

ReCUM12 - Sprint-1 Teknik Dökümanı

1. Sprint'in Genel Tanımı

Sprint Adı: ReCUM12 – RS485 Pump Controller Yeniden Yazım Sprint-1

Amaç:

- Eski ReCUM10 C++ uygulamasını tamamen terk etmek
- Aynı file-tree ve dosya isimleri korunarak Python pump controller mantığını yeni ve modüler C++/CMake projesine taşımak
- ReCUM12 projesi için sağlam, genişletilebilir mimari oluşturmak
- Modüller ayırtırmak (hw / core / gui / rfid / net / utils)
- Build, debug, log, versioning, GitHub, VS Code ortamını kurmak
- Şimdilik sadece: RS485 Pump Protocol → C++ port'u için hazırlık

2. Başlangıç Durumu

2.1. Eski C++ uygulaması

- Eski ReCUM10 uygulaması mantık olarak ÇÖP kabul edildi.
- Saha uyumluluğu gerekmıyor (eski uygulama yayında değil).
- Sadece file-tree, dosya isimleri, glade GUI yapısı referans olarak tutuluyor.

2.2. Python Pump Controller

- Python win_pump_ctrl_rs485.py dosyası sprint için tek gerçek davranış kaynağı seçildi.
- İçerdiği tüm mantık C++ tarafında modüllere dağıtılmak.

2.3. Yeni Proje Adı

- Proje adı: ReCUM12
- Uygulama hedefi (binary): recum12_app
- Sürüm numarası: v12.01.100

3. Sprint İçinde Karar Alınan Mimari

3.1. Modül yapısı (modules/ altında)

modules/hw – RS485 pompa haberleşmesi ve protokol

- PumpR07Protocol
 - Python protokol mantığının birebir C++ karşılığı
 - CRC, BCD helpers
 - Frame oluşturma
 - Frame ayırtırma
 - Callback tetikleyen olay modeli (status, fill, totals, nozzle)
- PumpInterfaceLvl3
 - RS485 transport
 - PumpR07Protocol ile üst katman arasındaki köprü
 - Event yönlendirmesi

modules/core – İş mantığı

- PumpSaleTracker
 - Fill record işleme
 - Sayaç yönetimi
 - Persist işlemleri
- UserManager
 - Kullanıcı listesi
 - Limit/policy kontrolü
 - (İleride RFID ile entegre olacak)

modules/gui – Arayüz

- MainWindow
 - Tüm GTK arayüzün kurulduğu yer
 - Butonlar → core/hw katmanlarına bağlanır
 - Callback'ler → GUI update

- rs485_gui_adapter
 - Pump state → ikon/metin/renk mapping
 - Python'daki _state, _hint_state_intent, nozzle icon, LED logic buraya taşınacak

modules/rfid

- Pn532Reader
 - Sonraki fazda kart okuyucu desteği
 - Event: on_card(uid)

modules/net

- RemoteProto
 - Uzak bilgisayar protokolü
 - Pump/total/log aktarımı
 - Remote komut alma
- BlynkClient
 - Mobil Blynk uygulamasına bilgi aktarma
 - Mobilden komut alma

modules/utils

- LogManager
 - logs/log_* klasörlerine log yazma
 - İleride log-forwarding entegrasyonu
- Settings
 - Konfigürasyon
- Version
 - APP_NAME, APP_VERSION (v12.01.100)

4. Sprint İçinde Yapılan Teknik Çalışmalar

4.1. Yeni ReCUM12 Dosya Ağacı

- apps/recum12_app
- modules/hw, core, gui, rfid, net, utils
- logs/log_user, logs/log_runtime, logs/log_pump, logs/log_remote
- backups, patches, docs, scripts, tools/win_pump_ctrl_rs485.py

4.2. CMake Altyapısı

- Root CMakeLists.txt alt modüllerini ekliyor
- Her modül kendi CMakeLists ile library üretiyor
- recum12_app executable'ı tüm modüllerini linkliyor
- C++17 etkin, build sistemi stabil

