ME218C 2018 Communications Protocol

Revision #		
1	**5/7/18**	**Initial Draft**
1.1	**5/10/18**	**Meet w/ Karl**
1.2	**5/11/18**	**Update State Diagrams to Reflect Unpair**
1.3	**5/17/18**	**Standardizing Ship Pairing Addresses**
1.4	**5/28/18**	**Revised Lost-Comm Time to 3s**

1. Definitions

Term	Definition1
ANSIBLE	Each team's mechanism for control- capable of sending and receiving data packets with the SHIP.
SHIP	Each team's watercraft- capable of receiving, decoding, and executing on data packets from the ANSIBLE.

2. Communications Overview

This communications protocol centers around the fact that an ANSIBLE must be paired with a SHIP in order for the ANSIBLE to be able to control it. When a SHIP is paired, it will ignore all other requests to pair with it. The SHIPs begin the game each paired to an ANSIBLE.

As the game progresses, a SHIP will have the capability to unpair from its ANSIBLE once its fuel tank has been depleted. Upon this event, the SHIP will indicated that it is unpaired, and may be paired with by an ANSIBLE from the opposite team. This presents the distinct communications actions of -

- Pairing an ANSIBLE to a SHIP
    - The action is engaged by the ANSIBLE to an unpaired SHIP. Upon receiving the request, the unpaired SHIP will send back an acknowledgement packet. Once

the ANSIBLE has received the acknowledgement packet. It can begin to send control messages, completing the pairing protocol.

- Unpairing a SHIP and ANSIBLE
    - This action is engaged by a paired SHIP. It will simply stop sending status packets to the ANSIBLE, and enter a state where it is waiting to be paired with.
- Controlling the SHIP from the ANSIBLE
    - This action is communicated from the ANSIBLE to its paired SHIP. The ANSIBLE will deliver a control packet to the SHIP containing commands for the ship to accept.
- Reporting status from SHIP to ANSIBLE
    - This message will be delivered from the SHIP to ANSIBLE. The ANSIBLE will display certain key indicators from within the packet.
- Dealing with errors in these transmissions
    - The errors during transmissions will be handled by use of a 3s watchdog timer. The driving idea is that the SHIP or ANSIBLE will try to circumvent the error for 3s. If it cannot, then it will fall back to a base state, or the state that it was previously in.

Further guidelines:

- All communications will be made through XBee radios in API mode.
- Any SHIP must be fully controllable from any ANSIBLE.
- Any ANSIBLE should be able to display the data from the status message of any SHIP.
- The communications rate will be at 5 Hz.

3. Hardware Requirements

a. ANSIBLE
    i. Each ANSIBLE must be capable of setting the team color.
    ii. Each ANSIBLE must have the capability to communicate with each SHIP.
    iii. Each ANSIBLE must be capable of displaying the unpaired, paired, and pairing states.
    iv. Each ansible must have control hardware for every byte in the control packet (defined in section 5).

b. SHIP
    i. Each SHIP must be capable of displaying the current team color, as well as the starting team color and paired status.
    ii. Each SHIP must be capable of ending a pairing when fuel has run out or when refueling has begun.

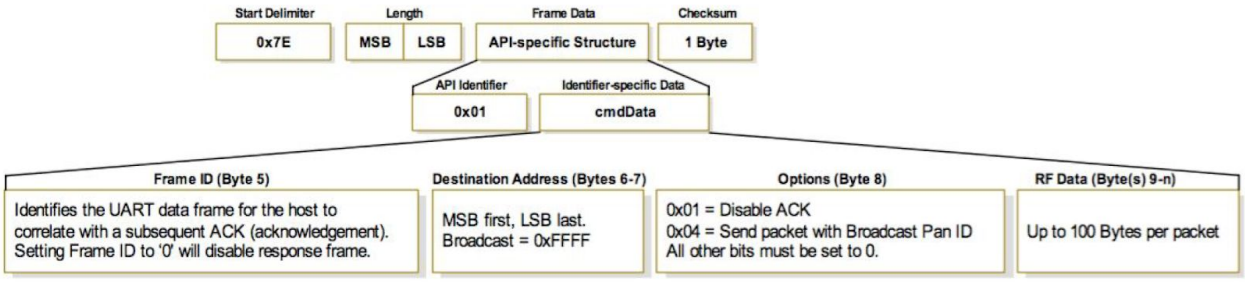
c. Xbee Standardization For Pairing
    i. All Xbees are standardized such that each team has 208# and 218#, where # is the corresponding team number [Team 1 is 2081 and 2181, team 11 is 208B and 218B]

