ME 218C: King of the Atoll Communications Protocol

Draft 3– 5/4/2011

Table of Contents

Message Standards

Format

The following are tables of standard message formats. Each of the specific command sections lists the data frames that must be included.

Standard Transmitted Message

Byte	Description	Value(s)
1	Start Delimiter	[0x7E]
2	Length (High Byte)	[0x00]
3	Length (Low Byte)	N minus 3
4	API Identifier	[0x01]
5	Frame ID	User defined. See Descriptions
6	Destination Address (High Byte)	Address
7	Destination Address (Low Byte)	Address
8	Options	See Descriptions
9	Command Type	0x66, 0x77, 0x89, 0xAB, 0xCD, or 0xEF (see Desc.)
10→N	Data Frame	See Individual Commands
N+1	Checksum	See Descriptions

Transmit Status Message

Byte	Description	Value
1	Start Delimiter	[0x7E]
2	Length MSB	[0x00]
3	Length LSB	[0x03]
4	API Identifier	[0x89]
5	Frame ID	User defined. See Descriptions
6	Status	0x00, 0x01, 0x02, 0x03 (See descriptions)
7	Checksum	See Descriptions

Received Message

Byte	Description	Value
1	Start Delimiter	[0x7E]
2	Length (High Byte)	[0x00
3	Length (Low Byte)	N minus 3
4	API Identifier	[0x81]
5	Source Address (High Byte)	Address
6	Source Address (Low Byte)	Address
7	Received Signal Strength Indic.	Hexadecimal Value equal to signal strength in dB
8	Options	See Descriptions
9	Command Type Byte	0x66, 0x77, 0x89, 0xAB, 0xCD, 0xEF
10→N	Data Frame	See Individual Commands
N+1	Checksum	See description below.

[…] Indicates this is a specified byte for this type of message.

Descriptions

API Identifier

This identifies the type of message sent over the XBee. Value 0x01 is for transmit, 0x89 for transmit status, and 0x81 for receive.

Frame ID

This is user-defined. When you send an addressed packet and enable ACK, you will receive an acknowledgement from the receiver's radio saying that they received the packet. That ACK will contain this frame ID so you can figure out which packet you sent is being acknowledged.

Status (transmit status only)

This is for transmitted status messages. The follow values indicate:
0x00: Success
0x01: No ACK received
0x02: CCA failure
0x03: Purged

Options

For **transmitting**, this allows the user to determine whether or not they want to receive an acknowledgement from the receiving XBee. Whenever a team transmits a **direct** message, it should expect an acknowledgement. After transmitting a message, the XBee will report whether or not the message was successful or not. The frame format for this message is listed above in the Format section.

For **receiving**, bit 1 indicates whether the message was direct (bit 1 high) or broadcast (bit 1 low).

Checksum

To **calculate**: Not including frame start delimiters and length, add all bytes (4→N) keeping only the lowest 8 bits of the result and subtract from 0xFF. To **verify**: Add all bytes (include checksum, but not the delimiter and length). If the checksum is correct, the sum will equal 0xFF.

Command Type Bytes

Document Abbrev.	Byte	Type	Broadcast Type
RQ	0x66	Request Capture	All
RP	0x67	Request Capture Reply	Direct
FT	0x89	Find Teammate	All
AAC	0xAB	Announce Atoll Capture	All
C2B	0xCD	CVC to ACV Command	Direct
B2C	0xEF	ACV to CVC Command	Direct

Data Frame

The data frame are the bytes, with the exception of the checksum, that follow the command type byte. In each message (except for transmit status messages), the data frame starts at byte 10. Technically, the command type byte and the data frame here constitute the RF Data Frame for the XBee. The data frame defined here should not be confused with the RF Data Frame defined in the XBee datasheet.

Request Capture of Atoll (RQ)

Description

This is the message that an ACV sends when it attempts to capture an atoll. The data frame consists of AN,S1, S2, S3, SK1, SK2, SK3 bytes (see below).

Transmission

Steps

- Send the packet
- Wait for a Transmit Status message from your XBee radio.
- If the status was success (check byte 6), then you're done.
- Else resend the packet till 3 tries.

