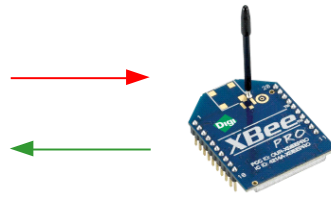


## 1. Trabalhando com API's

### 1.1 Exemplo de Aplicação - Lendo parâmetro do dispositivo local



#### API 08 (AT Command):

**0x7E 0x00 0x04 0x08 0x01 0x56 0x52 0x4E**

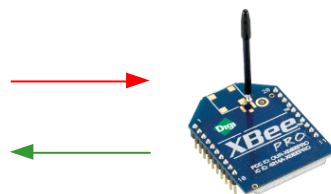
- 00 04 = Tamanho do pacote de informação (08 01 56 52)
- 08 = Função API
- 01 = Número de identificação do frame
- 56 52 = Comando AT requisitado (representação Hexa dos caracteres ASC – comando VR)
- 4E = Checksum da API [0xFF - (08 + 01 + 56 + 52)]

#### API 88 (AT Command Response):

**0x7E 0x00 0x07 0x88 0x01 0x56 0x52 0x00 0x21 0x41 0x6C**

- 00 07 = Tamanho do pacote de informação (88 01 56 52 00 21 41)
- 88 = Função API
- 01 = Número de identificação do frame
- 56 52 = Comando AT requisitado (representação Hexa dos caracteres ASC – comando VR)
- 00 = Status de entrega do pacote
- 21 41 = Valor do comando solicitado
- 6C = Checksum da API [0xFF - (88 + 01 + 56 + 52 + 00 + 21 + 41)]

### 1.2 Exemplo de Aplicação - Escrevendo parâmetro no dispositivo local



#### API 08 (AT Command):

**0x7E 0x00 0x05 0x08 0x01 0x44 0x31 0x04 0x7D**

- 00 05 = Tamanho do pacote de informação (08 01 44 31 04)
- 08 = Função API
- 01 = Número de identificação do frame
- 44 31 = Comando AT requisitado (representação Hexa dos caracteres ASC – comando D1)
- 04 = Valor a ser escrito no parâmetro

7D = Checksum da API [0xFF - (08 + 01 + 44 + 31 + 04)]

#### API 88 (AT Command Response):

**0x7E 0x00 0x05 0x88 0x01 0x44 0x31 0x00 0x01**

00 05 = Tamanho do pacote de informação (88 01 44 31 00)

88 = Função API

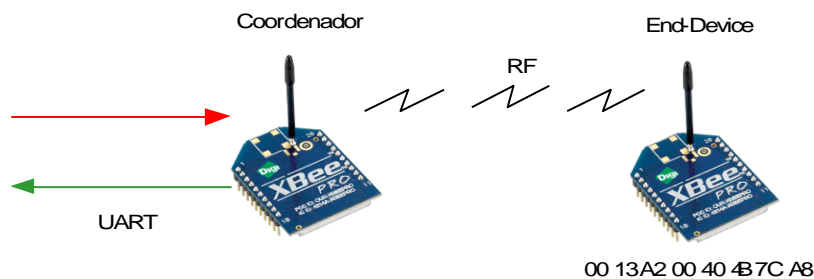
01 = Número de identificação do frame

44 31 = Comando AT requisitado (representação Hexa dos caracteres ASC – comando D1)

00 = Status de entrega do pacote

01 = Checksum da API [0xFF - (88 + 01 + 44 + 31 + 00)]

### 1.3 Exemplo de Aplicação - Lendo parâmetro do dispositivo remoto



#### API 17 ( Remote AT Command Request):

**0x7E 0x00 0x0F 0x17 0x01 0x00 0x13 0xA2 0x00 0x40 0x4B 0x7C 0xA8 0xFF 0xFE 0x02 0x43 0x48 0xF9**

00 0F = Tamanho do pacote de informação (17 01 00 13 A2 00 40 4B 7C A8 FF FE 02 43 48)

17 = Função API

01 = Número de identificação do frame (Se 0 não gera resposta)

00 13 A2 00 40 4B 7C A8 = endereço 64bits do módulo remoto

FF FE = endereço 16bits do módulo remoto (ou 0xFFFF se é desconhecido)

