```
/*
 * This is the .cpp file for the Si7051 sensor
 * The Si7015 is being used as the Globe Thermometer Sensor
 * The bulk of this library was retrieved on line:
 * https://github.com/closedcube/ClosedCube_Si7051_Arduino
 *
 * Part 1 of this library was retrieved on line,
 * while Part 2 was written by MECH 45X Team 26
 *
 * Team 26 does not fully understand how the on line
 * library works, so Part 1 is not commented
 *
 * Team 26 commented Part 2 as they wrote Part 2
 * and understand how the code in Part 2 works
 *
 * Please note that the Globe Thermometer does not
 * measure Mean Radiant Temperature (MRT), it
 * actually measures the globe temperature.
 * MRT is calculate later using air temperature and
 * globe temperature.
 */

#include <Wire.h>
#include "MRT.h"

ClosedCube_Si7051::ClosedCube_Si7051()
{
}

void ClosedCube_Si7051::begin(uint8_t address) {
    _address = address;
    Wire.begin();

    Wire.beginTransmission(_address);
    Wire.write(0xE6);
    Wire.write(0x0);
    Wire.endTransmission();

}

float ClosedCube_Si7051::readT() {
    return readTemperature();
}

float ClosedCube_Si7051::readTemperature() {
    Wire.beginTransmission(_address);
    Wire.write(0xF3);
    Wire.endTransmission();

    delay(15);

    Wire.requestFrom(_address, (uint8_t)2);
    delay(25);
    byte msb = Wire.read();
    byte lsb = Wire.read();

    uint16_t val = msb << 8 | lsb;

    return (175.72*val) / 65536 - 46.85;
}

/*
 * Part 2: Si7051 MECH 45X Team 26 library
 */

bool ClosedCube_Si7051::start_mrt(void) {
    /*
     * Start globe thermometer sensor
     *
     * The code will read a value of 128 or greater
     * if the sensor is broken or disconnected
     *
     * The start sequence returns false (sensor does not work)
     * if a value of 128 is read
     *
     * If the value is less than 128, it returns true
     * (sensor works)
     *
     * The code retrieved from the on line library should be improved
     * to fix this.
     */
    begin(ADDR_MRT);
    delay(500);
    if(run_mrt() > DEFAULT_AVERAGE) {
        Serial.println("Failed to start MRT sensor!");
        return false;
    } else {
        Serial.println("Successfully started MRT sensor!");
        return true;
    }
}

float ClosedCube_Si7051::run_mrt(void) {
    /*
     * Takes globe thermometer measurements until read_count
     * is exceeded.
     * Once read_count is exceeded, the average is taken.
     */
    read_count = 1;

    while(read_count <= MAX_READ_COUNT) {
        T_buf[read_count - 1] = readTemperature();
        Serial.print("Reading #");
        Serial.print(read_count);
        Serial.print(": Tg is: ");
        Serial.println(T_buf[read_count - 1]);
        read_count ++;
        delay(250);
    }
    if(read_count > MAX_READ_COUNT) {
        T_ave = 0;
        for(int k = 0; k < MAX_READ_COUNT; k++) {
            T_ave = T_ave + T_buf[k];
        }
        T_ave = T_ave / MAX_READ_COUNT;
        Serial.print("Average Tg is: ");
        Serial.println(T_ave);
        return(T_ave);
    }
}

// Getter function for Globe Thermometer average value
float ClosedCube_Si7051::get_MRT_ave(void) {return T_ave;}
```