Domestic Light
Sensor Assembly and Software build/ flash Guidelines

Last Revision Date: 23 May 2023

Covers Rev F: Green boards labeled Revision F

Pending variants for:
Rev G "Artist Prototype (ed of 50)",
and assembled "Artist Proof (ed of 50)

GITHUB DOCUMENTATION
https://github.com/thirtysevennorth/domesticlight#readme

working repository in process for the Domestic Light Project

//// OVERVIEW OF PROJECT

The primary working file for sensor code is DL_client

Code base: Github Repository for code
https://github.com/thirtysevennorth/domesticlight
Currently John MacCallum, Ian Winters, Weidong Yang, David Coll, and Jeff Lubow are invited.
Currently is set to private, but about to go public as soon as a few wifi and AWS secrets files are cleared out to a private repository.

Issue Tracking - GitHub
Open issues are being logged in the GitHub issue tracker.
Addition background and project documentation is available in the project confluence Confluence site.

Confluence Project Site
Programming: https://domesticlight.atlassian.net/wiki/spaces/DLP/overview

Project Overview
and Admin: https://domesticlight.atlassian.net/wiki/spaces/DLO/overview

Working Client / Branch in GitHub:
The currently working client file and primary project is called DL_client.
Please work on the main branch of the project for any edits. John is working on certain updates to the background OSE processing in a parallel branch and will merge when ready.

The AWS config requires a separate Private Key and Device Certificate for each device. These are indexed by a UUID tied to the MAC address of the boards. Currently that working list of

UUIDs and certificate files for testing are in GitHub but need to be moved out before making public and also before adding the production run of boards.

Files are intended to be built using the Arduino IDE v2.10 or later

PDF LINK TO BUILD GUIDE BELOW:
https://github.com/thirtysevennorth/domesticlight/blob/main/Domestic%20Light%20Sensor%20Assembly%20Guidelines.pdf

// REQUIRED IDE BOARD FILES, LIBRARIES, DEPENDENCIES ////////////////////////////////////

CONFIRM YOU HAVE ESP32S3 BOARD INSTALLED If you have not built a project in Arduino 2.x for ESP32-S3 on your machine please install the following in the Arduino IDE

1. In the Arduino IDE > Preferences > Additional Boards Manager URL add the URL for the Espressif Arduino ESP32 boards https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json
2. In Boards Manager add "esp32" by Espressif Systems. This should be on V 2.0.9 or later.

REQUIRES Arduino IDE 2.10 with the following Libraries added : From Arduino Library Manager install the following libraries:

1. RTClib
2. Adafruit_AS7341
3. pubSubClient by Nick O'leary http://knolleary.net
4. ArduinoJson https://arduinojson.org

Install the following libraries manually:

- MicroOSCcript. To install clone https://github.com/0Z3/MicrOSCript and add the library to your Arduino Libraries folder. Be sure the clone is recursive and includes libose, o.se.oscript, and o.se.stdlib if you clone from the web.

The following libraries are part of the esp32 board file and shoudl be able to be called without additional install once the board is installed.

- Arduino
- Time (esp32 board version)
- WebServer
- WiFiCLient (esp32 board version)
- WiFiClientSecure (esp32 board version)
- WifiUDP (esp32 board version)

- Wifi (esp32 board version)
- Wire (esp32 board version)

For the UM Feather S3 build add

- esp32s3-arduino-helper by Unexpected Maker

//// END REQUIRED BOARD FILES, LIBRARIES, DEPENDENCIES

//// PROJECT HARDWARE: The project requires the following hardware components to build
PROCESSOR - ESP32-S3
There are two processor types:
* Unexpected Maker ESP32-S3 Feather
* Adafruit ESP-32 S3 Feather - either w/ or W/o PSRAM