Format

Byte No.	Transmission format	Example	Comments
1	0x7E	[0x7E]	Start Delimiter
2	Length (High Byte)	[0x00]	Length of this packet (high byte).
3	Length (Low Byte)	[0x0D]	Length of this packet (low byte).
4	API Identifier	[0x01]	Transmit Request
5	Frame ID	0x03	User defined to correlate it with the Tx status
6	Destination Address (High Byte)	[0xFF]	Broadcast Message
7	Destination Address (Low Byte)	[0xFF]	Broadcast Message
8	Options Byte	[0x00]	Broadcast Message
9	Command Type Byte	[0x66]	Request Capture
10	AN	0x01	Atoll number (#1)
11	S1	0x83	Most Significant Byte (received first) of the Atoll serial number from the RFID reader.
12	S2	0xFC	2nd Byte of the Atoll serial number from RFID reader.
13	S3	0xE0	Least Significant Byte (received last) of the Atoll serial number from the RFID reader.
14	SK1	0x05	Most Significant Byte (received first) security key value from the security controller.
15	SK2	0x05	2nd Byte of the security key value from security controller.
16	SK3	0x05	Least Significant Byte (received last) security key value from the security controller.
17	Check Sum	0x00	

[…] Indicates this is a required byte for this type of message.

Receiving

Steps

This is a broadcast so everybody will receive the message. If Byte 9 is 0x66 (Request Capture), **then ignore this message.**

Format

Byte No.	Receive Format	Receive example	Comments
1	0x7E	[0x7E]	Start Delimiter
2	Length (High Byte)	[0x00]	Length of this packet (high byte).
3	Length (Low Byte)	[0x0D]	Length of this packet (low byte).
4	API Identifier	[0x81]	Receive Packet
5	Source Address (High Byte)	[0x21]	High Byte of Boat ID
6	Source Address (Low Byte)	0x83	Low Byte of Boat ID (in example, 0x2183
7	RSSI	0x20	Not used
8	Options Byte	0bXXXXXX0X	Indicates broadcast.
9	Command Type	[0x66]	Request Capture
10	Atoll Number	0x01	Must be a value between 0x01 and 0x05.
11	S1	0x83	Most Significant Byte (received first) of the Atoll serial number from the RFID reader.
12	S2	0xFC	Second Byte of the Atoll serial number from the RFID reader.
13	S3	0xE0	Least Significant Byte (received last) of the Atoll serial number from the RFID reader.
14	SK1	0x05	Most Significant Byte (received first) security key value from the security controller.
15	SK2	0x05	Second Byte of the security key value from the security controller.
16	SK3	0x05	Least Significant Byte (received last) security key value from the security controller.
17	Check Sum	0x00	

[…] Indicates this is a specified byte for this type of message.

Request Capture Reply from Atoll (RP)

Description

This is a direct message. It is sent only to the ACV that broadcasted out an RQ message. The data frame consists of AN,SF, and NO bytes (see below).

Transmission

N/A

Receiving

Steps

After sending a request to capture an atoll (RQ), the atoll well send back a Request Capture Reply (RP) message directed to your ACV. If you receive a success (byte SF, byte #11), you should transmit an Announcement that you Captured an Atoll (AAC). If you receive a failure on your transmit status, you should attempt to request capture (RQ) again.

Example

In the example, an XBee received a message with signal strength of -40dBm from atoll 5, which told it that the atoll was successfully captured for the green team.

Byte No.	Description	Receive example	Comments
1	Start Delimiter	[0x7E]	Always 0x7E
2	Length MSB	[0x00]	For this command type, always 0x00
3	Length LSB	[0x09]	For this command type, always 0x09
4	API Identifier	[0x81]	0x81 indicates a message to receive
5	Source Address MSB	0xFF	For this command type, this is the atoll address MSB
6	Source Address LSB	0xFF	For this command type, this is the atoll address LSB
7	RSSI	0x28	Signal strength indicator
8	Options	0bXXXXXX1X	Bit 1 high= Address Broadcast.
9	Command type	[0x67]	RP command
10	AN	0x05	Atoll number (0x01 through 0x05)
11	SF	0xFF	Success (0xFF) or Failure(0x00)
12	NO	0x01	New Owner (Green = 0x01, Red = 0xFE, unoccupied = 0x88)
13	Checksum	0xEA	

[…] Indicates this is a specified byte for this type of message.

Finding Teammate (FT)

Description

If two boats are on the same team, the operators will successively scan the team RFID card on each of the boats' card readers. Here is the sequence for establishing your teammate. For the sake of this example, you are Boat A assigned to red, your unknown teammate is Boat B, also on red. The data frame is a single byte, which is the NO byte. 0x01 = Green team, and 0xFE = red team.

