utilizados?
- Código
  - Como foi implementado?
  - Quais estruturas foram utilizadas?
- Análise:
  - Como você sabe que a implementação está correta?
  - O que seus resultados relevam?
  - Qual a complexidade das operações básicas?
- Conclusão
  - O que vocês aprenderam?
  - Algo surpreendeu-os?

## 5 Avaliação

Os projetos são pontuados conforme a tabela abaixo. **Lembrando que somente serão avaliados projetos apresentados.**



Métrica	Pontos
Implementação	4
Testes	2
Apresentação	1
Organização do projeto / Estilo de código / Documentação	3
Total	10