COMPONENTS
* Domestic Light circuit board - Rev F (preliminary prototype board, green solder mask), Rev G (Artist Prototype - no SMT assembly, white soldermask), Rev H (artist proof - SMT assembled, white solder mask)
* For Rev F and Rev G assembly w no SMD assembly
    * 2x tactile right angle button switches, E Switch TL1100CF160Q for switch S1 (adhoc mode), switch S2 (RST)
    * Adafruit AS7341 light sensor board https://learn.adafruit.com/adafruit-as7341-10-channel-light-color-sensor-breakout/pinouts
    * Adafruit DS3231 RTC board https://learn.adafruit.com/adafruit-ds3231-precision-rtc-breakout/pinouts
    * CR1220 coin cell
    * 3x 10k axial resistors OR adding same as SMD 0603 components on pads
    // For Rev F - add 2.2 or 4.7 ohm resistor between pin RST and S2 pin and a little bit of heat shrink or electrical tape
    * clear slim case for RPI 3 or 4. using https://www.amazon.com/Vilros-Transparent-Compact-Raspberry-Options/dp/B07VF6LCCL
  * USB-C cable to connect to 5v USB power and / or computer

//// END HARDWARE AND SOFTWARE LIST ////////////////////////////////////

// SOFTWARE SETUP AND CONFIG: //

//// NOTE ON AWS keys, certs and THINGNAME / UUID ////// These are assigned by admin and unique to each device / UUID. in the private repository they are saved by UUID which is keyed to the MAC ID

/// INTITIAL SETUP AND FLASHING
* Connect board via USB-C connector on ESP32 to your computer running the Arduino IDE with the DL_Client.ino file open.
* Edit secrets.h to include:
  * your wifi info for testing and flashing – be sure to remove when done before shipping.
  * your UUID //// UUID NOTE this is our internal six digit serial - ex. 100001.
    //// Series 1xxxxx are "prototypes" in general.
    //// Series 10000x are testing boards,
    //// Series 1xx2xx are the "artist prototypes" or first publicly distributed
    //// Series 2xxxxx are the "artist proof" series or the first SMT assembled series. These include hardware support to attach a VOC and Methane sensor.
    //// Series 3xxxxx are reserved for the future production run of sensors. These will include a VOC and Methane sensor.
  * AWS private key, //matching your UUID / THINGNAME
  * device cert to match your assigned UUID.
  * device private key

/// FIRST TIME FLASH // to correctly flash you must first put the board in to "boot" or reflash mode.

- press and hold "boot" button on the board
- while holding boot button press and release the "RST" button on the board or the RST button (S2) if correctly connected.
- in the IDE Tools> Port select the new serial port that appears.
- In the IDE Tools> Board select the correct board (UM Feather S3 or Adafruit Feather S3 no PSRAM)
- Build and flash the file to the board.
- after the file is fully uploaded and complete, press the RST button to reset the board. If the serial monitor doesn't appear try connecting / reconnecting power.
- hold down S1 (the adhoc mode switch) or connect pin IO17 to gnd, and press the reset button. The Serial monitor should give you instructions to join the "Domestic Light" adhoc network and go to 192.168.4.1 to enter the local wifi info. Release the adhoc mode switch.
- Connect to the adhoc mode wifi "domestic light" and go to the IP
- Enter the local wifi SSID and password, the IP address you want to send OSC to (if desired), and copy the MAC ID shown on the browser into the MAC ID section. Hit submit. The wifi will disconnect. Press the RST button if the board doesn't automatically reboot.

