/\*\*

\* @file AOSongAM2315.cpp

\* @copyright 2017-2022 Stroud Water Research Center

\* Part of the EnviroDIY ModularSensors library for Arduino

\* @author Sara Geleskie Damiano <sdamiano@stroudcenter.org>

\*

\* @brief Implements the AOSongAM2315 class.

\*/

#include <AM2315C.h>

#include "AOSongAM2315.h"

// The constructor - because this is I2C, only need the power pin

// This sensor has a set I2C address of 0XB8

AOSongAM2315::AOSongAM2315(TwoWire\* theI2C, int8\_t powerPin, uint8\_t measurementsToAverage)

: Sensor("AOSongAM2315", AM2315\_NUM\_VARIABLES, AM2315\_WARM\_UP\_TIME\_MS,

AM2315\_STABILIZATION\_TIME\_MS, AM2315\_MEASUREMENT\_TIME\_MS, powerPin,

-1, measurementsToAverage),

\_i2c(theI2C) {

am2315ptr = new AM2315C(\_i2c);

}

AOSongAM2315::AOSongAM2315(signed char powerPin, unsigned char i2cAddress) {

// Initialize your class members here

}

AOSongAM2315::~AOSongAM2315() {}

String AOSongAM2315::getSensorLocation(void) {

return F("I2C\_0x38");

}

bool AOSongAM2315::begin() {

if (!am2315ptr->begin()) {

Serial.println("Could not find a valid AM2315C sensor, check wiring!");

return false;

}

Serial.println("AM2315C sensor initialized successfully!");

return true;

}

bool AOSongAM2315::addSingleMeasurementResult() {

float temp = am2315ptr->getTemperature();

float hum = am2315ptr->getHumidity();

if (temp == -9999 || hum == -9999) {

Serial.println("Failed to read from AM2315C sensor!");

return false;

}

verifyAndAddMeasurementResult(AM2315\_TEMP\_VAR\_NUM, temp);

verifyAndAddMeasurementResult(AM2315\_HUMIDITY\_VAR\_NUM, hum);

// Unset the time stamp for the beginning of this measurement

\_millisMeasurementRequested = 0;

// Unset the status bits for a measurement request (bits 5 & 6)

\_sensorStatus &= 0b10011111;

return true;

}