4.3. VS Code Yapılandırması

- Remote-SSH ile RasPi'ye bağlanma
- CMake Tools kit seçimi (arm-linux-gnueabihf-g++)
- cmake.buildDirectory = \${workspaceFolder}/build
- cmake.sourceDirectory = \${workspaceFolder}
- launch.json ile gdb üzerinden recum12_app debug edilebiliyor

4.4. GitHub Entegrasyonu

- Local repo → GitHub (sertumit/ReCUM12.git)
- Branch: main
- .gitignore hazırlandı
- patches ve docs altına .gitkeep eklendi (isteğe bağlı)

5. Python Kodundan Taşınacak Yapılar (Resmi Lista)

5.1. Protokol Sabitleri

- STX, ETX, TRAIL
- R07 komut kodları (CD1, CD3, CD101...)

- Frame header/body/CRC formatı

5.2. Helper'lar

- `crc16_ibm()`
- `_bcd4_to_int()`
- `_bcd5_to_int()`
- `_int_to_bcd4()`

5.3. Gönderim Fonksiyonları

- `_send_min_poll()`
- `_send_min_ack()`
- `_send_cd1()`
- `_send_cd3_preset_volume()`
- `_send_cd101_total_counters()`

5.4. Parsing Fonksiyonları

- `_parse_and_update()`
- `_update_dc_from_payload()`
- Fill record decode
- Total counters decode
- Status decode
- Nozzle event decode

5.5. State Machine Mantığı

- `_state`
- `_hint_state_intent`
- UI LED/renk durumu
- Status label metinleri
- Nozzle icon mantığı

5.6. Heartbeat (MIN-POLL) Mantığı

- `_hb_start()`
- `_hb_stop()`
- `_hb_tick()`
- HB'nin state'e göre davranışı
- Loglama (periyodik TX)

5.7. Log Formatı

- Python log formatı C++'ta modüler hale getirilecek:
 - `log_user`
 - `log_runtime`
 - `log_pump`
 - `log_remote`

6. İlerleme Disiplini

6.1. Dosya Bazlı SHA Kontrolü

- Her teknik değişiklikte:
 - 1) Sen en güncel dosyayı yüklersin
 - 2) Ben dosyanın SHA256 / satır sayısı / byte uzunluğunu raporlarım
 - 3) Sen doğrulsın
 - 4) Ben minimal unified diff üretirim

6.2. Minimal Patch Kuralı

- Tüm dosyanın yeniden yazılması yok
- Her adımda yalnızca küçük bir bölgeye diff
- Gereksiz açıklama yok, net ve tek odaklı değişiklikler

6.3. Adım Sırası

- 1) Hedef belirlenir
- 2) En güncel dosya yüklenir
- 3) SHA doğrulanır

- 4) Patch verilir
- 5) Sen uygular, derler, test eder, sonucu bildirirsin
- 6) Sonraki adıma geçilir

6.4. Modüller Arası Ayrım

- hw/core/gui/rfid/net/utils arasında sınırlar korunur
- GUI sadece view-model ile çalışır
- core sadece pump + user mantığı
- hw sadece protokol
- net/rfid/utils kendi sorumluluk alanında kalır

7. Sprint-1 Son Durumu

- ReCUM12 proje yapısı tamamlandı
- CMake build sistemi tam çalışıyor
- VS Code + Remote + Debug stabil
- GitHub repo aktif
- Python win_pump_ctrl_rs485.py tek kaynak olarak seçildi
- Modüler C++ mimari tasarımları netleşti
- Log/backup/patch/docs klasörleri hazır
- C++ kod portu henüz başlamadı; Sprint-2'de başlayacak

8. Sprint-2 Başlangıç Planı

- 1) PumpR07Protocol için CRC + BCD helper fonksiyonlarının C++'a taşınması
- 2) Frame builder fonksiyonlarının portu
- 3) parse_frame() iskeletinin doldurulması
- 4) PumpInterfaceLvl3'ün minimal implementasyonu