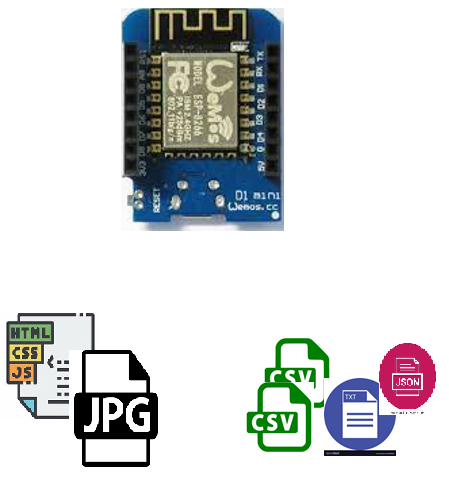
File management

Dalam pembuatan software ini, ESP-8266 terbagi dalam beberapa kelompok penyimpana file :

* File program 2 MB
* Local storage LittleFS 2 MB
  + Konfiguration – Sensor, User
  + Logsheet – data trending, hourly average selama 1 minggu (7 hari), daily average selama 1 tahun
  + Web file – HTML, CSS, JS dan Images



File: /modal-dialog.js

File: /report.html

File: /report.js

File: /style.css

File: /widgetsClass.js

**Log-sheet**

**root**

File: /authentication.js

File: /config.html

File: /config.js

File: /gmf.png

File: /index.html

File: /index.js

File: /login.html

File: /login.js

File: /logoGMF.png

File: /logsheet/Friday\_ls.csv

File: /logsheet/Monday\_ls.csv

File: /logsheet/Saturday\_ls.csv

File: /logsheet/Sunday\_ls.csv

File: /logsheet/Thursday\_ls.csv

File: /logsheet/Tuesday\_ls.csv

File: /logsheet/Wednesday\_ls.csv

File: /logsheet/sensors.cfg

File: /users.cfg

Software Structure

1. Logsheet (data gathering, collection and persistenace from sensor)
   * Get sensor data complete with filtering (alfa ema)
   * Display to OLED
   * Averaging – minute, hourly and daily
   * Saving to local storage (LittleFS) hourly base on day of week, daily base on year
   * Data Services for web
     + Sensor configuration
     + Logsheet hourly
     + Trending data (temperature and humidity)
2. Web services :
   * Server side (back end)
     + Static files services (Images, CSS and JS)
     + HTML
     + AJAX
   * Client side (Front end) pages / User Interface
     + Home
     + Report
     + Configuration
     + Login
     + Logout
     + About
3. Data Model
   * Sensor parameters (High, Low, Unit, Alarm, alfa ema)
   * Loghsheet (temperature and humidity)
   * User (username, password, email, level)
4. Time keeping for synchronizing sampling
5. Start up
   * Display logo during start up
   * Display welcome messages

Adapun fungsi tersebut di atas, dibagi menjadi beberapa bagian software dengan memisahkan sesuai dengan fungsinya (OOP principles) sebagai berikut :

* dataLogger.ino, datalogger.h

Sebagai program utama berikut sebagai pintu masuk (entry point) yang di dalamnya terdapat tulang punggung (skeleton) program Arduino – ESP :

dataLogger.h berisi seluruh definisi global variables dan konstanta dalam berikut fungsi yang digunakan dalam keseluruhan program ESP-8266 ini.

* Importing libraries
* Global variable and objects declaration
* Setup

Dalam function setup ini, ada beberapa hal yang dikerjakan sbb:

* Init OLED 48x64 pixel dengan I2C
* Display logo GMF
* Display Welcome dan program step
* Init pin mode
* Init serial communication dengan baud rate 11520 bit/sec
* Init local storage (LittleFS) untuk server files, configuration dan logsheet
* List files dalam LittleFS
* Load User data (engineer dan operator)
* Setup default user, active user = guest
* Init sensor DHT11 untuk temperature dan humidity dengan one wire communication
* Setup logsheet meliputi :
  + Attach parameter temperature dan humidity
  + Attach sensor DHT11
  + Attach display OLED
  + Set time
* Setup urlController meliputi :
  + Static url (meliputi image files, CSS, JS)
  + HTTP url (meliputi HTML files)
  + AJAX url (data files)
* Start WIFI (MODE\_AP atau Multifiwi)
* Start server
* Loop

Function loop ini yang dijalankan (execution) secara terus-menerus selama ESP8266 ini jalan. Adapun tugas/function yang dijalankan adalah :

* Execute logsheet serta memberikan nilai sampling time
* Execute mainSequence
* Memberikan seting time setiap 1 menit ke logsheet

Adapun detail dari masing-masing fungsi di atas dijelaskan dalam sub bab terpisah.

