**Environmental Monitoring Station**

**Parts List**

Part Altronics Product no.

* 1 x Vero Board 55mmx43mm H0711
* 4 x PCB Supp Flush 10mm H1381
* 1 x Cable Gland w/Strain Relief H4403A
* 1 x DB9 Line Male P3000
* 1 x DB9 Female Line P3010
* 1 x DB 15/DB9 Backshell RF Shield P3092
* 2 x Spacer Screws 8mm P3310
* 2 x header Socket 5 Pin 8.5mm P5390
* 2 x header Socket 3 Pin 8.5mm
* 2 x Header Pin 1 way P5430
* 1 x Header Pin 2 way
* 1 x Header Pin 3 way
* 1 x Arduino EtherTen board
* 1 x Humidity Sensor (DHT22)
* 1 x Pressure Sensor(BMP085 or BMP180)
* 1 x 1m Garland 14/0 .20mm 7 core screened cable
* 9 x male to male solderless flexible breadboard jumper cable Arduino wires
* 1 x set 3D printed case for the EtherTen Arduino (case and base)
* 1 x set 3D printed case for the sensors (case and base)

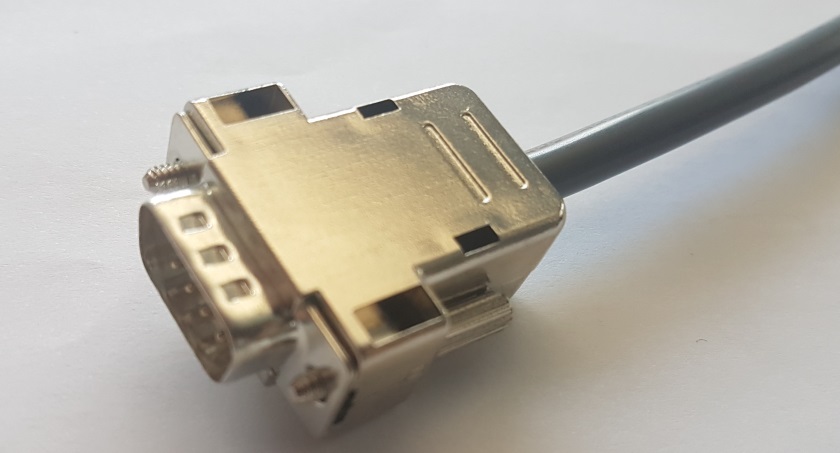
**3D Printed Arduino and Sensor Case Files**

* etherTen\_base.stl
* etherTen\_case.stl
* Sensor\_Base.stl
* Sensor\_Case.stl

**Assembling the Environmental Monitoring Station**

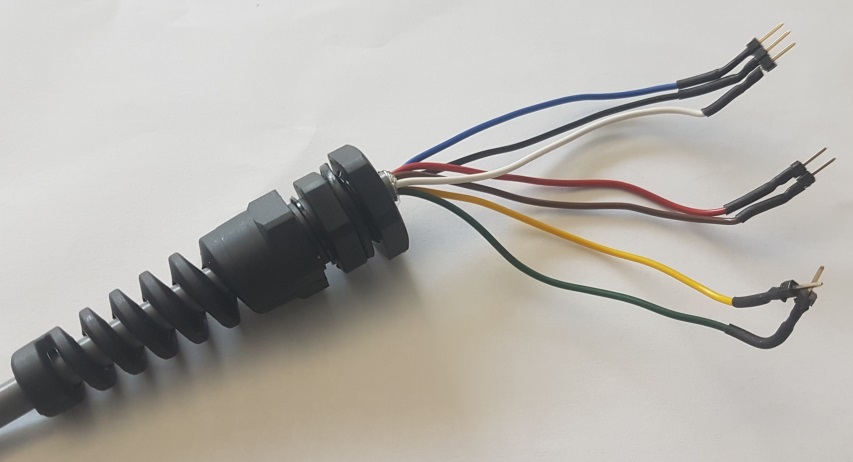
***Connecting cable***

1. Strip 90mm of the casing away from the 1m cable on one end and 25mm from the other.
2. Slide the Cable gland onto the cable at the stripped 90mm end and line up with the stripped end of the 7 core cladding.
3. Strip 5mm from each exposed single core wire end and tin each wire.
4. Solder the tinned single core wires to the 9DB male plug according to table 1.
5. Assemble the Back shell RF shield around the 9DB male plug.



*Figure 1: Assembled cable – Sensor End*

1. Slide heat shrink onto each of the 9 single core wires.
2. At the 90mm stripped single core end of the cable, solder the yellow and green single core wires to a Header Pin 1 way each.
3. Solder the red and brown single core wires to a Header Pin 2 way.
4. Solder (in this order) the white, black and blue single core wires to the Header Pin 3 way.
5. Use a heat gun to shrink the heat shrink around each of the single core Header Pin connections.



