

Protokol váhy S5000 Lite

Popis komunikace mezi aplikací a kuchyňskou váhou S5000 Lite přes Bluetooth Low Energy: jak váhu najít, jaké pakety posílá, jak z nich spočítat hmotnost a jakými příkazy se váha ovládá.

REFERENČNÍ IMPLEMENTACE S5000/lib/s5000_protocol.dart	TRANSPORT BLE GATT, služba FFF0	STAV DOKUMENTU 5. 10. 2026
--	------------------------------------	-------------------------------

01

Míra ověření

Protokol pochází z výrobcovy dokumentace a byl implementován v testovací aplikaci S5000. Ne všechno bylo vyzkoušeno na skutečné váze. Každý příkaz v tomto dokumentu je proto označen jedním ze tří stavů.

- Ověřeno na zařízení
- Pozorováno odlišně od dokumentace
- Podle dokumentace, neověřeno

02

Vyhledání váhy

Váha se při skenování pozná podle jedné ze dvou věcí v reklamním paketu (advertisement). Stačí, když platí kterákoli z nich.

- Seznam služeb obsahuje UUID FFF0, celé 0000fff0-0000-1000-8000-00805f9b34fb.
- Data výrobce (AD typ 0xFF) začínají bajty FA FB.

Data výrobce

0 FA	1 FB	2 VV	3 TT	4... ..
hlavička	hlavička	verze	typ zařízení	další data, MAC

TYP (BAJT 3)	ZAŘÍZENÍ
00	osobní váha
01	tuková (tělesná) váha
02	kuchyňská váha, sem patří S5000 Lite
03	tlakoměr

Pozor na knihovny. Většina BLE knihoven (například flutter_blue_plus nebo Android ScanRecord) bere první dva bajty dat výrobce jako ID firmy a vrací je odděleně jako číslo v pořadí little-endian. Bajty FA FB pak dostanete jako klíč 0xFBFA a zbytek paketu jako hodnotu. Referenční aplikace celý paket skládá zpět takto: [key & 0xFF, key >> 8, ...value] .

03

GATT

UUID	TYP	VLASTNOST	SMĚR
FFF0	služba		
FFF1	charakteristika	notify	váha → aplikace
FFF2	charakteristika	write	aplikace → váha

- Plná podoba UUID je 0000fff1-0000-1000-8000-00805f9b34fb , analogicky pro fff0 a fff2 .
- Po připojení je potřeba zapnout notifikace na FFF1 zápisem 01 00 do deskriptoru CCCD 0x2902 . Většina knihoven to dělá jedním voláním.
- Do FFF2 referenční aplikace zapisuje s potvrzením (write with response), pokud to charakteristika umí, jinak bez potvrzení.
- Všechny pakety se vejdou do výchozího MTU (20 bajtů dat), změna MTU není potřeba.

04

Formát paketu

Každý paket v obou směrech začíná hlavičkou 0xFA . V Javě a Kotlinu se ta hodnota jako znaménkový bajt zobrazí jako -6 . Druhý bajt je kód příkazu nebo zprávy. Pakety nemají kontrolní součet.

Aplikace → váha

0 FA	1 CC	2 LL	3... DD
hlavička	příkaz	délka dat	data

Dokumentace uvádí, že za data se doplňuje výplň 0x00 . Referenční aplikace výplň neposílá a váha přesto reaguje. Příkazy bez dat (tára, baterie, historie) fungují i jako dvoubajtové pakety bez délky, u táry je to ověřeno.

Váha → aplikace

0 FA	1 CC	2... DD
hlavička	typ zprávy	data

Kódy se v obou směrech překrývají, ale mají jiný význam. Například FA 04 poslané do váhy je dotaz na počet záznamů historie, zatímco FA 04 přijaté z váhy je průběžná hmotnost. Vždy je tedy potřeba vědět, odkud paket přišel.

05

Váhový paket

Hmotnost posílá váha nepřetržitě, dokud se na ní něco mění, a to zprávou 0x04. Jakmile se hodnota ustálí, pošle zprávu 0x05. Obě mají stejné rozložení. Stejně rozložení má podle referenční implementace i záznam historie 0x06.

0 FA hlavička	1 04 04 / 05 / 06	2 SS stav	3 UU jednotka	4 DD des. místa	5 HI hodnota, horní	6 LO hodnota, dolní	7... 00 u kuchyňské váhy nuly
----------------------------	--------------------------------	------------------------	----------------------------	------------------------------	----------------------------------	----------------------------------	--

```
hodnota = ((HI << 8) | LO) / 10^DD
pokud SS == 2:  hodnota = -hodnota
```

Hodnota je 16bitové číslo bez znaménka v pořadí big-endian. Znaménko nese stavový bajt. Při jednom desetinném místě je tak největší zobrazitelná hodnota 6553,5 jednotky.

