**GitHub Copilot**

Used 1 reference

To integrate the functionality of the AM2315C.ino program into the AOSongAM2315.cpp and AOSongAM2315.h files, you would need to modify these files to use the AM2315C class instead of the AOSongAM2315 class. Here's a step-by-step plan:

1. In AOSongAM2315.h, replace the AOSongAM2315 class definition with the AM2315C class definition from AM2315C.h. You'll also need to replace the #include "AOSongAM2315.h" line with #include "AM2315C.h".
2. In AOSongAM2315.cpp, replace all instances of AOSongAM2315 with AM2315C. You'll also need to replace the #include "AOSongAM2315.h" line with #include "AM2315C.h".
3. In GRWX3.cpp, replace all instances of AOSongAM2315 with AM2315C. You'll also need to replace the #include "AOSongAM2315.h" line with #include "AM2315C.h".
4. Update the getSensorData() method in AOSongAM2315.cpp to use the read() method from the AM2315C class. This method should return a boolean value indicating whether the sensor data was successfully updated.
5. Update the sensorValues array in AOSongAM2315.cpp to store the humidity and temperature readings from the AM2315C sensor. You can get these readings using the getHumidity() and getTemperature() methods from the AM2315C class.

Here's an example of how you might update the getSensorData() method and sensorValues array:

bool AOSongAM2315::getSensorData() {

    int status = AM2315C.read();

    if (status == AM2315C\_OK) {

        sensorValues[AM2315\_HUMIDITY\_VAR\_NUM] = AM2315C.getHumidity();

        sensorValues[AM2315\_TEMP\_VAR\_NUM] = AM2315C.getTemperature();

        return true;

    } else {

        return false;

    }

}

Please note that this is a high-level plan and the exact changes you need to make might vary depending on the specifics of your code. Also, remember to update any other methods in AOSongAM2315.cpp and AOSongAM2315.h that use the AOSongAM2315 class to use the AM2315C class instead.