02 = Aplica mudanças no módulo remoto (no caso de escrita)

43 48 = Comando AT requisitado (representação Hexa dos caracteres ASC – comando: CH)

F9 = Checksum da API [0xFF - (17+01+00+13+A2+00+40+4B+7C+A8+FF+FE+02+43+48)]

#### API 97 (Remote AT Command Response):

**0x7E 0x00 0x10 0x97 0x01 0x00 0x13 0xA2 0x00 0x40 0x4B 0x7C 0xA8 0xFA 0x7E 0x43 0x48 0x00 0x13 0xED**

00 10 = Tamanho do pacote de informação (97 01 00 13 A2 00 40 4B 7C A8 FA 7E 43 48 00 13)

97 = Função API

01 = Número de identificação do frame

00 13 A2 00 40 4B 7C A8 = endereço 64bits do módulo remoto

FA 7E = endereço 16bits do módulo remoto

43 48 = Comando AT requisitado (representação Hexa dos caracteres ASC – comando CH)

00 = Status de entrega do pacote

13 = Valor do comando solicitado

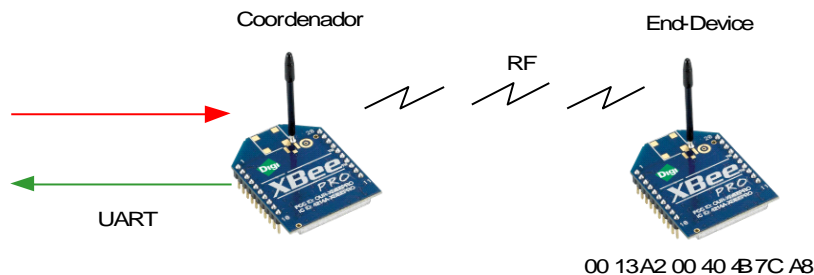
ED = Checksum da API [0xFF -

(97+01+00+13+A2+00+40+4B+7C+A8+FA+7E+43+48+00+13)]

### 1.4 Exemplo de Aplicação - Solicitando informações dos I/Os digitais e Conversor A/D do dispositivo remoto (Comando IS)

a) Configuração do End-Device:

D1 = 3 – Digital input  
D2 = 2 – ADC



#### API 17 ( Remote AT Command Request):

**0x7E 0x00 0x0F 0x17 0x01 0x00 0x13 0xA2 0x00 0x40 0x4B 0x7C 0xA8 0xFF 0xFE 0x02 0x49 0x53 0xE8**

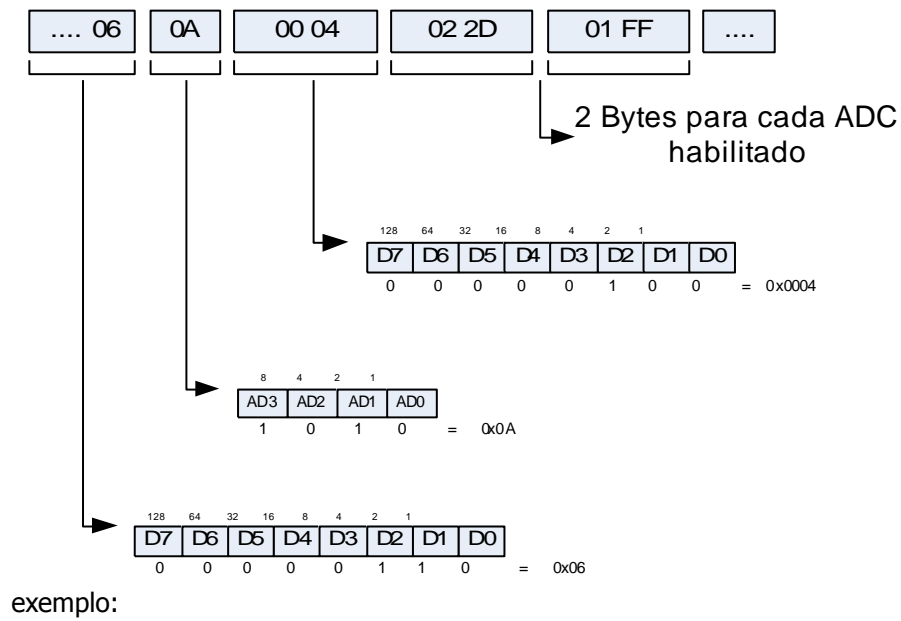
00 0F = Tamanho do pacote de informação (17 01 00 13 A2 00 40 4B 7C A8 FF FE 02 49 53)  
 17 = Função API  
 01 = Número de identificação do frame (Se 0 não gera resposta)  
 00 13 A2 00 40 4B 7C A8 = endereço 64bits do módulo remoto  
 FF FE = endereço 16bits do módulo remoto (ou 0xFFFF se é desconhecido)  
 02 = Aplica mudanças no módulo remoto (no caso de escrita)  
 49 53 = Comando AT requisitado (representação Hexa dos caracteres ASC – comando: IS)  
 E8 = Checksum da API [0xFF – (17+01+00+13+A2+00+40+4B+7C+A8+FF+FE+02+49+53)]