* at this point the board shoudl be fully configured. Watch the serial monitor for output. you should see something like this:

//////////// START OF SAMPLE SERIAL MONITOR OUTPUT
14:37:27.380 -> done
14:37:27.380 -> IP address: 192.168.1.189 // the boards IP address

14:37:27.380 -> Listening for OSC on UDP port 15000 // the UDP port the board is listening on
14:37:27.380 -> Connecting to AWS IOT // AWS config info
14:37:29.447 -> AWS IoT Connected! - subscribe and publish topics
14:37:29.447 -> DomesticLight/100004/sub // these topics should reflect the boards UUID which should be tied to the MAC ID
14:37:29.447 -> DomesticLight/100004/data
14:37:29.447 -> $aws/things/100004/shadow/update
14:37:29.447 -> $aws/things/100004/shadow/get
14:37:29.447 -> Initializing the RTC...done
14:37:29.447 -> Setting the system clock...done
14:37:32.012 -> Current Time, Date and Temp set to current date and time
14:37:32.012 -> 2023/5/20 (Saturday) 13:37:32 /// THIS should reflect reality
14:37:32.012 -> since midnight 1/1/1970 = 1684589852s = 19497d
14:37:32.012 -> Temperature: 26.25 C
14:37:32.012 -> Starting the color sensor...done.
14:37:32.143 -> Sent to data and shadow 203 bytes: // these data points are what is going to AWS every x seconds. The timestamps should be ~ 1s apart. If not there is probably a missing pullup resistor on SQW. This will shortly crash the board and AWS connection.
14:37:32.143 ->
{"MAC_ID":"70:04:1D:AD:D2:54","F1_415":16,"F2_445":22,"F3_480":21,"F4_515":45,"A5_VIS1":247,"A6_NIR1":303,"F5_555":30,"F6_590":33,"F7_630":36,"F8_680":50,"F9_VIS":244,"F10_NIR":296,"unixtime":1684589852}
14:37:33.135 -> Sent to data and shadow 203 bytes:
14:37:33.135 ->
{"MAC_ID":"70:04:1D:AD:D2:54","F1_415":14,"F2_445":14,"F3_480":16,"F4_515":23,"A5_VIS1":203,"A6_NIR1":250,"F5_555":25,"F6_590":28,"F7_630":29,"F8_680":41,"F9_VIS":202,"F10_NIR":250,"unixtime":1684589853}
////// END SAMPLE SERIAL MONITOR OUTPUT

/// PHYSICAL and SOFTWARE TESTING SUGGESTIONS ////

1) Verify that S1 and S2 pull the respective pins low on pressing.
2) Verify that a 10k Resistor is soldered between DS3231 pin SQW and VCC
3) Verify each header pin soldering is good - no shorts or bridges from soldering and that all solder joints are good.
4) When plugged into power and flashed LED's on AS7341, DS3231 and ESP 32 should all light up.
5) If the board disconnects from serial after reboot there is likely a config issue with AWS. The AWS MQTT library is prone to crashing the USB serial connection if it encounters any errors.
6) Each board has a unique MACID, UUID / THINGNAME, and AWS certs. If the device cert / private key do not match the UUID / assigned serial number AWS will block the connection as a security violation.
7) ultimately test the board using the AWS MQTT test client - instructions and logins pending. In the meantime - if it prints to console its probably getting to AWS.

PHYSICAL SETUP GUIDANCE FOR REV F:

The REV F has a few quirks that need some TLC to make work:

It consists of the

1) Domestic Light - REV F board
2) Unexpected Maker ESP32-S3 Feather Board
3) Adafruit DS3231 RTC board
4) CR1220 Battery
5) Adafruit AS7341 light sensor board
6) Clear Case
7) 2 right angle buttons for exterior reset and adhoc boot mode control