* logsheet.h, logsheet.cpp

Sebagaimana dijelaskan sepintas dalam Software structure point i. di atas, logSheet dengan entry point function execute, mempunyai tugas yang dijalankan sesuai dengan periode yang berikan pada nilai samplingTime. Nilai ini di-setting pada dataLogger.ino (function loop). Logsheet.h mendefinisikan class Logsheet yang selanjutnya dalam program utama (dataLogger.ino) digunakan untuk membuat object dengan nama logsheet(“logsheet”). Selanjutnya object ini (logsheet) digunakan untuk operasi selanjutnya. Dalam class Logsheet ada beberapa fungsi/method utama yang mendukungnya.

* Function execute()

Fungsi ini sebagai titik masuk (entry point) pada class Logsheet (yang selanjutnya digunakan untuk mendefinisakan object logsheet) dengan beberapa tugas sebagai berikut :

* + Melakukan sinkronisasi waktu antara dataLogger.ino (program utama).
  + Melakukan perhitungan jumlah sampling data per menit, dan membatasi 60 sampling per menit.
  + Melakukan beberapa kegiatan saat sampling time telah sesuai (tiba):
    - Mengambil data dari sensor (\_getsensorValue())

Fungsi ini akan mengambil data dari DHT11 berupa data temperature dan humidity serta melakukan filtering pada data yang diambil berdasarkan parameter alfa ema (default 80%). Pengambilan data DHT11 ini dilakukan ketika tidak pada mode SIMULATION. Saat mode SIMULATION data diambil dari random data.

Data temperature dan humidity selanjutnya disimpan dalam data parameter glogal yang di-attach ke logsheet object. Parameter ini baik untuk temperature dan humidity.

Selanjuntya, data filtering ini juga dibandingkan/dikomparasi dengan data konfigurasi untuk masing-masing konfigurasi sensor, baik untuk alarm high, low serta memberikan status untuk alarm yang dimaksud.

* + - Menampilkan data pengukuran pada OLED 64x48 pixel, lengkap dengan unit untuk Temperature dan Humidity.
    - Melakukan perhitungan event/kejadian mulai dari samplingTrending, samplingSecond, samplingMinute, samplingHour
    - Melakukan kegiatan logsheet sesuai dengan event yang terjadi seperti : perhitungan averaging, penyimpanan data ke dalam local storage (LittleFS) sesuai dengan file ynag dimaksud. Penyimpanan file dengan type text dengan format CSV (Coma Seprated Value).
* Function pendukung WEB
  + Mengambil data rata-rata jam (hourlyAverage) selanjutnya dikirim ke WEB untuk ditampilkan sebagai data harian dalam setiap minggunya (Monday – Sunday).
  + Mengambil data trending per sampling, data ini selanjutnya dikirim ke WEB untuk graphic trending temperature dan humidity (24 sampling data).
  + Mengambil data konfigurasi parameter sensor (range high/low, alarm high/low, unit) selanjutnya dikirim ke WEB untuk mebuat graphic dial baik temperature dan humidity.
  + Mengambil data pengukuran per-sampling untuk menampilkan nilai dalam graphic pada WEB.

Data yang diambil tersebut di atas disimpan dalam local storage (LittleFS) kecuali data pengukuran per-sampling.

* Function private untuk keperluan internal loghseet, fungsi ini hanya bisa digunakan dalam lingkup class logsheet sendiri untuk mendukung kegiatan yang diperlukan meliputi :
  + Setup default parameter – untuk setup default saat pengembangan software
  + Setup parameter sensor dari file berupa data parameter (range high/low, alarm high/low, unit).
  + Init random JSON – untuk setup JSON saat pengembangan software
  + Fungsi penyimpanan dan pembacaan file ke dan dari local storage (LittleFS)
* model.h, model.cpp

Secara umum dijelaskan pada Software structure point iii di atas meliputi :

* + Sensor parameters (High, Low, Unit, Alarm, alfa ema)
  + Loghsheet (temperature and humidity)
  + User (username, password, email, level)

Data model ini selanjutnya digunakan untuk penyimpanan data secara global untuk operasional program. Data model yang dimaksud dibentuk dalam class ataupun data struktur biasa. Data model yang dibentuk dalam class juga dilengkapi dengan API (Aplication Program Interfacing) untuk mempermudah penggunaanya.