ii. The XBee with address 208# MUST go on the ANSIBLE
iii. The XBee with the address 218# MUST go on the SHIP
iv. To pair to a specific SHIP, send to the address 218#, where # is the SHIP you are trying to pair to.
v. To determine which ANSIBLE is trying to pair with the SHIP, the Source address will be 208#, where # is the ANSIBLE team number.

4. XBee API Structure

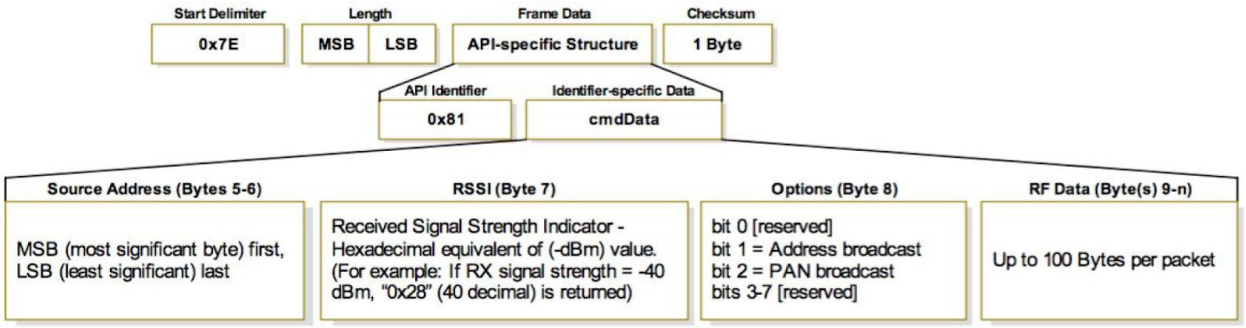
The figure below provides reference for the structure of the XBee Transmit (TX) packet used. All the packets described in Section 5 are contained within the RF data section of the transmission. The Frame ID must be a nonzero value to assure that the receiving XBee provides an acknowledge of receipt to the transmitting XBee. This acknowledgement is intended for use in individual teams' transmit functions, such that a software transmit module will attempt to transmit a message several times until an acknowledgement is generated. In the case of heavy RF traffic, this implementation ensures robustness. The options byte must be set to 0x00 so that no special options are enabled during transmission. Due to these software configurations, an XBee will provide the sender (Tiva) with an acknowledgement that an RF message at least made it to the destination XBee. While implementation of this functionality is not explicitly required in terms of this protocol, it is recommended for redundancy in checking message receipt during development.

Transmit request (16-bit address):

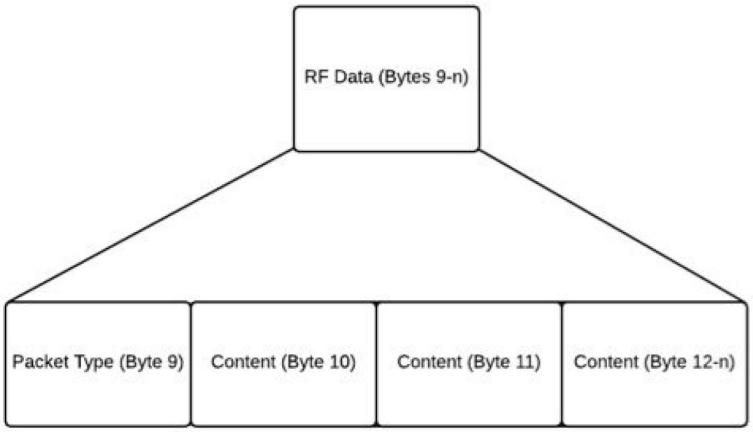


An outline of the receive packet has also been included below for reference. When the receiving XBee gets an RF packet, the information is output from the XBee's UART module to a microcontroller in the following format:

