Протокол обміну інформацією між пристроями проєкту U-FOC. Версія 1.0

Загальна інформація про протокол

Для керування платою контролера двигуна і отримання від неї інформації застосовується протокол, який розроблений з урахуванням вимог та особливостей CAN шини, та UART порту. Структура даних протоколу однакова для обох інтерфейсів CAN та UART. Це дає змогу використовувати один обробник пакетів для обох інтерфейсів і спрощує перехід від UART до CAN шини під час розробки та масштабування проєкту.

Пакетна передача даних

Протокол використовує пакетну передачу даних. Кожен пакет має фіксовану довжину. Використовується два типи пакетів:

- **Request** - пакет-запит, який змушує плату контролера двигуна виконувати певні дії.
- **Responce** - відповідь на пакет-запит.

Пакет-запит містить обов'язкову інформацію про його тип, та ID пристрою до якого цей пакет направляється. Також може містити додаткову інформацію.
Пакет-відповідь містить обов'язкову інформацію про його тип, та ID пристрою який відправляє відповідь. Може містити корисну інформацію, або не містити корисної інформації і сама наявність пустого пакету-відповіді використовується як сигнал того, що пакет-запит було прийнято і оброблено. Для деяких пакетів-запитів відповідь може бути не обов'язкова. Тобто, допускається, що не на кожен пакет-запит буде надіслана відповідь.

Адресація пристроїв

Кожен пристрій, який має виконувати команди (плата контролера двигуна) має власний ID. ID - це ціле число, яке лежить у межах від 1 до 255 включно.

Кожен пакет-запит містить ID пристрою, якому відправляється цей пакет.
Кожен пакет-відповідь містить ID пристрою, який відправляє відповідь.

Якщо задати ID пристрою 0, це буде означати, що пакет призначений всім пристроям.

Примітка: При використанні інтерфейсу UART, можна підключити лише одну плату контролера двигуна, тому адресація не принципова. Програмне забезпечення, яке керує

платою контролера двигуна по UART надсилає пакети з ID отримувача “0”. Це зроблено для того, щоб підключена плата з будь-яким ID отримувала пакети.

Packet ID — типи пакетів

Packet ID, або **ID пакету** визначає його призначення. Для пакетів-запитів **Packet ID визначає** що має виконати отримувач при надходженні пакету. Це може бути команда на запуск двигуна, запит інформації, запит на зміну налаштувань, тощо. Саме **Packet ID** визначає ті дії, які буде виконувати отримувач.

Пакет-відповідь також має **Packet ID** і він означає на який пакет-запит сформовано відповідь.

Для того, щоб відрізняти пакет-запит від пакета-відповіді їхні Packet ID рознесено на два діапазони:

- 0..126 — Request (пакет-запит)
- 127..255 — Responce(пакет-відповідь)

У протоколі **Packet ID** пакету-відповіді формується як **Packet ID** пакету-запиту + 127. Це дозволяє швидко визначити на який пакет-запит надійшла відповідь.

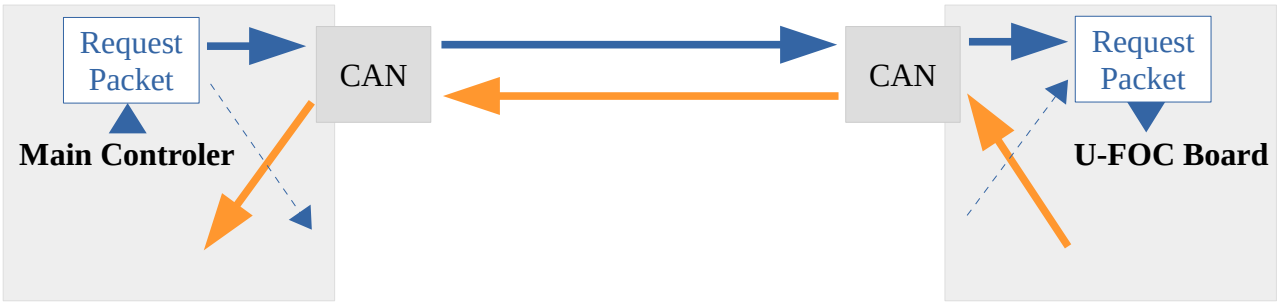
Обробка пакетів

Відправлення пакетів керівним пристроєм, обробка пакетів отримувачем і відправлення відповіді відбувається наступним чином.