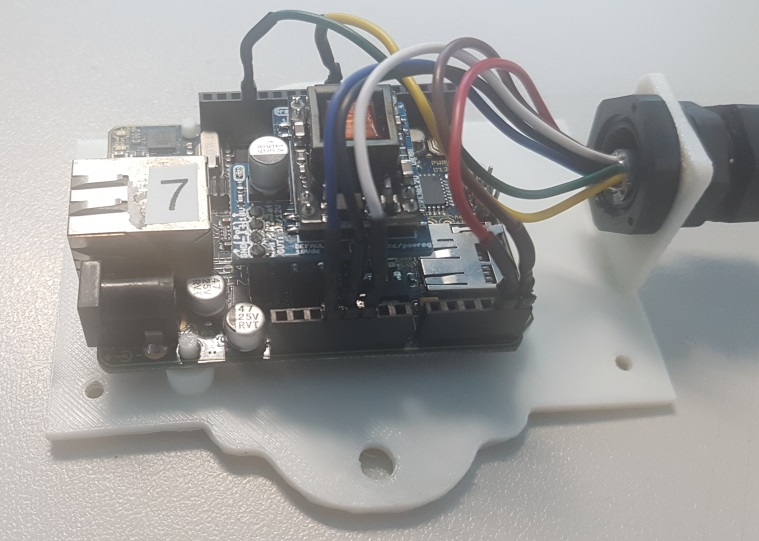
*Figure 2: Assembled cable – Arduino End*

*Table 1: 7 Core Cladded Cable 9DB Plug Connection*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Arduino | Cable 7 Cores + Screen | 9DB Male Plug |  | 9DB Female Socket | Single Cores | Sensor Type | Sensor Function |
|
| Analogue | 1m Length | Pins |  | Pins |  | Pressure | Pins |
| GND | White | 1 | **---** | 1 | White |  | GND |
| SCL | Brown | 2 | **---** | 2 | Brown | **---------------** | SCL |
| SDA | Red | 3 | **---** | 3 | Red | **---------------** | SDA |
| Vcc (+3.3V) | Blue | 4 | **---** | 4 | Blue | **---------------** | Vcc (+3.3V) |
|  | Not Used | 5 | **X** | 5 | Not Used | **X** | Not Used |
| Digital |  |  |  |  |  | Humidity / Temp Sensor |  |
| GND | Green | 6 | **---** | 6 | Green | **---------------** | GND |
| DATA | Yellow | 7 | **---** | 7 | Yellow | **---------------** | DATA |
| Vin (+5V) | Black | 8 | **---** | 8 | Black | **---------------** | Vin (+5V) |
|  | Not Used | 9 | **X** | 9 | Not Used | **X** | Not Used |
|  | Screen | Shell | **---** | Shell | Screen |  |  |

***EtherTen Case***

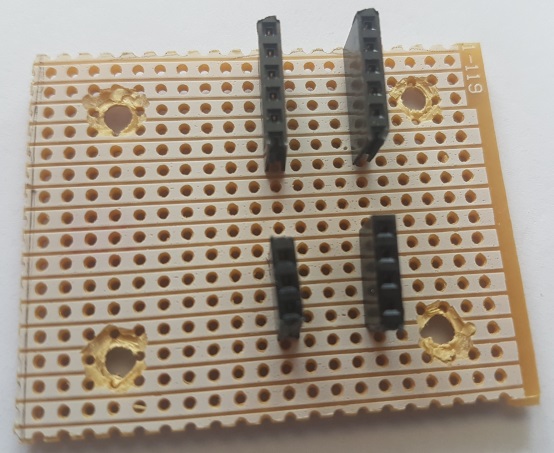
1. Sit the EtherTen Arduino onto the EtherTen Base pins and ensure that it sits flat.
2. Unscrew the first nut on the cable gland and attach to the EtherTen base by threading the single core cables through the supporting hole keeping the rubber stopper to the outside.
3. Tighten the nut back on to secure in place.
4. Connect the single core Header Pin terminated cables to the Arduino as per Table 1.
5. Clip the base/case closed.