Receive packet (16-bit address):



RF Data structure:



All subsequent packets will be sent in bytes 9-n of the transmission. The number of content bytes will vary depending on the packet type. Packet types are outlined in Section 5.

5. Class Packets

General protocols:

- Send MSB to LSB (same as Xbee)
- 3s timeout before trying to pair again
- Everyone operates at 5Hz transmission rate

Packet Type	Direction	Header	Data Byte Contents
REQ_2_PAIR	ANSIBLE → SHIP	0x01	[ANSIBLE Team Color]
PAIR_ACK	SHIP→ ANSIBLE	0x02	Nothing
CTRL	ANSIBLE→ SHIP	0x03	[Forward/back][Left/right][Turret rotation][Turret pitch][Control byte]
STATUS	SHIP→ANSIBLE	0x04	[Fuel Status] [SHIP Status]

REQ_2_PAIR Packet

- A directed message from an ANSIBLE to a specific SHIP, containing the requester's team color.
- Length: header + 1 byte = 2 bytes
- Purpose: When a team identifies a SHIP it wants to pair with, the team can use its ANSIBLE to try to pair with it. The team color is sent so that the SHIP can change the on-board Team Color indicator.
- ANSIBLE XBee to SHIP XBee Standardization - see section 3.C

Header (Byte 0)	ANSIBLE Team Color (Byte 1)
0x01	0x0=Blue, 0x01 = Red

PAIR_ACK Packet

- A directed message from an *unpaired* SHIP to ANSIBLE after receiving a REQ_2_PAIR and validating that the ANSIBLE qualifies for pairing.
- Length: header = 1 byte
- Purpose: When the SHIP receives a REQ_2_PAIR message, it needs to accept or reject the request after validating the ANSIBLE. The PAIR_ACK message will be returned upon receiving REQ_2_PAIR message only if the SHIP is unpaired and the ANSIBLE is not the most recently paired ANSIBLE. If the ANSIBLE is invalid, no message will be sent back.

Header (Byte 0)
0x02

CTRL Packet

- A directed message from an ANSIBLE to the paired SHIP providing SHIP control data. Run at 5Hz.
- Length: header + Digital control byte + Analog Bytes = 6 bytes
- Byte 0: Header
- Byte 1: Analog Forward/Back
    - An unsigned 8-bit integer for analog forward/backward control.
        - 255 = max effort forward
        - 127 = no effort
        - 0 = max effort backwards
- Byte 2: Analog Ship Yaw (Left/Right)
    - An unsigned 8-bit integer for analog left/right control.
        - 255 = max effort rightward

        - 127 = no effort (straight)
        - 0 = max effort leftward
- Byte 3: Analog Turret Yaw
    - An unsigned 8-bit integer for analog turret position control in yaw
        - 255 = max angle clockwise
        - 127 = centered
        - 0 = max angle counter-clockwise
- Byte 4: Analog Turret Pitch
    - An unsigned 8-bit integer for analog turret position control in pitch
        - 255 = max angle up
        - 127 = centered
        - 0 = max angle down
- Byte 5: Digital Control
    - Bit 0: Shoot Water
        - 1 = ON
        - 0 = OFF
    - Bit 1: Valve/Refuel
        - 1 = self-refuel ON
        - 0 = self-refuel OFF (regular mode ON)
    - Bit 2: Reserved for Special Function
        - Up to individual team what special function this bit controls
- Purpose: The ANSIBLE provides the SHIP with information on how to control its on-board actuation. Upon receipt of 1 packet per second, the SHIP also knows that it is paired.