STAV (SS)	VÝZNAM
00	v pořádku
01	přetížení
02	záporná hodnota, typicky po táře, když se něco z váhy sundá
03	slabá baterie
04	hledání nuly, váha se právě nuluje

JEDNOTKA (UU)	VÝZNAM	POZNÁMKA
00	kg	
01	lb	
02	st:lb	složená jednotka, viz níže
03	g	běžná jednotka kuchyňské váhy
04	ml	
05	oz	
06	lb:oz	složená jednotka, viz níže

Stejné kódy jednotek platí i pro příkaz změny jednotky. U složených jednotek 02 a 06 nese podle dokumentace horní polovina bajtu DD počet desetinných míst první jednotky a dolní polovina druhé. Referenční aplikace bere jen dolní polovinu. Na zařízení to ověřené není.

Příklady

PAKET	VÝSLEDEK
FA 04 00 03 01 01 F4	průběžně 50,0 g 0x01F4 = 500, jedno desetinné místo
FA 05 00 04 00 02 58	ustáleno 600 ml 0x0258 = 600, bez desetinných míst
FA 04 02 03 00 00 1E	průběžně -30 g stav 02 = záporná hodnota

06

Zprávy z váhy

Všechny zprávy chodí notifikací na FFF1 .

KÓD	ZPRÁVA	FORMÁT
00-03	Potvrzení příkazu 00 až 03 00 uživatel, 01 čas, 02 jednotka, 03 tára	FA CC SS SS: 00 úspěch, 01 chyba
04	Průběžná hmotnost	FA 04 SS UU DD HI LO
05	Ustálená hmotnost	FA 05 SS UU DD HI LO ...
06	Záznam historie	FA 06 SS UU DD HI LO ...
07	Počet záznamů historie	FA 07 NN
08	Stav baterie	FA 08 PP PP = procenta, 0x55 = 85 %
09	Data tlakoměru	jen u jiného typu zařízení, váha je neposílá
0A	Odpověď kalibrace	FA 0A FF FF = fáze kalibrace

07

Příkazy do váhy

Všechny příkazy se zapisují do FFF2 .

KÓD	PŘÍKAZ	PAKET	ODPOVĚĎ	STAV
03	Tára (vynulování)	FA 03	FA 03 00	● Ověřeno stačí 2 bajty bez délky dat
02	Nastavení jednotky kódy jako v tabulce jednotek	FA 02 01 UU gramy: FA 02 01 03	FA 02 SS	● Dokumentace

KÓD	PŘÍKAZ	PAKET	ODPOVĚĎ	STAV
06	Stav baterie	FA 06	FA 08 PP	Pozorováno S5000 Lite na dotaz zřejmě neodpovídá
01	Synchronizace času unix čas v sekundách, big-endian	FA 01 04 T3 T2 T1 T0	FA 01 SS	Dokumentace
00	Nastavení uživatele UU číslo uživatele, XX pohlaví (00 žena, 01 muž), AA věk, HH výška v cm	FA 00 04 UU XX AA HH FA 00 04 01 01 1E AF	FA 00 SS	Dokumentace pro tělesné váhy, kuchyňská nejspíš ignoruje
04	Počet záznamů historie	FA 04	FA 07 NN	Dokumentace
05	Všechny záznamy historie	FA 05	n × FA 06 ...	Dokumentace
07	Jas LED displeje MM: 00 vypnuto, 01 postupné rozsvícení, 02 jasně	FA 07 01 MM	neuvedeno	Dokumentace
08	Kalibrace MM: 00 zobrazit interní kód, 01 spustit kalibraci	FA 08 01 MM	FA 0A FF	Dokumentace

Kalibrace může váhu rozladit. Příkaz FA 08 01 01 přepne váhu do kalibračního režimu. Bez správného kalibračního závaží může váha ztratit přesnost. Do běžné aplikace pro uživatele ho nedávejte.

08

Průběh komunikace

1. Naskenovat okolí a najít zařízení se službou FFF0 nebo s daty výrobce FA FB .
2. Připojit se a zjistit služby a charakteristiky.
3. Zapnout notifikace na FFF1 .
4. Volitelně poslat synchronizaci času a nastavit jednotku, například FA 02 01 03 pro gramy.
5. Přijímat zprávy 0x04 s průběžnou hmotností a 0x05 , když se hodnota ustálí.
6. Pro vynulování poslat FA 03 . Váha odpoví FA 03 00 .