Class AccesParameter adalah class yang didedikasikan untuk akses parameter sensor dengan API sbb:

* Init
* getJSON – mengambil parameter dalam format JSON
* getOperation – mengambil data operasi untuk temperature dan humidity
* getParameter – mengambil data dalam bentuk struktur
* setParameter – seting parameter dalam bentuk struktur
* sequenceTimer.h, sequenceTimer.cpp

Sebagaimana dala Software structure point iv, class ini mengatur waktu utamanya setiap menit. Jika durasi satu menit telah tercapai, akan memberikan peringatan berupa event/kejadian sehingga dapat digunakan untuk proses eksekusi kegiatan yang ada pada object logsheet (dari class Logsheet) seperti dijelaskan dalam paparan logsheet di atas.

* start\_up.h

Secara garis besar dijelaskan dalam sub bab Software structure point v, StartUp merupakan Class (yang selanjutnya menadi object startUp) yang bertugas untuk :

* Menampilkan logo Garuda, seperti pada gambar berikut.

Gambar ini merupakan konversi dari file NPG menjadi bitmap dengan ukuran pixel 64x48 sesuai dengan ukuran OLED yang dipakai. Selanjutnya hasil konversi dijadikan konstanta dengan nama logo\_gmf. Untuk menampilkan logo ini menggunakan fungsi logoDisplay() yang ada pada class tersebut.

* Menampilkan welcome, seperti pada urutan gambar berikut.

Pesan welcome ditampilkan secara berurutan oleh fungsi welcomeDisplay(). Pesan welcome mulai dari GMF AeroAsia, lalu IoT DataLogger lengkap dengan versinya dan diakhiri dengan pesan Salman Alfarisi serta bulan pembuatan software ini.





* Menampilkan Step start up.

Untuk mengetahui urutan sebagaimana dijelaskan pada fungsi setup di atas (lihat penjelasan datalogger.h dan datalogger.ino pada sub bab setup) diberikan display pada OLED sehingga tiap Langkah/step start up bisa diketahui. Step mulai dari 00 yaitu pada saat init OLED 64x48 pixel berikut I2C. seperti pada gambar di samping dan dilanjutkan pada urutan yang ada pada fungsi setup.

Pada STEP 05, IoT DataLogger berupaya untuk connect dengan WIFI yang telah dikonfigurasi pada datalogger.h. Jika koneksi berhasil, maka step dilanjutkan sampai dengan step 09.



Step 09 adalah dimulainya aktivasi server sehingga memungkinkan IoT DataLogger bisa diakses via internet/WEB.

Selanjutnya IoT DataLogger bertugas untuk melakukan pengambilan data dari sensor DHT11, menampilkan nilai pada OLED 64x48 pixel serta tugas yang lain seperti dijelaskan dalam logsheet.h dan logsheet.cpp di atas.

Selain dari itu, IoT DataLogger juga siap sebai server untuk menangani permintaan via internet/WEB. Penjelasan terkait dengan ini dipaparkan dalam sub ba tersendiri. Fungsi yang menangani penampilan step ini adalah stepDisplay() dengan memberikan parameter step yang sedang berjalan padanya.

Flowchart – datalogger

Start

Import required libraries

Declare Variables & Objects

Protype functions

Display.begin() – init OLED I2C for 64x48 pixel

startUp.AttachDisplay() – attach OLED to startUp

startUp.logoDisplay – display logo to OLED

startUp.welcomeDisplay – display welcome messages

setup pin for LED

Serial.begin(11520) – baud rate 11520 bit/sec

LittleFS.begin() – init little FS (local storage ESP8266)

listAllFilesInDir() – list file in LittleFS

loadUser() – for engineer and operator

setupDefaultUser() – active user is guest

dhtSensor.begin() – init DHT11 sensor

logsheet.AttachParameter() – temperature & humidity

logsheet.AttachSensor() – sensor DHT11

logsheet.AttachDisplay() – OLED

logsheet.AttachLed() – LED

logsheet.setTime() -synchronize

N

Y

StartWIFI\_AP(); - WIFI AP init

startMultiWifi(); - Multi WIFI init

WIFI\_AP

urlController() – WEB handling (req – resp)

server.begin() – start server

1

1

isMinute

logSheet.setTime() – set time from NTP

Y

N

logSheet.execute() – per sampling time

mainSequence.execute()

End

Logsheet.execute()

Start

\_samplingTime = samplingTime

\_synchronized

\_samplingSec = 0

\_synchronized = true

End