The REV F fixes consist of

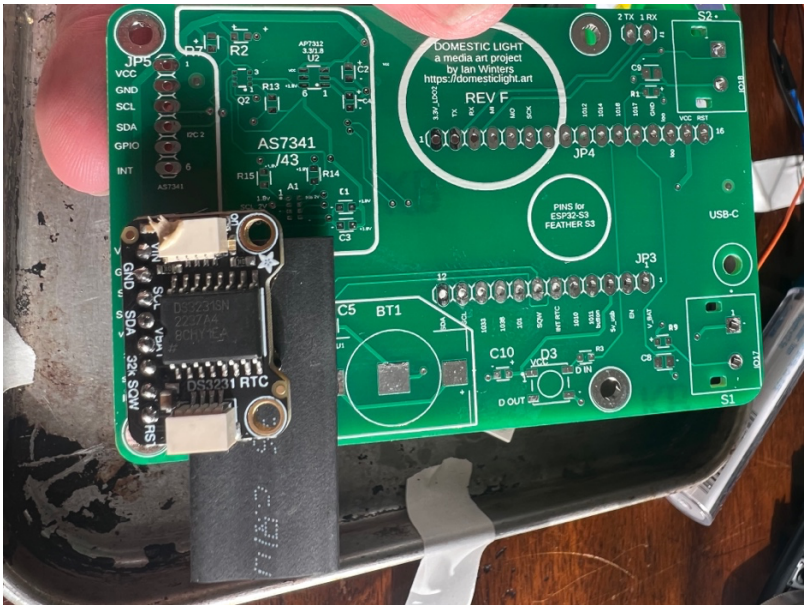
8) 3 10k axial resistors
9) 1 ~ 2.2ohm or 4.7 ohm axial resistor
10) A few pieces of heat shrink

The physical assembly steps:

1) Solder the header pins to the ESP32-S3, the DS3231 RTC, and the AS7341 light sensor boards. Short end of pin to the board, long end down.

Begin attaching the secondary boards to the PCB

2) Attach the DS3231 RTC board as shown below with a piece of heat shrink under the battery contact. This is to prevent the battery contact from shorting the main board via the pads for assembly below it. Be sure pins match the labeling on the circuit board. Be aware that the clearance between the DS3231 and the AS7341 boards is ~ .5mm



3) Once the DS3231 is soldered in place from below you are going to get ready to connect a 10k axial resistor between the underside of the VCC pin of the RTC and the SQW pin. See below indicating the VCC and SQW pin.

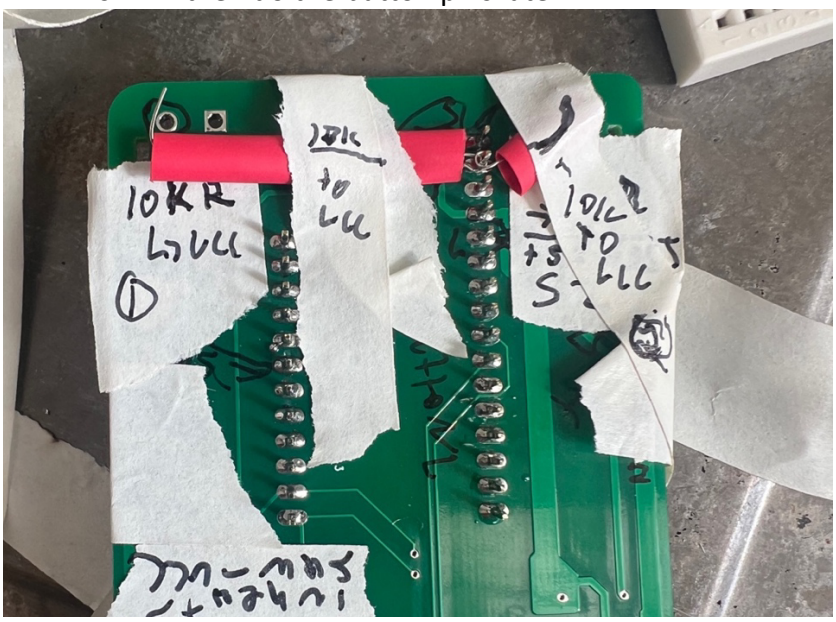
3 B – Finished view

You'll want to trim the resistor to size and loop the wire over the pins. You'll also want to slide. A piece of heat shrink over it to prevent shorting something out in the case. The heat from the soldering iron will take care of the shrink.

4) The next step is soldering on the AS 7341 board in the JP5 / AS7341 section of pins. No special tricks needed.