Керівний пристрій формує, та відправляє пакет-запит (Request), у якому вказано ID пристрою отримувача, ID пакету який визначає що саме має виконати одержувач, та, можливо, необхідні данні. Пакет відправляється по одному з інтерфейсів CAN або UART.

Пакет приймається по одному з інтерфейсів CAN або UART, залежить від того, який інтерфейс Ви вирішили застосовувати у Вашому проєкті. У випадку використання CAN - шини, пакет отримують всі отримувачі. Після отримання пакету він віддається програмному обробнику який визначає за ID пристрою, чи призначений цей пакет йому. Якщо так - виконує дії, формує пакет-відповідь, і відправляє у той інтерфейс, з якого прийшов пакет-запит.

Керівний пристрій отримує пакет-відповідь який містить ID пристрою, що відповів, Packet ID, за яким визначається на який саме пакет-запит прийшла відповідь.



мал. 1

Технічно є можливість використовувати одночасно обидва інтерфейси CAN та UART, але НЕ рекомендується це робити, щоб не спричинити плутанину.

Загальна структура пакету

Пакет має фіксовану дожину у 8 байт. Це максимально можлива довжина пакета для протоколу CAN-шини. Тому саме такий формат пакету був обраний за стандарт для обох інтерфейсів.

Загальна структура пакету, та призначення кожного байту:

Byte 0 - Device ID
Byte 1 - Packet ID - Тип пакету (0..126 - Request, 127..255 - Responce)
Byte 2 - Data
Byte 3 - Data
Byte 4 - Data
Byte 5 - Data
Byte 6 - Data
Byte 7 - Data

Де:

- **Device ID** - ID пристрою, якому відправляється пакет у випадку пакету-запиту (Request). Device ID пристрою, який відповідає на запит у випадку пакету-відповіді (Responce) . Якщо **Device ID** = 0 — це означає, що пакет-запит для всіх пристроїв.
- **Packet ID** - код пакету, який визначає що це за пакет. У випадку пакету-запиту (Request) означає яку саме дію має виконати отримувач. У випадку пакету-відповіді (Responce) містить **Packet ID запиту + 127**. Зверніть увагу, що нумерація пакетів-запитів від 0..126. Тобто отримавши пакет можна швидко визначити це Request, чи Responce пакет.
- **Data** - данні, які може містити пакет. Для кожного типу пакету, данні можуть бути різні.

Обробка пакетів

Відповідь на пакет-запит може бути, а може і ні. Пакет може містити актуальні данні, або данні можуть не мати значення. Наприклад, пакет-запит, який зупиняє мотор не містить актуальної інформації. Байти з 2 по 7 не мають значення. Але вони будуть передані, по

довжина пакета фіксована. А от відповідь на пакет-запит про поточний стан системи керування мотором буде містити актуальну інформацію.

Пакети отримують всі пристрої які працюють на одній CAN-шині. При використанні UART, може бути лише два пристрої - один керівний, що віддає команди і один що виконує команди.

Відповідь на пакет-запит може бути не обов'язковою. Якщо Ваша реалізація дозволяє, пакети-запити можуть відправлятися, а відповідь на них не очікуватися. Але це не завжди доцільно, тому що керівний контролер, або комп'ютер не буде напевно знати чи є зв'язок з пристроєм. Наявність відповіді, як мінімум, підтверджує наявність зв'язку.

Отримавши пакет-запит, пристрій порівнює вказаний **Device ID** зі своїм і визначає чи потрібно йому його обробляти. Якщо так, далі по значенню **Packet ID** визначається чи здатен пристрій взагалі обробляти цей тип пакетів, і коли так, то виконуються відповідні операції, при при необхідності з використанням даних які містить пакет-запит. Потім, якщо потрібно, формується пакет-відповідь.