#### API 97 (Remote AT Command Response):

**0x7E 0x00 0x17 0x97 0x01 0x00 0x13 0xA2 0x00 0x40 0x4B 0x7C 0xA8 0xFA 0x7E 0x49 0x53 0x00 0x01 0x00 0x02 0x04 0x00 0x02 0x02 0x33 0xB1**

00 17 = Tamanho do pacote de informação (97 01 00 13 A2 00 40 4B 7C A8 FA 7E 49 53 00 01 00 02 04 00 02 02 33)  
 97 = Função API  
 01 = Número de identificação do frame  
 00 13 A2 00 40 4B 7C A8 = endereço 64bits do módulo remoto  
 FA 7E = endereço 16bits do módulo remoto  
 49 53 = Comando AT requisitado (representação Hexa dos caracteres ASC – comando IS)  
 00 = Status de entrega do pacote  
 01 00 = Informação de que existe IO ou ADC habilitados  
 02 = Bitfield que informa quais IOs estão habilitados  
 04 = Bitfield que informa quais ADC estão habilitados  
 00 02 = 2 bytes que informam os estados dos IOs habilitados  
 02 33 = 2 bytes que representam o valor convertido do AD (são 2 bytes para cada ADC habilitado)  
 B1 = Checksum da API [0xFF – (97+01+00+13+A2+00+40+4B+7C+A8+FA+7E+49+53+00+01+00+02+04+00+02+02+33)]

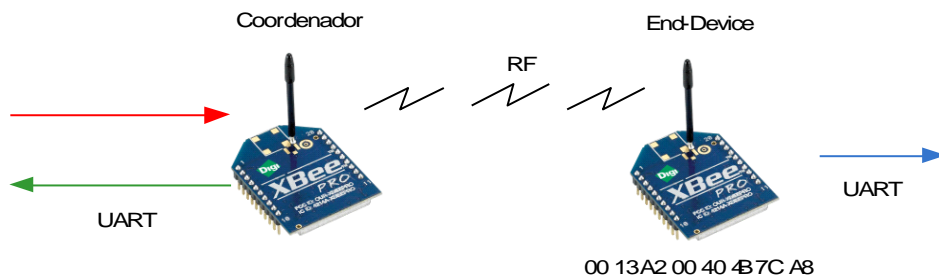
**1.4.1 Como interpretar os Bitfields**



---

**1.5 Exemplo de Aplicação - Transferindo informações para UART do dispositivo remoto**


---

**API 10 ( ZigBee Transmit Request):**

**0x7E 0x00 0x14 0x10 0x01 0x00 0x13 0xA2 0x00 0x40 0x4B 0x7C 0xA8 0xFF 0xFE 0x00 0x00 0x7A 0x69 0x67 0x62 0x65 0x65 0x17**

00 14 = Tamanho do pacote de informação (10 01 00 13 A2 00 40 4B 7C A8 FF FE 00 00 7A 69 67 62 65 65)

10 = Função API

01 = Número de identificação do frame (Se 0 não gera resposta)

00 13 A2 00 40 4B 7C A8 = endereço 64bits do módulo remoto

FF FE = endereço 16bits do módulo remoto (ou 0xFFFF se é desconhecido)

00 = Broadcast Radius

00 = Multicast/Unicast

7A 69 67 62 65 65 = Informação a ser transmitida ("zigbee") (máx: 72 bytes)