5) Then solder on a 10k resistor between the switch pin for the button S1 and VCC and between S2 and VCC.

   The resistors go from the VCC pin to the switch pin for switch 1 and switch 2. These are labeled 1017 and 1018 on the top of the board. They are the pins that are NOT connected to ground. I found it was easiest to solder them both the VCC pin in heat shrink then do the button pins later.



6) Next solder the two buttons at S1 and S2. They have little plastic tabs that snap into the slots and go in nicely. Once done move on the tricky part.

7) Solder a very low resistance resistor between the RST pin (pin 1) and the switch pin on S1 / 1017. I used a 4.7 ohm resistor. This compensates for a board error that had the reset button going to IO17 instead of RST.

8) Slide a battery into the RTC (+ side is up)
9) **UUID and labeling:** Issue the board a unique user ID / aka UUID . Using the label maker, make 2 labels using ½" tape (in the label maker) that have the UUID on it.
   For ex: 100010**.**
   Record the UUID on the tracking sheet. It needs to be associated with a MAC ID, and the UUID must match the AWS Certificate.
   Put one of the UUID label on the underside of the board in the clear space under the battery holder. Save the second to go on the bottom of packing box.

For the artist proof and prototype edition using the marker indicate the number on the edition line (on the underside of the board) using a black marker.

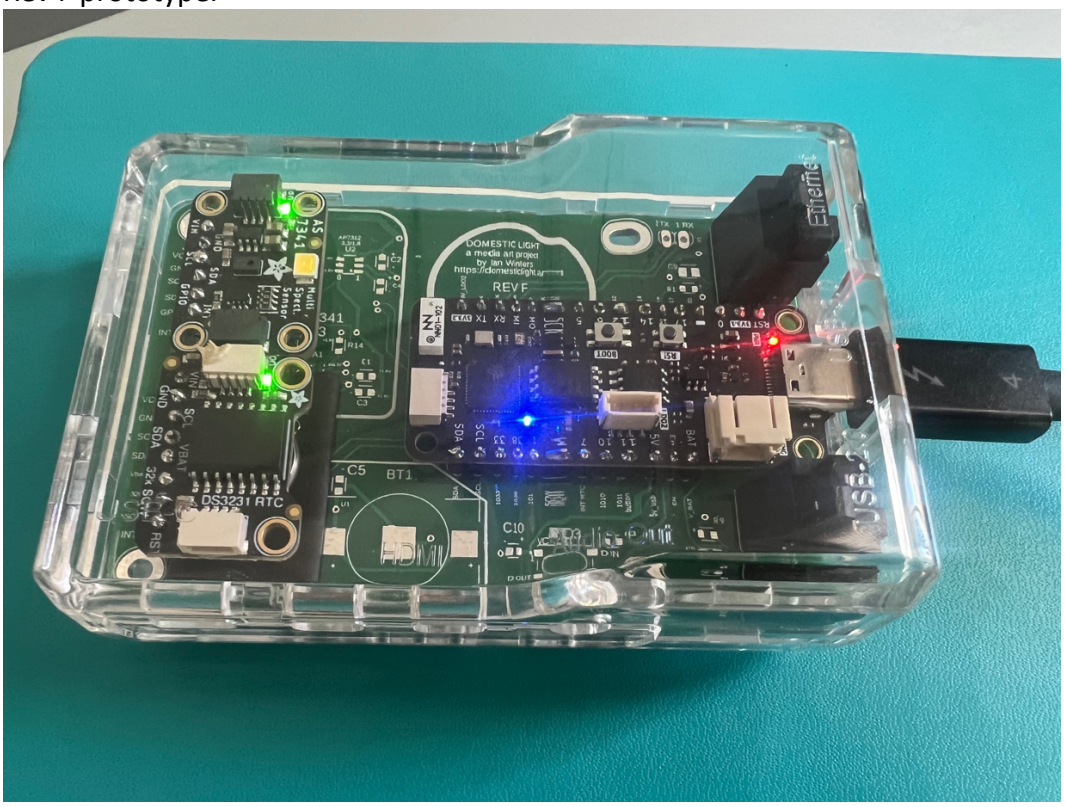
FYI - The series number schema:
//// Series 1000xx are testing boards, using boards labeled REV F that are green boards
//// Series 1xx2xx are the "artist prototypes" or first publicly distributed and use the white circuit "artist prototype, ed of 50"
//// Series 2xxxxx are the "artist proof" series or the first SMT assembled series. These include hardware support to attach a VOC and Methane sensor
//// Series 3xxxxx are reserved for the future production run of sensors. These will include a VOC and Methane sensor