Пакет-відповідь містить **Device ID** пристрою який, власно, відповідає; **Packet ID**, який формується як **Packet ID** пакету-запиту + 127; при необхідності заповнює решту байтів пакету даними. Після чого пакет відправляється у той інтерфейс (CAN або UART) з якого було отримано Request пакет.

Структура пакетів запитів та відповідей на них

PACKET_GET_VERSION

Request (Packet ID = 0)

Byte	**Content**	**Example**
0	Device ID	1
1	Packet ID	1
2	Empty data	0
3	Empty data	0
4	Empty data	0
5	Empty data	0
6	Empty data	0
7	Empty data	0

Responce (Packet ID = 127)

Byte	**Content**	**Example**
0	Device ID	1

1	Packet ID	128
2	Version number	1
3	Comment	U
4	Comment	-
5	Comment	F
6	Comment	O
7	Comment	C

PACKET_GET_INF_1

Request (Packet ID = 1)

Byte	**Content**	**Example**
0	Device ID	1
1	Packet ID	1
2	Empty data	0
3	Empty data	0
4	Empty data	0
5	Empty data	0
6	Empty data	0
7	Empty data	0

Responce (Packet ID = 128)

Byte	**Content**	**Example**
0	Device ID	1
1	Packet ID	128
2	Reserved	0
3	Reserved	0
4	Speed RPM byte1	7
5	Speed RPM byte0	105
6	State	3
7	Mode	1

Speed = (byte1<<8) | byte0
Speed =(7<<8) | 105 = 1890 (RPM)

State:
MC_STATE_INIT 0
MC_STATE_IDLE 1
MC_STATE_START 2
MC_STATE_RUN 3
MC_STATE_STOP 4
MC_STATE_FAULT 5

Mode:
MC_MODE_TORQUE 0
MC_MODE_SPEED 1

PACKET_GET_INF_2

Request (Packet ID = 2)

Byte	Content	Example
0	Device ID	1
1	Packet ID	2
2	Empty data	0
3	Empty data	0
4	Empty data	0
5	Empty data	0
6	Empty data	0
7	Empty data	0

Responce (Packet ID = 129)

Byte	Content	Example
0	Device ID	1
1	Packet ID	129
2	Voltage byte1	0
3	Voltage byte0	12
4	Temperature byte1	0
5	Temperature byte0	25
6	Fault	0
7		0

Voltage = (byte1<<8) | byte0
Voltage = (0<<8) | 12 = 12 (V)

Temperature = (byte1<<8) | byte0
Temperature = (0<<8) | 25 = 25

Fault:

MC_NO_FAULTS	0
MC_FOC_DURATION	1
MC_OVER_VOLT	2
MC_UNDER_VOLT	3
MC_OVER_TEMP	4
MC_START_UP	5
MC_SPEED_FDBK	6
MC_BREAK_IN	7
MC_SW_ERROR	8

PACKET_GET_INF_3

Request (Packet ID = 3)

Byte	**Content**	**Example**
0	Device ID	1
1	Packet ID	3
2	Empty data	0
3	Empty data	0
4	Empty data	0
5	Empty data	0
6	Empty data	0
7	Empty data	0

Responce (Packet ID = 130)

Byte	**Content**	**Example**
0	Device ID	1
1	Packet ID	130
2	Power byte1	1
3	Power byte0	15
4	IVector byte1	0
5	IVector byte0	48
6		0
7		0

Power = (byte1<<8) | byte0
Power = (1<<8) | 15 = 271 (W)

IVector = (byte1<<8) | byte0
IVector = (0<<8) | 48 = 48 (умовні позиції)

PACKET_GET_INF_4

Request (Packet ID = 4)

Byte	**Content**	**Example**
0	Device ID	1
1	Packet ID	4
2	Empty data	0
3	Empty data	0
4	Empty data	0
5	Empty data	0
6	Empty data	0
7	Empty data	0

Responce (Packet ID = 131)