17 = Checksum da API [0xFF -

(10+01+00+13+A2+00+40+4B+7C+A8+FF+FE+00+00+7A+69+67+62+65+65)]

**API 8B (ZigBee Transmit Status):**

**0x7E 0x00 0x07 0x8B 0x01 0xFA 0xFE 0x00 0x00 0x00 0xFB**

00 07 = Tamanho do pacote de informação (8B 01 FA FE 00 00 00)

8B = Função API

01 = Número de identificação do frame

FA FE = endereço 16bits do módulo remoto que o pacote foi entregue

00 = Quantidade de tentativas para transmissão do pacote

00 = Status de entrega do pacote

00 = Status informando a necessidade de discovery na rede

FB = Checksum da API [0xFF - (8B+01+FA+FE+00+00+00)]

**API 90 (ZigBee Receive Packet):**

**0x7E 0x00 0x12 0x90 0x00 0x13 0xA2 0x00 0x40 0x4B 0x7C 0xBC 0x00 0x00 0x01 0x7A 0x69 0x67 0x62 0x65 0x65 0x80**

00 12 = Tamanho do pacote de informação (90 00 13 A2 00 40 4B 7C BC 00 00 01 7A 69 67 62 65 65)

90 = Função API

00 13 A2 00 40 04 7C BC = endereço 64bits do módulo que enviou a informação

00 00 = endereço 16bits do módulo remoto que enviou a informação

01 = packet acknowledged

7A 69 67 62 65 65 = Informação recebida via RF e enviada pela serial ("zigbee") (máx: 72 bytes)

80 = Checksum da API [0xFF -

(90+00+13+A2+00+40+4B+7C+BC+00+00+01+7A+69+67+62+65+65)]

### 1.6 Exemplo de Aplicação - Recebendo informações dos I/Os digitais e Conversor A/D do dispositivo remoto ciclicamente (cada 5 segundos)

a) Configuração do End-Device:

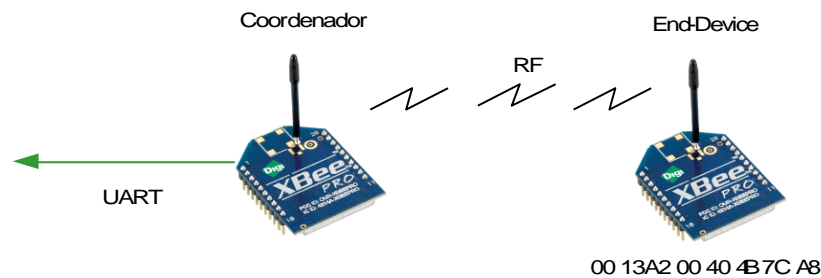
D1 = 3 – Digital input

D2 = 2 – ADC

IR = 0x1388

DH = 0 (64bits High do dispositivo remoto que receberá pacote – 0 = Coordenador)

DL = 0 (64bits Low do dispositivo remoto que receberá pacote – 0 = Coordenador)



#### API 92 ( ZigBee IO Data Sample Rx Indicator ):

**0x7E 0x00 0x14 0x92 0x00 0x13 0xA2 0x00 0x40 0x4B 0x7C 0xA8 0xD4 0x64 0x01 0x01 0x00 0x02 0x04 0x00 0x02 0x02 0x31 0x94**

00 14 = Tamanho do pacote de informação (92 00 13 A2 00 40 4B 7C A8 D4 64 01 01 00 02 04 00 02 02 31)

92 = Função API

00 13 A2 00 40 4B 7C A8 = endereço 64bits do módulo remoto

D4 64 = endereço 16bits do módulo remoto (ou 0xFFFE se é desconhecido)

01 = packet acknowledged

01 00 = Num Samples

02 = Indica quais IOs estão habilitados \*\*

04 = Indica quais ADC estão habilitados \*\*

00 02 = os bits contém a informação dos estados dos IOs \*\*

02 31 = indicam a conversão do A/D (são 2 bytes para cada conversor A/D habilitado) \*\*

94 = Checksum da API [0xFF -

(92+00+13+A2+00+40+4B+7C+A8+D4+64+01+01+00+02+04+00+02+02+31)]

\*\* Para entender o funcionamento do bitfields vide tópico: 1.4.1