Protocol

1. Boat A scans a red card with its RFID reader, so team color is red.
2. Boat A waits 1 second listening for a team mate's "Hello world, I'm on red, anyone there?" message broadcast.
    a. See "Receive Broadcast Team Message" table below
        i. If Boat A receives a broadcast from a teammate (Boat B), go to Case: Boat A Second (i.e. second to read card on the red team)
        ii. Otherwise after that amount of time, Boat A decides it's the first to read it's card on the red team, go to Case: Boat A First
3. Case: Boat A is First:
    a. Boat A starts Broadcasting the "Hello world, I'm on red, anyone there?" message itself
        i. Transmitting a maximum of 255 times at 5 Hz
            1. See "Send Broadcast Team Message"
            2. See the general "Transmit Status Message" above
        ii. While also listening for communication of "Hi! I'm also on red!" message from a team mate
            1. See "Receive Directed Reply Team Acknowledgement Message" table below
            2. If directed communication is received, your Modem should automatically send an ACK (thanks to the option settings) and team mate is confirmed
                a. Stop Broadcasting
                b. Raise your flag!
            3. Else go to Case: Time Out
4. Case: Boat A is Second:
    a. Parse out the Xbee hardware address from Boat B's received message (See "Receive Broadcast Team Message" table below)
    b. Send a directed reply (with ACK) to Boat B's Address with a byte that has the red team flag
        i. See "Send Directed Reply Team Acknowledgement" table below
        ii. Try to resend the directed reply for a maximum of 10 times at 5 Hz or until ACK is confirmed.
            1. If this doesn't work, go to Case: Time Out
            2. Upon successful ACK: Stop retrying to send, and raise your flag!
5. Case: Time Out

a. In the very unlikely event of a timeout from this process, assign yourself a default value of teammate, raise your flag anyway and move on.

Transmit Broadcast "Team Message" for Finding Teammate Format

This is the Data Frame that goes to the modem for sending the Broadcast Team Message letting your soon-to-be teammate know you are out there.

Byte No.	Description	Receive example	Comments
1	Start Delimiter	[0x7E]	Always 0x7E
2	Length MSB	[0x00]	For this command type, always 0x00
3	Length LSB	[0x07]	For this command type, always 0x07
4	API Identifier	[0x01]	0x01 indicates a message to Send
5	Frame Identifier	0xDD	User Defined
6	Destination Address MSB	[0xFF]	For Broadcast, this is 0XFF
7	Destination Address LSB	[0xFF]	For Broadcast, this is 0XFF
8	Options	[0x00]	Don't disable ACK
9	Command Byte	[0x89]	Find Teammate Identifier FT
10	Team Color	0xFE	Select proper byte for team color
11	Checksum		

Receive Broadcast "Team Message" when attempting to Find Teammate Format

This is the Data Frame that will come from the modem when you have received a Broadcast Team Message from a teammate.

Byte No.	Description	Receive example	Comments
1	Start Delimiter	[0x7E]	Always 0x7E
2	Length MSB	[0x00]	For this command type, always 0x00
3	Length LSB	[0x07]	For this command type, always 0x07
4	API Identifier	[0x81]	0x81 indicates a message to receive
5	Source Address MSB	[0x21]	Broadcaster ID MSB Generated by modem
6	Source Address LSB	0x85	Broadcaster ID LSB Generated by modem
7	RSSI	0x28	Signal strength indicator
8	Options	0bXXXXXX0X	Broadcast
9	Command Byte	[0x89]	Find Teammate Identifier FT
10	Team Color	0xFE	Indicates appropriate team color
11	Checksum	0x82	

[…] Indicates this is a specified byte for this type of message.

Send Directed Reply Team Acknowledgment Message Format

This is the Data Frame that will be sent to the modem in order to send a Directed Reply Team Acknowledgment.

Byte No.	Description	Receive example	Comments
1	Start Delimiter	[0x7E]	Always 0x7E
2	Length MSB	[0x00]	For this command type, always 0x00
3	Length LSB	[0x07]	For this command type, always 0x07
4	API Identifier	[0x01]	0x01 indicates a message to Send
5	Frame Identifier	0xCC	User Defined
6	Destination Address MSB	[0x21]	MSB of Teammate's Address
7	Destination Address LSB	0x85	LSB of Teammate's Address
8	Options	0x00	Don't disable ACK
9	Command Byte	0x89	Find Teammate Identifier FT
10	Team Color	0xFE	Select proper byte for team color
11	Checksum	0x84	

Receive Directed Reply Team Acknowledgment Message Format

This is the Data Frame that will come from the modem when a teammate successfully sends a Directed Reply Team Acknowledgement Message to you. This would indicate the confirmation that your teammate knows who you are, and you can stop sending out Broadcasts as described in the detailed instructions above.