Byte	**Content**	**Example**
0	Device ID	1
1	Packet ID	131
2	Speed Task byte1	7
3	Speed Task byte0	8
4	Torque Task byte1	20
5	Torque Task byte0	75
6		0
7		0

Speed Task = (byte1<<8) | byte0
Speed Task = (7<<8) | 8 = 1800 (RPM)

Torque Task = (byte1<<8) | byte0
Torque Task = (20<<8) | 75 = 5195

PACKET_SET_SPEED

Request (Packet ID = 10)

Byte	Content	Example
0	Device ID	1
1	Packet ID	10
2	Speed RPM byte1	8
3	Speed RPM byte0	105
4		0
5		0
6		0
7		0

Speed = (byte1<<8) | byte0
Speed =(7<<8) | 105 = 1890 (RPM)

No Responce

PACKET_SET_TORQUE

Request (Packet ID = 11)

Byte	Content	Example
0	Device ID	1
1	Packet ID	11
2	Torque byte1	8
3	Torque byte0	75
4		0
5		0
6		0
7		0

Torque Task = (byte1<<8) | byte0
Torque Task = (20<<8) | 75 = 5195

No Responce

PACKET_START

Request (Packet ID = 12)

Byte	Content	Example
0	Device ID	1
1	Packet ID	12
2	Empty data	0
3	Empty data	0
4	Empty data	0
5	Empty data	0
6	Empty data	0
7	Empty data	0

No Responce

PACKET_STOP

Request (Packet ID = 13)

Byte	Content	Example
0	Device ID	1
1	Packet ID	13
2	Empty data	0
3	Empty data	0
4	Empty data	0
5	Empty data	0
6	Empty data	0
7	Empty data	0

No Responce

PACKET_RESET_FAULTS

Request (Packet ID = 14)

Byte	Content	Example
0	Device ID	1
1	Packet ID	14
2	Empty data	0
3	Empty data	0
4	Empty data	0

5	Empty data	0
6	Empty data	0
7	Empty data	0

No Responce

PACKET_GET_PID_IA

Request (Packet ID = 20)

Byte	**Content**	**Example**
0	Device ID	1
1	Packet ID	20
2	Empty data	0
3	Empty data	0
4	Empty data	0
5	Empty data	0
6	Empty data	0
7	Empty data	0

Responce (Packet ID = 147)

Byte	**Content**	**Example**
0	Device ID	1
1	Packet ID	147
2	Kp byte1	1
3	Kp byte0	244
4	Ki byte1	1
5	Ki byte0	44
6	Reserved for Kd byte1	0
7	Reserved Kd byte0	0

Kp = (byte1<<8) | byte0
Kp = (1<<8) | 244 = 500

Ki = (byte1<<8) | byte0
Ki = (1<<8) | 44 = 300

PACKET_GET_PID_IM

Request (Packet ID = 21)

Byte	Content	Example
0	Device ID	1
1	Packet ID	21
2	Empty data	0
3	Empty data	0
4	Empty data	0
5	Empty data	0
6	Empty data	0
7	Empty data	0

Responce (Packet ID = 148)

Byte	Content	Example
0	Device ID	1
1	Packet ID	148
2	Kp byte1	6
3	Kp byte0	64
4	Ki byte1	2
5	Ki byte0	88
6	Reserved for Kd byte1	0
7	Reserved Kd byte0	0

Kp = (byte1<<8) | byte0
Kp = (6<<8) | 64 = 1600

Ki = (byte1<<8) | byte0
Ki = (2<<8) | 88 = 600

PACKET_GET_PID_SPEED

Request (Packet ID = 22)

Byte	Content	Example
0	Device ID	1
1	Packet ID	22
2	Empty data	0

3	Empty data	0
4	Empty data	0
5	Empty data	0
6	Empty data	0
7	Empty data	0

Responce (Packet ID = 149)

Byte	**Content**	**Example**
0	Device ID	1
1	Packet ID	149
2	Kp byte1	0
3	Kp byte0	200
4	Ki byte1	0
5	Ki byte0	100
6	Reserved for Kd byte1	0
7	Reserved Kd byte0	0

