Κώδικας Γεωργία Ακριβείας 1

```
#include "thingProperties.h"
#include <Arduino_MKRIoTCarrier.h>
MKRIoTCarrier carrier;
int moistPin = A5;
String waterPumpState;
String coolingFanState;
String lightState;
uint32_t lightsOn = carrier.leds.Color(82, 118, 115);
uint32_t lightsOff = carrier.leds.Color(0, 0, 0);

void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
  // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
  delay(1500);
  // Defined in thingProperties.h
  initProperties();
  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);
  //Get Cloud Info/errors , 0 (only errors) up to 4
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
  //Wait to get cloud connection to init the carrier
  while (ArduinoCloud.connected() != 1) {
    ArduinoCloud.update();
    delay(500);
  }
  delay(500);
  CARRIER_CASE = false;
  carrier.begin();
  carrier.display.setRotation(0);
  delay(1500);
}
void loop() {
  //Update the Cloud
  ArduinoCloud.update();
  //read temperature and humidity
  temperature = carrier.Env.readTemperature()-8;
  humidity = carrier.Env.readHumidity();
  //read raw moisture value
  int raw_moisture = analogRead(moistPin);
  //map raw moisture to a scale of 0 - 100
  moisture = map(raw_moisture, 0, 1023, 100, 0);
  //read ambient light
  while (!carrier.Light.colorAvailable()) {
    delay(5);
  }
  int none; //We dont need RGB colors
  carrier.Light.readColor(none, none, none, light);
  delay(100);
}
void onWaterpumpChange() {
  if (waterpump == true) {
    carrier.Relay2.open();
    waterPumpState = "PUMP: ON";
  } else {
    carrier.Relay2.close();
    waterPumpState = "PUMP: OFF";
  }
  updateScreen();
}
void onCoolingFanChange() {
  if (cooling_fan == true) {
    carrier.Relay1.open();
    coolingFanState = "FAN: ON";
  } else {
    carrier.Relay1.close();
    coolingFanState = "FAN: OFF";
  }
  updateScreen();
}
void onArtificialLightChange() {
  if (artificial_light == true) {
    carrier.leds.fill(lightsOn, 0, 5);
    carrier.leds.show();
    lightState = "LIGHTS: ON";
  } else {
    carrier.leds.fill(lightsOff, 0, 5);
    carrier.leds.show();
    lightState = "LIGHTS: OFF";
  }
  updateScreen();
}
//Update displayed Info
void updateScreen() {
  carrier.display.fillScreen(ST77XX_BLACK);
  carrier.display.setTextColor(ST77XX_WHITE);
  carrier.display.setTextSize(3);
  carrier.display.setCursor(40, 50);

  carrier.display.print(waterPumpState);

  carrier.display.setCursor(40, 90);

  carrier.display.print(coolingFanState);

  carrier.display.setCursor(40, 130);

  carrier.display.print(lightState);

}
```