Files needed: import amazon-freertos(v.exp)

**Basic IOT-to-XMC connection**

Under application\_code >> common\_demos >> include >> aws\_clientcredential.h

Set the MQTT channel, IOT thing name and the Wi-Fi network name, password and security. Connecting to a mobile hotspot is also possible.

**Basic DPS310 configuration**

Under application\_code >> infineon\_code >> dps310\_hal.c || dps310\_hal.h || dps310.c || dps310.h

These 4 files were written by Infineon engineers for configuring the DPS310 barometric pressure sensor.

Dps310.c and .h are used to configure the pressure sensor settings. Further documentation can be found here: <https://www.infineon.com/dgdl/Infineon-DPS310-DataSheet-v01_01-EN.pdf?fileId=5546d462576f34750157750826c42242>

\_hal.c and .h stands for hardware abstraction layer and gets the pressure reading after initialising settings using dps310.c and .h

**Operation**

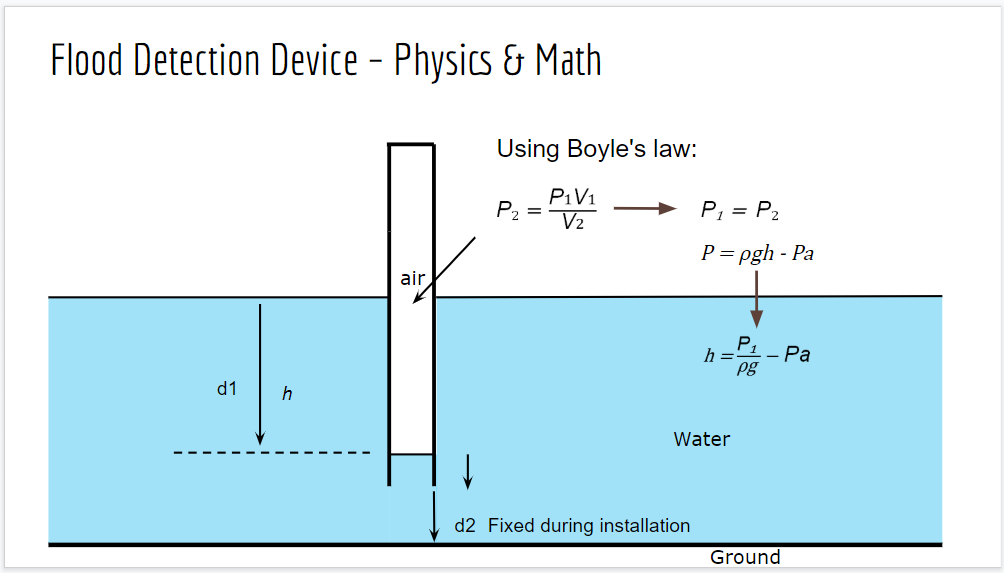
1. The program initialises and runs tasks, one of which is aws\_hello\_world (application\_code >> common\_demos >> source).
2. Function **prvMQTTConnectAndPublishTask** starts by initialising the DPS310 and attempts to connect to the MQTT broker (see **Basic IOT-to-XMC connection**)
3. If successful, it then prepares to calibrate the pressure sensor by measuring the ambient temperature. It calls get\_ref\_pressure, a function in global.c. The purpose of this is to get the atmospheric pressure as a reference. This reference will be subtracted from future readings to obtain the pressure difference within the tube (and ignore atmospheric pressure).
4. Then, it will call function **prvPublishNextMessage** to print a reading once every 5 seconds.
5. The function **prvPublishNextMessage** gets the pressure and temperature from the function dps310\_hal\_get\_temp\_press. The readings are used by water\_level.c to measure the water level (see **Calculate water level**) for the theory. In water\_level.c, the temperature, air pressure inside the tube (named as air\_prs) and the ref\_pressure (from global.c) are used to calculate the water level in metres(m), then converted to centimetres(cm).
6. Several values like pressure, temperature, address, coordinates, location and water level are put into a string and published to the topic.
7. If it is published successfully, sound\_buzzer (function of sound\_buzzer.c) will check the water level and sound the busser according to arbitrary values (e.g. sound 2 seconds for 30 cm of water).

**Setting up:**

1. Familiarise with setting up via this pdf walkthrough: <https://www.infineon.com/dgdl/Infineon-DAVE_Quick_Start-GS-v02_00-EN.pdf?fileId=5546d4624cb7f111014d059f7b8c712d>
2. For this case, select File >> import...  and the Infineon >> DAVE Project >> Browse…
3. Find the file amazon-freertos(v.exp), select it. A couple of options will come up in the Project List box. Select the one that starts with aws\_demos\_trustx\_exp and click finish.

Note: There is a DAVE APP that is used: DIGITAL\_IO\_0. The programmer should take note of this

**Calculation of water level**



1. Purpose is to find d1, with d2 being a fixed value.
2. Using Boyle’s Law, for a given temperature, a decrease in volume will result in an increase in pressure.
3. P₂ = P₁V₁ / V₂, remove both V and you get P₁ = P₂
4. P₂ is broken down into its constituent parts, ρ denoting density, g denoting gravitational force and h denoting height. In our equation we also subtract Pa, denoting atmospheric pressure (ref\_pressure from global.c)

Things to note: The main.c initialises with an attempt to connect to a Wi-Fi network or mobile hotspot. If it cannot find a suitable connection, it will fail to initialise and do nothing. Press the “Reset” button on the board.