

NOT: Bu belge sade teknik mimari özetidir; kod içermez.

1. Genel Mimari

Sistem üç ana katmandan oluşur: 1. **Core** – bina, brülör, ısı algoritması 2. **I/O** – röleler, RGB, buzzer, sensör arabirimleri 3. **Tools** – testler, syslog üreticisi

Her bileşen modülerdir ve config_statics + config_runtime yapılarını kullanır.

2. Config Yapısı

2.1. config_statics.py

- BUILDING_NAME / LABEL
- GEO bilgileri
- Röle tanımları
- BURNER_GROUPS
- SENSOR statik alanları (serial, flat_no, floor...)

Bu dosya değişirse uygulama yeniden yüklenmelidir.

2.2. config_runtime.py

- BUILD_BURNER (B/F)
- BUILDING_SETPOINT_C
- BUILDING_HEAT_TOLERANCE
- OUTSIDE_SMOOTH_SECONDS
- Kullanıcı tarafından değiştirilebilir değerler

Değişiklikler **ConfigReloader** ile canlı uygulanır.

3. Building (core.building)

Bina sınıfı: - FLAT_AREA listesini yükler - DS18B20 sensörlerini okur - Kat ve yön bazlı sıcaklık istatistikleri üretir

effective_mode hesaplanır: - config_mode = B - bina verisi varsa → effective = B - bina verisi yoksa → effective = F

Syslog çıktısında 3 satır bina ısısı:

[effective] Avg: xx Min: xx Max: xx

4. Sensor Yapısı

4.1. DS18B20

- Daire / oda ısıları
- Bina içi istatistiklerin temel kaynağıdır.

4.2. BuildingEnvironment

- DHT11 / DHT22 → dış ısı + nem
- LDR → ışık seviyesi
- Rain sensor → yağmur/nem

Bu sınıf used_outside hesaplamasına veri sağlar.

5. Röle Sürücüsü (io.relay_driver)

Röle kontrolü iki katmandan oluşur:

5.1. Tekil Röleler

- igniter
- circulation_a
- circulation_b
- gas_sensor
- pressure_sensor

5.2. BURNER_GROUPS

Örnek yapı:

```
BURNER_GROUPS = {
  0: {
    "name": "MainBurner",
    "igniter": "burner_1_igniter",
    "circulation": ["burner_1_circ_1", "burner_1_circ_2"],
    "circ_flags": {"burner_1_circ_1":1, "burner_1_circ_2":0},
    "location": "sol bina"
  }
}
```

Test aracı: tools/relay_test.py

6. Brülör Kontrol (core.burner_controller)

İki çalışma modu vardır:

6.1. F Modu (Dış ısı odaklı)

Adımlar: 1. used_out_heat hesaplanır: - dış ısı - hafta sonu / tatil artışı - comfort_offset çıkarma 2. Saatlik tasarruf yüzdesi ile default_max kırılır 3. BURNER_FIRE_SETPPOINT_MAP → hedef fire alınır 4. fire_sp = min(map_setpoint, adjusted_max) 5. DELTA_T_MAP_HOURLY uygulanır 6. Heat decision → want_heat

6.2. B Modu (Bina içi ısı odaklı)

1. Referans ısı seçimi:
2. BUILD_EXIT_HEAT (avg/top/bottom/min)
3. BUILDING_SETPPOINT_C ± tolerance bandı kontrol edilir
4. Fire_sp kademeli olarak BUILDING_INCREMENT_C ile ayarlanır
5. Delta T uygulanır
6. want_heat hesaplanır

6.3. B → F Fallback

B modunda bina sensörleri okunamıyorsa: - effective = F - fire hesaplaması F modunda yapılır Sensörler düzelince tekrar B'ye döner.

Syslog etiketi:

```
[effective=B]
[effective=F]
```

7. Legacy Syslog (tools.legacy_syslog)

Çıktı formatı klasik Rasp2 log formatıdır. Her turda: 1. Versiyon ve tarih 2. Gün doğumu / batımı 3. Tatil bilgisi 4. Bina ısı satırları ([effective]) 5. Dış ısı, çıkış ısı, dönüş ısıları 6. used_outside değeri 7. want_heat → brülör / pompalar durumu

Kesin format örnekleri B/F moduna göre değişir.

8. Test Araçları

- relay_test.py → tüm rölelerin çalıştığını doğrular
- env_test (gelecek) → dış ortam sensör testi
- fire_calc_test → F/B algoritma testleri

9. Dosya Ağacı (özet)

```
ebuild/  
├─ core/  
│   ├── building.py  
│   ├── burner_controller.py  
│   ├── heat_engine.py (öneri)  
│   └─ environment.py  
│  
├─ io/  
│   ├── relay_driver.py  
│   ├── rgb_driver.py  
│   ├── buzzer_driver.py  
│   └─ sensor_ds18b20.py  
│  
├─ tools/  
│   ├── relay_test.py  
│   └─ legacy_syslog.py  
│  
├─ config_statics.py  
├─ config_runtime.py  
└─ __main__.py
```

10. Özet

Bu doküman, Abrulor sisteminin çekirdek mimarisini, modüller arası ilişkileri, bina ve brülör kontrol modellerini, sensör yapılarını ve syslog akışını açıklar. Koda bağımlı değildir; isimler ve tasarım ilkeleri üzerinden anlaşılabilir bir topoloji sunar.