**ATLAS**

ENV-20

* Baixar e adicionar a biblioteca Gravity da Atlas (https://www.atlas-scientific.com/files/atlas\_gravity.zip)
* Conectar a Placa Gravity ao Arduino e executar a leitura.

ENV-50

PH

* Fios preto (terra) e marrom (sinal) são do PH
* Vermelho e Branco são temperatura (PT-1000 sem polaridade)
* Na Carrier Board, o Chip do EZO PH, vai conectado com o TX e RX do EZO na mesma linha onde tem o TX e RX da Carrier Board
* Se piscar verde, está em UART, se piscar em Azul, está em I2C. (para trocar, fonte: <https://how2electronics.com/interfacing-atlas-scientific-ph-sensor-with-arduino-via-uart-i2c/>)
* Executar projeto trocarUARTtoI2CAtlas e digitar i2c,99 com CR ativo

|  |  |
| --- | --- |
| EZO BOARD | Arduino UNO |
| VCC | 3.3v |
| GND | GND |
| TX | D3 |
| RX | D2 |

* Projeto de leitura dos valores, pinagem:

|  |  |
| --- | --- |
| EZO BOARD | Arduino UNO |
| VCC | 3.3v |
| GND | GND |
| TX | A4 |
| RX | A5 |

ORP

* Fios preto (terra) e marrom (sinal) são do ORP
* Na Carrier Board, o Chip do EZO ORP, vai conectado com o TX e RX do EZO na mesma linha onde tem o TX e RX da Carrier Board
* Se piscar verde, está em UART, se piscar em Azul, está em I2C. (para trocar, fonte: <https://how2electronics.com/interfacing-atlas-scientific-ph-sensor-with-arduino-via-uart-i2c/>)
* Executar projeto trocarUARTtoI2CAtlas e digitar i2c,98 com CR ativo

|  |  |
| --- | --- |
| EZO BOARD | Arduino UNO |
| VCC | 3.3v |
| GND | GND |
| TX | D3 |
| RX | D2 |

* Projeto de leitura dos valores, pinagem:

|  |  |
| --- | --- |
| EZO BOARD | Arduino UNO |
| VCC | 3.3v |
| GND | GND |
| TX | A4 |
| RX | A5 |

TEMPERATURA

* Fios verde e vermelho são GND e Fios Branco e Preto são sinal.
* Na Carrier Board, o Chip do EZO RTD, vai conectado com o TX e RX do EZO na mesma linha onde tem o TX e RX da Carrier Board
* Se piscar verde, está em UART, se piscar em Azul, está em I2C. (para trocar, fonte: <https://how2electronics.com/interfacing-atlas-scientific-ph-sensor-with-arduino-via-uart-i2c/>)
* Executar projeto trocarUARTtoI2CAtlas e digitar i2c,62 com CR ativo

|  |  |
| --- | --- |
| EZO BOARD | Arduino UNO |
| VCC | 3.3v |
| GND | GND |
| TX | D3 |
| RX | D2 |

* Projeto de leitura dos valores, pinagem:

|  |  |
| --- | --- |
| EZO BOARD | Arduino UNO |
| VCC | 3.3v |
| GND | GND |
| TX | A4 |
| RX | A5 |

OD

* Do conjunto de fio preto, a malha é (terra) e marrom (sinal) são do OD
* Na Carrier Board, o Chip do EZO DO, vai conectado com o TX e RX do EZO na mesma linha onde tem o TX e RX da Carrier Board
* Se piscar verde, está em UART, se piscar em Azul, está em I2C. (para trocar, fonte: <https://how2electronics.com/interfacing-atlas-scientific-ph-sensor-with-arduino-via-uart-i2c/>)
* Executar projeto trocarUARTtoI2CAtlas e digitar i2c,97 com CR ativo

|  |  |
| --- | --- |
| EZO BOARD | Arduino UNO |
| VCC | 3.3v |
| GND | GND |
| TX | D3 |
| RX | D2 |

* Projeto de leitura dos valores, pinagem:

|  |  |
| --- | --- |
| EZO BOARD | Arduino UNO |
| VCC | 3.3v |
| GND | GND |
| TX | A4 |
| RX | A5 |

Referência interessante:

<https://www.warburtech.co.uk/products/sensors/orp/atlas.scientific.ezo-orp.ezo.orp.circuit/>

Ligar placa EZO diretamente:

<https://atlas-scientific.com/blog/temperature-sensor-arduino/>

**DFRobot**

Temperatura - DS18B20

* Projeto de leitura dos valores, pinagem:

|  |  |
| --- | --- |
| SENSOR | Arduino UNO |
| VERMELHO | 5v |
| PRETO | GND |
| AMARELO | D2 |

PH

* Projeto de leitura dos valores, pinagem:

|  |  |
| --- | --- |
| SENSOR | Arduino UNO |
| VERMELHO | 5v |
| PRETO | GND |
| AZUL | A0 |

ORP - DS18B20

* Biblioteca necessária: <https://github.com/cdjq/DFRobot_Analog_ORP_Sensor>
* Projeto de leitura dos valores, pinagem:

|  |  |
| --- | --- |
| SENSOR | Arduino UNO |
| VERMELHO | 5v |
| PRETO | GND |
| AZUL | A1 |

Turbidez

* Projeto de leitura dos valores, pinagem:

|  |  |
| --- | --- |
| SENSOR | Arduino UNO |
| VERMELHO | 5v |
| PRETO | GND |
| AZUL | A2 |

Oxigênio dissolvido

* Referência: <https://wiki.dfrobot.com/Gravity__Analog_Dissolved_Oxygen_Sensor_SKU_SEN0237>
* Projeto de leitura dos valores, pinagem:

|  |  |
| --- | --- |
| SENSOR | Arduino UNO |
| VERMELHO | 5v |
| PRETO | GND |
| AZUL | A3 |

**VERNIER**

\*Baixar bibliotecas da Vernier pelo gerenciador de bibliotecas do Arduino.

Na biblioteca da VERNIER, para trabalhar com o ARDUINO MEGA, deve se alterar a linha 114 do arquivo VernierLib.cpp para A2 ao invés de A5 no comando analogREAD.

Outra alteração é comentar as linhas 240, 241, 332 e 333.

Na conexão física, os pinos de A4 e A5 da SHIELD da Vernier devem ser conectados aos pinos SDA e SCL do ArduinoMEGA (20 e 21). O A4 no 20, o A5 no 21.