Digital Control (Byte 5)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Reserved	Reserved	Reserved	Reserved	Special function	Valve/ refuel	Shoot Water

STATUS Packet

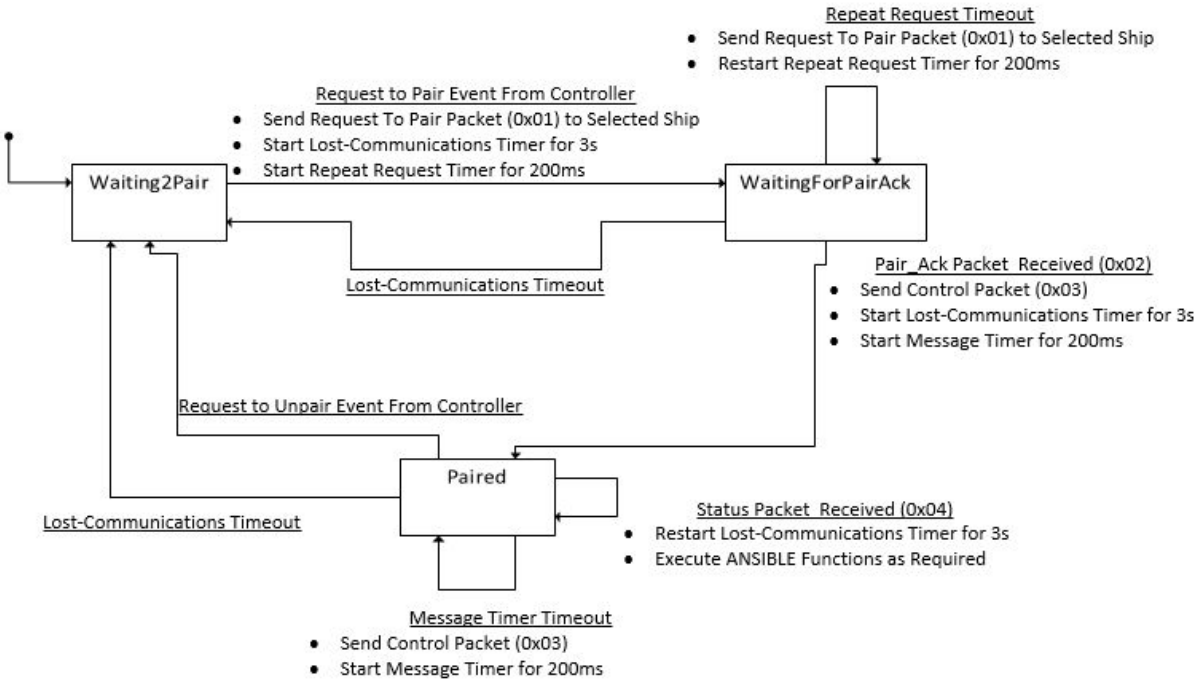
- A directed message from a SHIP to the paired ANSIBLE providing information on some on-board sensing status. Sent continuously at 5Hz while paired. Also functions as a heartbeat.
-
- Length: header + Fuel Status byte + SHIP Status byte = 3 bytes
- Byte 0: Header
- Byte 1: Fuel Status
    - Identical to the byte provided by Fuel Tank Specs

- Byte 2: SHIP Status
    - Bit 0: Shoot Water
    - Bit 1: Valve/refuel
        - 1 = Currently in self refuel mode
        - 0 = Currently in regular mode
    - Bit 2: Special function
        - Up to individual team what special function this bit controls
    - Bit 3: Battery Critical State
        - 1 = Battery in “bad” shape (defined by individual team)
        - 0 = Battery in “good” shape (defined by individual team) - this is the default state that should be transmitted if a team does not implement sensing on whether a battery is in “good” shape or not
- Purpose: By receiving the STATUS message at least once a second, the ANSIBLE can verify that it is paired with the SHIP. The message also gives information to the paired ANSIBLE on the fuel tank level, whether the refueling is being implemented, and (optionally) what the battery level is.