Praktické poznámky

Automatické vypínání

Protokol nemá příkaz, kterým by šlo automatické vypnutí váhy zakázat. Referenční aplikace váhu drží zapnutou tím, že zhruba každých 20 sekund pošle neškodný příkaz. Jako výchozí používá znovu posláni aktuální jednotky, tedy `FA 02 01 UU`. Dotaz na baterii k tomu nestačí, protože na něj S5000 Lite zřejmě neodpovídá. Pokud by ani jednotka nestačila, aplikace umí ještě táru na prázdné váze nebo přepínání mezi g a ml, které přinutí firmware překreslit displej.

Znovupřipojení

Když se váha vypne, spojení spadne. Na Androidu se osvědčilo zavolat `connect` s parametrem `autoConnect = true`. Systém pak pasivně čeká a připojí se, jakmile váhu někdo zapne, bez opakovaných pokusů a timeoutů.

Oprávnění na Androidu

- Android 12 a novější: `BLUETOOTH_SCAN` a `BLUETOOTH_CONNECT`.
- Android 11 a starší: `BLUETOOTH`, `BLUETOOTH_ADMIN` a `ACCESS_FINE_LOCATION`.

Ukázky kódu

Dart (z referenční aplikace)

zkráceno

```
const int header = 0xFA;

// Příkazy do FFF2
List<int> tare() => const [header, 0x03];
List<int> setUnit(int unit) => [header, 0x02, 0x01, unit & 0xFF];
List<int> setTime(DateTime t) {
  final ts = t.millisecondsSinceEpoch ~/ 1000;
  return [header, 0x01, 0x04,
    (ts >> 24) & 0xFF, (ts >> 16) & 0xFF, (ts >> 8) & 0xFF, ts & 0xFF];
}

// Váhový paket z FFF1 (zpráva 0x04 nebo 0x05)
double? parseWeight(List<int> d) {
  if (d.length < 7 || d[0] != header) return null;
  if (d[1] != 0x04 && d[1] != 0x05) return null;
  final status = d[2];
  final decimals = (d[3] == 0x02 || d[3] == 0x06) ? d[4] & 0x0F : d[4];
  var value = ((d[5] << 8) | d[6]) / math.pow(10, decimals);
  if (status == 2) value = -value;
  return value;
}
```

Python s knihovnou bleak

ilustrační ukázka, na váze nespustěná

```
import asyncio
from bleak import BleakClient, BleakScanner
```

```

NOTIFY = "0000fff1-0000-1000-8000-00805f9b34fb"
WRITE  = "0000fff2-0000-1000-8000-00805f9b34fb"
UNITS  = {0: "kg", 1: "lb", 2: "st:lb", 3: "g", 4: "ml", 5: "oz", 6: "lb:oz"}

def on_data(_, data: bytearray):
    if len(data) >= 7 and data[0] == 0xFA and data[1] in (0x04, 0x05):
        dec = data[4] & 0x0F if data[3] in (2, 6) else data[4]
        value = ((data[5] << 8) | data[6]) / 10 ** dec
        if data[2] == 2:
            value = -value
        state = "ustáleno" if data[1] == 0x05 else "průběžně"
        print(f"{value:.{dec}f} {UNITS.get(data[3], '?')} ({state})")

async def main():
    dev = await BleakScanner.find_device_by_filter(
        lambda d, adv: "0000fff0-0000-1000-8000-00805f9b34fb" in adv.service_uuids)
    async with BleakClient(dev) as client:
        await client.start_notify(NOTIFY, on_data)
        await client.write_gatt_char(WRITE, bytes([0xFA, 0x03])) # tára
        await asyncio.sleep(60)

asyncio.run(main())

```

11

Otevřené otázky

Tyto body dokumentace popisuje nejasně nebo nebyly vyzkoušeny na zařízení. Pokud je někdo ověří, stojí za to je doplnit.

- Desetinná místa u složených jednotek st:lb a lb:oz, tedy zda platí rozdělení bajtu DD na dvě poloviny.
- Přesný formát záznamu historie `0x06` a zda obsahuje i čas měření.
- Zda váha vyžaduje výplň `0x00` za daty příkazu a do jaké délky.
- Pořadí bajtů času v příkazu `0x01`. Referenční aplikace posílá big-endian.
- Zda S5000 Lite odpovídá na dotaz na baterii za jiných podmínek, například jen po synchronizaci času.
- Jaké odpovědi posílá váha na příkazy LED a kalibrace.

Sestaveno z projektu S5000: soubory `lib/s5000_protocol.dart` a `lib/scale_session.dart`.