Kp = (byte1<<8) | byte0
Kp = (0<<8) | 200 = 200

Ki = (byte1<<8) | byte0
Ki = (0<<8) | 100 = 100

PACKET_SET_PID_IA

Request (Packet ID = 23)

Byte	**Content**	**Example**
0	Device ID	1
1	Packet ID	23
2	Kp byte1	1
3	Kp byte0	244
4	Ki byte1	1
5	Ki byte0	44
6	Reserved for Kd byte1	0
7	Reserved Kd byte0	0

Kp = (byte1<<8) | byte0

Kp = (1<<8) | 244 = 500

Ki = (byte1<<8) | byte0
Ki = (1<<8) | 44 = 300

No Responce

PACKET_SET_PID_IM

Request (Packet ID = 24)

Byte	Content	Example
0	Device ID	1
1	Packet ID	24
2	Kp byte1	6
3	Kp byte0	64
4	Ki byte1	2
5	Ki byte0	88
6	Reserved for Kd byte1	0
7	Reserved Kd byte0	0

Kp = (byte1<<8) | byte0
Kp = (6<<8) | 64 = 1600

Ki = (byte1<<8) | byte0
Ki = (2<<8) | 88 = 600

No Responce

PACKET_SET_PID_SPEED

Request (Packet ID = 25)

Byte	Content	Example
0	Device ID	1
1	Packet ID	25
2	Kp byte1	0
3	Kp byte0	200
4	Ki byte1	0
5	Ki byte0	100
6	Reserved for Kd byte1	0
7	Reserved Kd byte0	0

Kp = (byte1<<8) | byte0
Kp = (0<<8) | 200 = 200

Ki = (byte1<<8) | byte0
Ki = (0<<8) | 100 = 100
No Responce

PACKET_GET_SETTINGS_1

Request (Packet ID = 30)

Byte	**Content**	**Example**
0	Device ID	1
1	Packet ID	30
2	Empty data	0
3	Empty data	0
4	Empty data	0
5	Empty data	0
6	Empty data	0
7	Empty data	0

Responce (Packet ID = 157)

Byte	**Content**	**Example**
0	Device ID	1
1	Packet ID	157
2	RpmMax byte1	11
3	RpmMax byte0	184
4	CurrentMax byte1	0
5	CurrentMax byte0	50
6	PowerMax byte1	0
7	PowerMax byte0	34

PACKET_GET_SETTINGS_2

Request (Packet ID = 31)

Byte	**Content**	**Example**

0	Device ID	1
1	Packet ID	31
2	Empty data	0
3	Empty data	0
4	Empty data	0
5	Empty data	0
6	Empty data	0
7	Empty data	0

Responce (Packet ID = 158)

Byte	**Content**	**Example**
0	Device ID	1
1	Packet ID	158
2	VoltageMax byte1	0
3	VoltageMax byte0	16
4	VoltageMin byte1	0
5	VoltageMin byte0	8
6	TemperatureMax byte1	0
7	TemperatureMax byte0	70

PACKET_GET_SETTINGS_3

Request (Packet ID = 32)

Byte	**Content**	**Example**
0	Device ID	1
1	Packet ID	32
2	Empty data	0
3	Empty data	0
4	Empty data	0
5	Empty data	0
6	Empty data	0
7	Empty data	0

Responce (Packet ID = 159)

Byte	Content	Example
0	Device ID	1
1	Packet ID	159
2	RotorPolePairs	2
3	DevID	1
4	BrakeVoltageOn byte1	0
5	BrakeVoltageOn byte0	15
6	BrakeVoltageOff byte1	0
7	BrakeVoltageOff byte0	13

PACKET_GET_SETTINGS_4

Request (Packet ID = 33)

Byte	Content	Example
0	Device ID	1
1	Packet ID	33
2	Empty data	0
3	Empty data	0
4	Empty data	0
5	Empty data	0
6	Empty data	0
7	Empty data	0

Responce (Packet ID = 160)