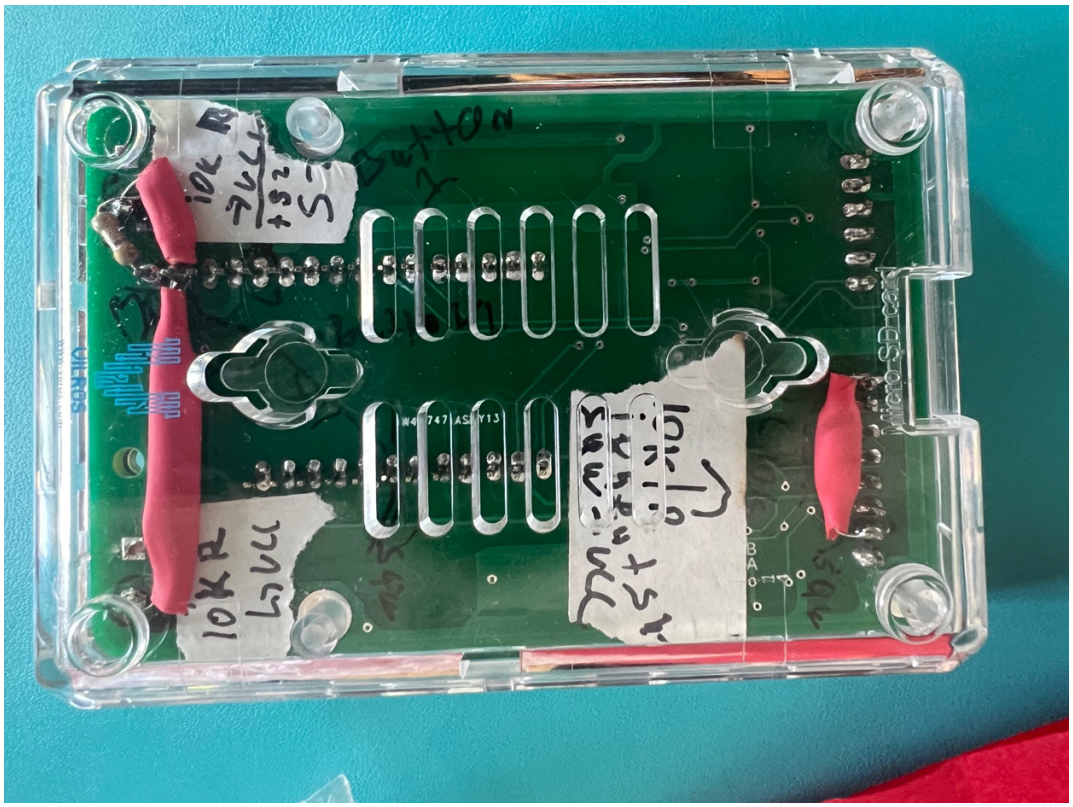
10) Place the completed board into the clear case base.

11) Use 4 2.5mm x 4mm stainless screws to secure the board to the case.

YOU ARE NOW READY TO FLASH THE BOARD – Go back to the steps SOFTWARE SETUP AND CONFIG.

12) Once flashed snap the lid – here's a finished view ready to be boxed up of the very first Rev F prototype.




10K R
Button
10K R
MicroSDboard