Byte No.	Description	Receive example	Comments
1	Start Delimiter	0x7E	Always 0x7E
2	Length MSB	0x00	For this command type, always 0x00
3	Length LSB	0x07	For this command type, always 0x07
4	API Identifier	0x81	0x81 indicates a message to receive
5	Source Address MSB	[0x21]	MSB of Address of Teammate
6	Source Address LSB	0x86	LSB of Address of Teammate
7	RSSI	0x28	Signal strength indicator
8	Options	0bXXXXXX1X	Bit 1 indicates direct message
9	Command Byte	0x89	Find Teammate Identifier FT
10	Team Color	0xFE	Team color (Green = 0x01, Red = 0xFE)
11	Checksum	0x49	

[…] Indicates this is a specified byte for this type of message.

Announcing Atoll Capture (AAC)

Description

This message lets everyone know that you have captured an atoll. It is very important you don't screw up and transmit this when you shouldn't; there is no game master that keeps track of which atolls belong to whom. **This message must be transmitted and must only be transmitted after your ACV receives a successful request capture reply (RQ). It must only be transmitted by your ACV.**

The data frame consists of two bytes: the team color byte (NO) and the atoll number byte (AN).

Transmission

Steps

- The message should be sent twice with a 200ms interval between the two transmits.
- After each message transmission, you should expect to receive the message received data from the XBee as defined above.
    - Byte 6 determines whether the original broadcast message was successful.

Format

Byte	Transmission format	Transmission example	Comments
1	Start Delimiter	[0x7E]	Start delimiter for Xbee
2	Length (MSB)	[0x00]	MSB length
3	Length (LSB)	[0x08]	LSB length is 8
4	API Identifier	[0x01]	API identifier
5	Frame ID	0x01	User defined
6	Dest. Addr. (MSB)	[0xFF]	MSB address for broadcast message
7	Dest. Addr. (LSB)	[0xFF]	LSB address for broadcast message
8	Options	[0x00]	Options byte, send with Ack and no PAN ID
9	Command type	[0xAB]	Byte for announcing atoll capture (AAC)
10	Team Color	0xFE	Team Color (0x01 = Green, 0xFE = Red)
11	Atoll Number	0x05	Atoll number #5
12	Checksum	0xCF	

[…] Indicates this is a specified byte for this type of message.

Receiving

Steps

Other XBee radios receiving this message can update their statuses (on the CVC), ignore the message (ACV), or react in some other way. No confirmation needs to be sent. The data frame consists of two bytes: the team color byte (NO) and the atoll number byte (AN).

Format

Byte No.	Transmission format	Transmission example	Comments
1	Start delimiter	[0x7E]	Start delimiter for Xbee
2	Length (MSB)	[0x00]	MSB length
3	Length (LSB)	[0x08]	LSB length is 8
4	API Identifier	[0x81]	API identifier
5	Src. Addr. MSB	[0x21]	MSB of address of the sender
6	Src. Addr. LSB	0x85	LSB of address of the sender
7	RSSI	0x28	Received signal strength indicator, in -dBm
8	Options	0bXXXXXX1X	Options byte, bit 1 indicates address broadcast or direct communication (1=broadcast, 0=direct comm)
9	0xAB	[0xAB]	Byte for announcing atoll capture
10	Team Color	0x01	Green
11	Atoll number	0x05	Atoll #5
12	Checksum	0xD0	

[…] Indicates this is a specified byte for this type of message.

CVC to ACV Communications (C2B)

Description

This command type gives each team freedom to choose what they send from their CVC to their ACV. The number of bytes in the data frame that they send after the command type byte must be no more than 99.

Transmission

Steps

- Compile packet to send.
- Send it.
- Listen for ACK and compare to Frame ID.
    - If STATUS byte returned to you is 0 for success, then the transfer was successful.
    - If STATUS byte returned to you is 1 for when all retries are expired and no ACK is received, then resend. Repeat as necessary up to three times.
    - If STATUS byte returns 1 for all four attempts, handle as you see appropriate.
        - Ex. Light LED on CVC indicating communication has failed.
- Listen for response from CVC (Optional). Up to the team to implement if they wish.

