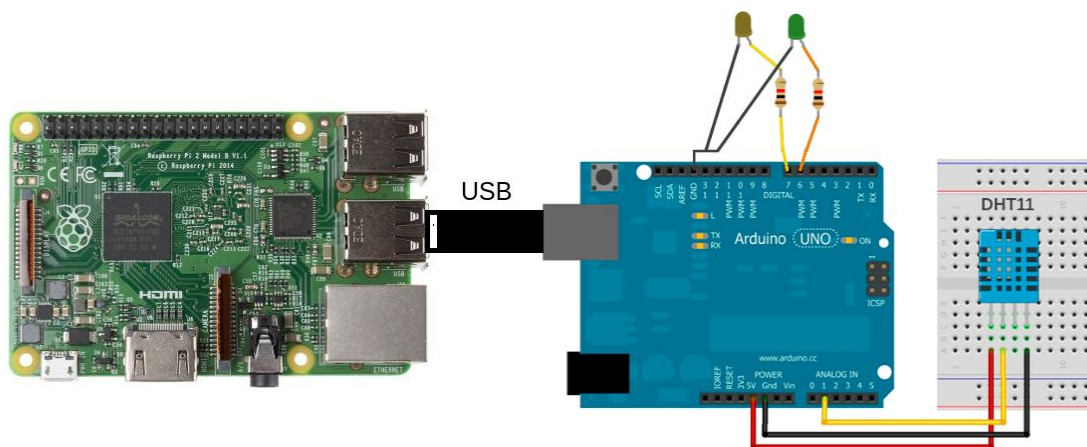


Etapa b.2: Implementar a leitura dos sensores de temperatura e umidade

1. Esquemático do microcontrolador e sensores

As leituras de temperatura e umidade serão feitas utilizando-se o sensor DHT11. O microcontrolador estará conectado ao sensor e LEDs para comunicação visual conforme figura 1.

Figura 1 - Esquemático de conexão da raspberry pi, microcontrolador e sensor

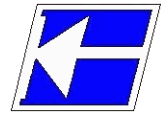


Dica: Para leituras das grandezas físicas, recomenda-se utilizar a biblioteca DHT sensor library (<https://github.com/adafruit/DHT-sensor-library>) e a Adafruit Unified Sensor. A própria IDE do Arduino pode ser usada para programação do microcontrolador.

2. Protocolo de comunicação com o microcontrolador

A interface de comunicação entre o microcontrolador e a raspberry pi será a porta serial /dev/ttyACM0 (porta emulada na conexão USB). Desta maneira, a saída dos dados do ATmega328p se dará através dos pinos 0 (Rx) e 1 (Tx) da placa Arduino Uno.

Haverão dois códigos: o do microcontrolador, que lerá os valores de temperatura e umidade do sensor; e o da raspberry, que requisitará as leituras ao microcontrolador. O formato das requisições e respostas é descrito abaixo.

**Projeto 2020/1**

Para o protocolo de comunicação o *frame* da tabela 1 deve ser implementado.

Número da requisição	Comando	Dados	Checksum 16 bits complemento de um
N_REQ	CMD	DATA	CHECKSUM
4 bytes	1 byte	4 bytes	2 bytes

Os campos do frame de dados podem ser feitos através de *structs* em linguagem C ou através das *dataclasses* em Python.

(<https://docs.python.org/3/library/dataclasses.html>)

Comandos:

N_REQ = número aleatório

CMD = 0x54h Leitura de temperatura ou

CMD = 0x48 Leitura de umidade

DATA = 0x00000000h

CHECKSUM = O checksum deve ser feito como o complemento de um da soma simples dos campos N_REQ, CMD e DATA (em C -> checksum= ~ soma)

As respostas do microcontrolador devem ser:

N_REQ = mesmo número enviado na requisição

CMD = 0xFFh Resposta do comando de leitura

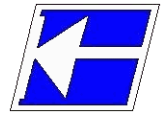
DATA = valor de temperatura ou umidade (4 bytes)

CHECKSUM = O checksum deve ser feito como o complemento de um da soma simples dos campos N_REQ, CMD e DATA (em C -> checksum= ~ soma)

OBS.: Sempre deve-se verificar se existe erro na comunicação através da análise do checksum recebido.

3. Acesso remoto

O acesso remoto aos hardwares será feito através de uma sessão ssh (Secure



Projeto 2020/1

Socket Shell). Para isso, é necessária a chave privada *rpi_rsa* que foi enviada por e-mail. A passphrase da chave é *aluno*

Comando:

```
ssh -i path_to_key/rpi_rsa user@189.34.36.204 -p 12000  
Enter passphrase for key 'rpi_rsa': aluno
```

Para visualização dos LEDs, pode-se visitar a webcam do projeto:

<http://189.34.36.204:12666>

4. Transferência de arquivos

Para transferir arquivos para a Raspberry Pi, pode-se utilizar o comando scp.

a. Exemplo para transferência de um arquivo .hex:

```
scp -P 12000 -i rpi_rsa  
/tmp/build13772453501128518130.tmp/humidity.cpp.hex  
user@189.34.36.204:/home/user/
```

5. Script para gravar o código do microcontrolador

Uma vez compilado o código do microcontrolador ATmega328p, presente na placa Arduino Uno, o binário de saída será carregado utilizando-se o script *load_uC.sh*.

OBS.: a flag *-e* deve ser indicada como *um*, conforme exemplo de uso abaixo.

Usage: *./load_uC.sh -e exp -b binary*

-e Selecionar o experimento: um, dois, tres, quatro

-b Arquivo que sera carregado na memoria do microcontrolador da placa Arduino

Exemplo de uso:

```
./load_uC.sh -e um -b humidity.cpp.hex
```



6. Entrega da tarefa

Os códigos devem estar armazenados dentro da pasta `/home/user/sensing` (user é o nome do usuário criado para cada dupla) até o dia 17/09.

Cronograma de entrega

Etapas	Deliverable	Data Limite
b.1	Apresentação de 10 minutos sobre o broker MQTT escolhido e seu funcionamento.	03/09/2020
b.2	Códigos e demonstração da leitura dos sensores.	17/09/2020
b.3	Códigos e demonstração da comunicação MQTT de leitura dos sensores	24/09/2020
b.4	Códigos e demonstração da visualização Web	08/10/2020