Byte	Content	Example
0	Device ID	1
1	Packet ID	160
2	Filter_Ia byte1	3
3	Filter_Ia byte0	232
4	Filter_Im byte1	3
5	Filter_Im byte0	232
6	Filter_Speed byte1	0
7	Filter_Speed byte0	6

PACKET_GET_SETTINGS_5

Request (Packet ID = 34)

Byte	Content	Example
0	Device ID	1
1	Packet ID	34
2	Empty data	0
3	Empty data	0
4	Empty data	0
5	Empty data	0
6	Empty data	0
7	Empty data	0

Responce (Packet ID = 161)

Byte	Content	Example
0	Device ID	1
1	Packet ID	161
2	TorqueMax byte1	11
3	TorqueMax byte0	184
4		0
5		0
6		0
7		0

PACKET_GET_SETTINGS_6

Request (Packet ID = 35)

Byte	Content	Example
0	Device ID	1
1	Packet ID	35
2	Empty data	0
3	Empty data	0
4	Empty data	0
5	Empty data	0

6	Empty data	0
7	Empty data	0

Responce (Packet ID = 162)

Byte	**Content**	**Example**
0	Device ID	1
1	Packet ID	162
2	HallAngleOffset byte1	0
3	HallAngleOffset byte0	0
4	Revers	0
5		0
6		0
7		0

PACKET_SET_SETTINGS_1

Request (Packet ID = 40)

Byte	**Content**	**Example**
0	Device ID	1
1	Packet ID	40
2	RpmMax byte1	11
3	RpmMax byte0	184
4	CurrentMax byte1	0
5	CurrentMax byte0	50
6	PowerMax byte1	0
7	PowerMax byte0	34

No Responce

PACKET_SET_SETTINGS_2

Request (Packet ID = 41)

Byte	**Content**	**Example**
0	Device ID	1
1	Packet ID	41

2	VoltageMax byte1	0
3	VoltageMax byte0	16
4	VoltageMin byte1	0
5	VoltageMin byte0	8
6	TemperatureMax byte1	0
7	TemperatureMax byte0	70

No Responce

PACKET_SET_SETTINGS_3

Request (Packet ID = 42)

Byte	**Content**	**Example**
0	Device ID	1
1	Packet ID	42
2	RotorPolePairs	2
3	DevID	1
4	BrakeVoltageOn byte1	0
5	BrakeVoltageOn byte0	15
6	BrakeVoltageOff byte1	0
7	BrakeVoltageOff byte0	13

No Responce

PACKET_SET_SETTINGS_4

Request (Packet ID = 43)

Byte	**Content**	**Example**
0	Device ID	1
1	Packet ID	43
2	Filter_Ia byte1	3
3	Filter_Ia byte0	232
4	Filter_Im byte1	3
5	Filter_Im byte0	232
6	Filter_Speed byte1	0
7	Filter_Speed byte0	6

No Responce

PACKET_SET_SETTINGS_5

Reserved

PACKET_SET_SETTINGS_6

Request (Packet ID = 45)

Byte	Content	Example
0	Device ID	1
1	Packet ID	45
2	HallAngleOffset byte1	0
3	HallAngleOffset byte0	0
4	Revers	0
5		0
6		0
7		0

No Responce

PACKET_SAVE_SETTINGS

Request (Packet ID = 50)

Byte	Content	Example
0	Device ID	1
1	Packet ID	50
2	Empty data	0
3	Empty data	0
4	Empty data	0
5	Empty data	0
6	Empty data	0
7	Empty data	0

Responce (Packet ID = 177)

Byte	Content	Example
0	Device ID	1
1	Packet ID	177
2		0
3		0
4		0
5		0
6		0
7		0

PACKET_RESET_SETTINGS

Request (Packet ID = 51)

Byte	Content	Example
0	Device ID	1
1	Packet ID	51
2	Empty data	0
3	Empty data	0
4	Empty data	0
5	Empty data	0
6	Empty data	0
7	Empty data	0

Responce (Packet ID = 178)

Byte	Content	Example
0	Device ID	1
1	Packet ID	178
2		0
3		0
4		0
5		0
6		0
7		0