Format

Byte No.	Description	Transmission example	Comments
1	Start Delimiter	[0x7E]	always 7E
2	Length (high byte)	[0x00]	Length (High Byte) of frame data, which is everything between the length bytes and the checksum
3	Length (low byte)	0x07	Length (Low Byte) of frame data (N-3)
4	API identifier	[0x01]	Value for “Transmit”
5	Frame ID	0x0F	User-defined.
6	Dest. Addr.	[0x21]	Destination address MSB
7	Dest. Addr.	0x85	Destination address LSB
8	Options byte	[0x00]	Options byte—setting to 0x00 means you will receive an ACK for your packet and that you are sending to a specific address.
9	Command Type	[0xCD]	First Data byte—signifies that this is a command for the AVC
10→N	Data	0xDF	Bytes 10 to N and onward are team--specific. In this particular example, DF could mean PWM the motors.
N+1	Checksum	0x69	

[…] Indicates this is a specified byte for this type of message.

Receiving

Steps

- Receive transmission.
- Identity as receiving packet. (Check for 0xCD for C2B).
- Optionally implement a transfer of successfully received packet back to CVC.

Format

Byte No.	Description	Receive example	Comments
1	Start Delimiter	[0x7E]	Always 0x7E
2	Length (MSB)	[0x00]	Length of the frame data, which is everything between the length (LSB) and the checksum
3	Length (LSB)	0x07	Length (Low Byte) of frame data (N-3)
4	API identifier	[0x81]	Always 0x81, which identifies that this is an incoming packet
5	Source Address (MSB)	[0x20]	
6	Source Address (LSB)	0x85	
7	Signal Strength	0x28	Hex number, the units are -dBm
8	Options	0bXXXXXX1X	Bit 1 – address broadcast, bit 2 = PAN broadcast. For this particular receive, bit 1 should always be set.
9	Command	[0xCD]	The first byte should be CD for this particular command type
10→N	Data	0xDF	Command to PWM motors
N+1	Checksum	0x69	

[…] Indicates this is a specified byte for this type of message.

ACV to CVC Communications (B2C)

Description

This command type gives each team freedom to choose what they send from their ACV to their CVC. The number of bytes they send in the data frame after the command type byte must be no more than 99.

Transmitting

Steps for transmission

- Compile packet to send.
- Send it.
- Listen for ACK and compare to Frame ID.
    - If STATUS byte returned to you is 0 for success, then the transfer was successful.
    - If STATUS byte returned to you is 1 for when all retries are expired and no ACK is received, then resend. Repeat as necessary three times.
    - If STATUS byte returns 1 for all four attempts, handle as you see appropriate.
        - Ex. Light LED on ACV.
- Listen for response from CVC (Optional). Up to the team to implement if they wish.

Format

Byte No.	Description	Transmission example	Comments
1	Start Delimiter	[0x7E]	always 7E
2	Length (high byte)	[0x00]	
3	Length (low byte)	0x07	Length (Low Byte) of frame data (N-3)
4	API identifier	[0x01]	Value for “Transmit”
5	Frame ID	0x0F	User-defined
6	Dest. Addr. (MSB)	[0x20]	Destination address MSB
7	Dest. Addr. (LSB)	0x85	Destination address LSB
8	Options byte	[0x00]	Options byte—setting to 0x00 means you will receive an ACK for your packet and that you are sending to a specific address.
9	Data	[0xEF]	First Data byte—signifies that this is a command for the AVC
10→n	Data	0xEF	Bytes 10 and onward are team-specific. In this particular example, 0xEF could be low battery status.
n+1	Checksum	0x52	

[…] Indicates this is a specified byte for this type of message.

Receiving

Steps

- Receive transmission.
- Identity as receiving packet. (Check for 0xEF for B2C).
- Optionally implement a transfer of successfully received packet back to AVC.

Format

Byte No.	Description	Receive example	Comments
1	Start Delimiter	0x7E	Always 0x7E
2	Length (MSB)	[0x00]	Length of the frame data, which is everything between the length (LSB) and the checksum
3	Length (LSB)	0x07	Length (Low Byte) of frame data (N-3)
4	API identifier	[0x81]	Always 0x81, which identifies that this is an incoming packet
5	Source Address (MSB)	[0x21]	Address of the sender
6	Source Address (LSB)	0x85	
7	Signal Strength	0x28	Hex number, the units are -dBm
8	Options	0bXXXXXX1X	Bit 1 – address broadcast, bit 2 = PAN broadcast. For this particular receive, bit 1 should always be set.
9	Command type	0xEF	B2C Command type
10→N	Data	0xEF	E.g. command to PWM motors
N+1	Checksum	0x51	

[…] Indicates this is a specified byte for this type of message.