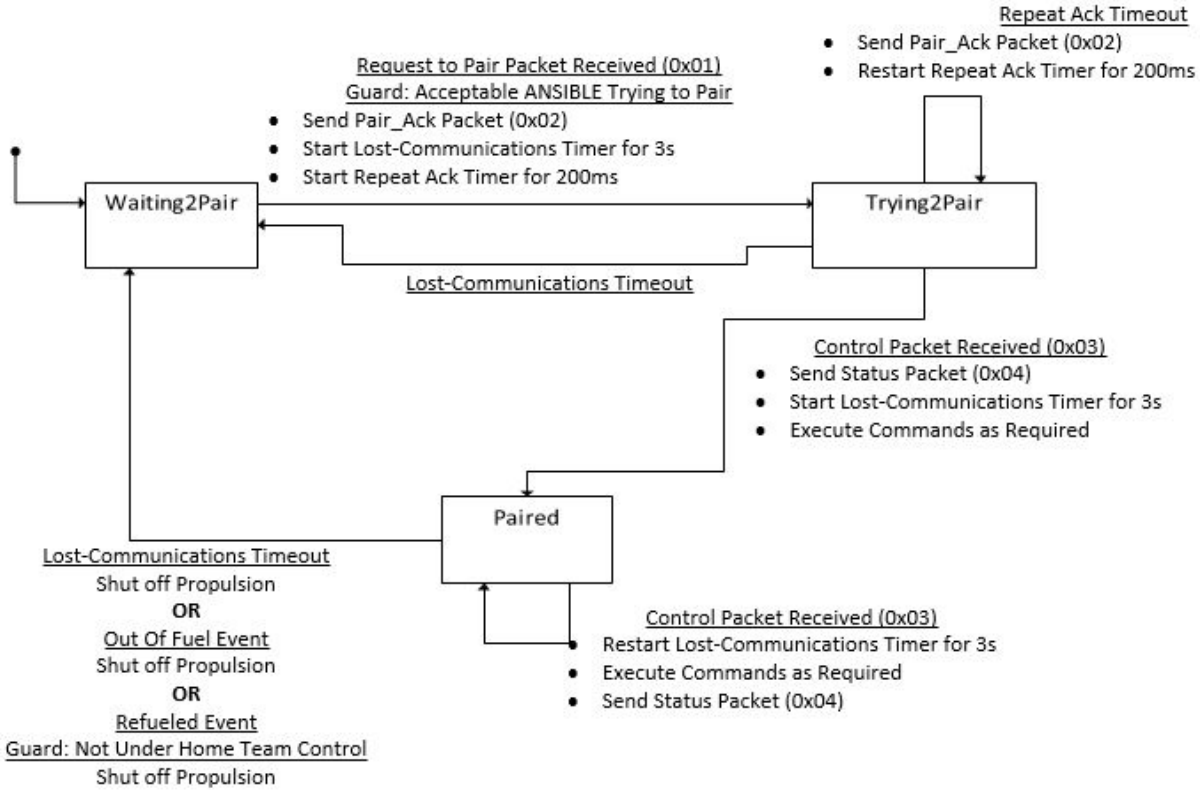
SHIP Status (Byte 2)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Reserved	Reserved	Reserved	Battery Critical State	Special function	Valve/ Refuel	Shoot Water

6. State Charts

A) ANSIBLE Communications



B) SHIP Communications



7. Error - Handling Sequencing

- If ANSIBLE does not receive a PAIR_ACK packet
    - The ANSIBLE should continue to send REQ_2_PAIR packets until 3s timer expires
- If ANSIBLE does not receive a Status Packet from the SHIP
    - The ANSIBLE should continue to send Control Packets until 3s timer expires
- If SHIP does not receive a Control Packet from the ANSIBLE
    - The SHIP should continue to send PAIR_ACK packets until 3s timer expires