*Figure 3: EtherTen Setup*

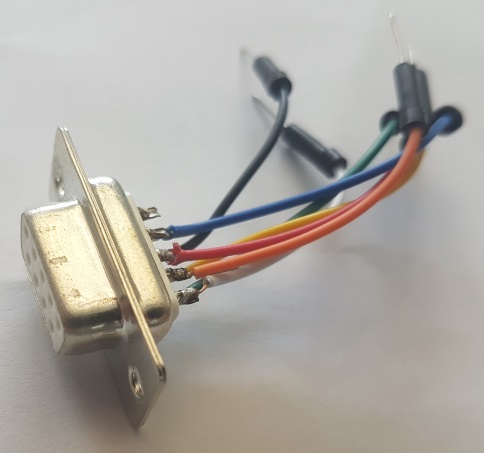
***Sensor Case***

1. On the Vero Board, mark a dot 7.5mm from the right side and 9.5mm from the bottom.
2. From this first dot, mark three more dots forming a 27mm x 35mm rectangle.
3. Drill the marked holes with a 3mm drill.
4. Solder one 5 pin header socket on the Vero board in the 3rd line above the bottom drilled hole and 1st row from the right.
5. Leaving 5 rows clear solder the 3 pin header socket on the same line.
6. 4 lines above these header sockets solder the 3 and 5 pin header sockets to match the rows of the already soldered sockets.



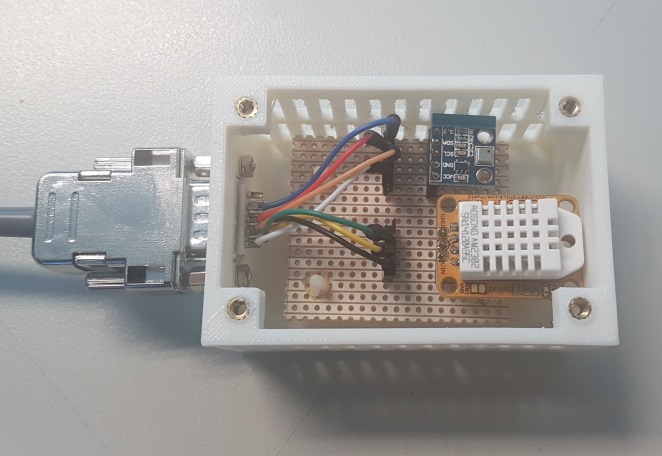
*Figure 4: Assembled Vero Board*

1. Cut 9 male to male solderless flexible breadboard jumper cable wires so that there is a length of 45mm (plus termination pin) on each.
2. Tin the 9 wires and solder to the DB9 female plug (leaving ports 5 and 9 empty).



*Figure 5: DB9 Sensor Plug*

1. Insert four PCB Support Flush Mounts into the 3D printed Sensor Case making sure that each one clicks into place.
2. Place the assembled Vero Board into the case and onto the four PCB Support Flush Mounts making sure that the Header Sockets are closest to the hole for the female plug.
3. Thread the wires from the assembled DB9 female plug through the opening of the Sensor Case and screw the plug onto the case using the Spacer Screws.
4. Place the Humidity Sensor in the 3 Pin Header Socket furthest from the DB9 female plug.
5. Place the Pressure Sensor in the 5 Pin Header Socket furthest from the DB9 female plug.
6. Using Table 1, connect each of the DB9 female single core wires to the appropriate Header Pin socket in the row closest to the plug.
7. Screw the Sensor Base onto the Sensor Case.
8. Connect the DB9 Male end of the cable to the sensor box.



*Figure 6: Assembled Sensor Case*

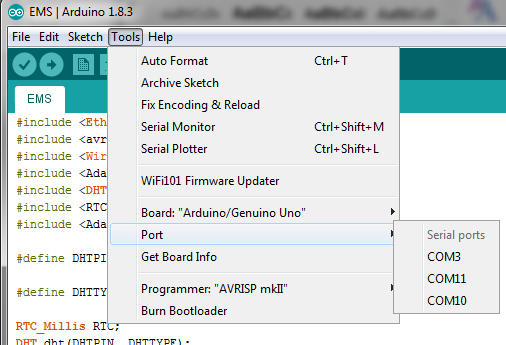
**Uploading the Code to the Arduino**

***Supporting Documents and Files***

* Adafruit\_BMP085.h
* DHT.h
* Ethernet.h
* SPI.h
* Wire.h
* EMS.ino

***Procedure***

1. Make sure to have all the .h Arduino library files loaded into the correct Arduino directory.
2. Connect the device to a computer with the Arduino software installed.
3. Select Tools dropdown menu, port and then the COM port (it will come up with a different number to 3, 10 or 11 which are reserved ports).



*Figure 7: Selecting COM port*

1.  Compile/upload the EMS.ino Arduino file onto the Arduino using the button.
2. Ctrl+Shift+M will bring up the serial monitor to ensure that the Arduino and sensors are functioning correctly.

**Deploying the EMS**

***Deploying the Device***

Once a suitable location has been chosen, determine if the closest network the EMS will be connected to is Power over Ethernet (PoE) enabled.

If so:

* Plug an Ethernet cable into the Arduino and the network. This will power up the device.

If not:

* A PoE switching power supply will also need to be connected by plugging an Ethernet cable into the network and the LAN port of the switch.
* Then plug another Ethernet cable from the PoE port of the switch to the device.