Communications Protocol

ME218C

2009 Final Project

Draft 2, Revision E

Communications Committee
Friday, the 22nd of May, 2009

0 Revision History

05/05/2009	1-A	Pre-Release 1	Communications Committee
05/07/2009	1-B	Pre-Release 2	Communications Committee
05/08/2009	2-A	Initial Release	Communications Committee
05/11/2009	2-B	Corrected checksum description (pp. 5-6) Clarified when to use 2’s complement (p. 8) Clarified how 5 Hz limitation is enforced (p. 8) Added morale value for Jettison command (p. 9) Added morale value for Mutiny command (p. 9) Clarified morale during “Not Paired” (p. 11) Deleted free command responses (p. 11) Updated format of Context column (pp. 9-11) Changed superstate names to “Paired” (all)	Communications Committee
05/12/2009	2-C	Proof-read text (all) Replaced conversation diagrams with a single, comprehensive state chart (pp. 13-14) Filled in time-out and resend times (pp. 9-11)	Communications Committee
05/12/2009	2-D	Replaced state charts (pp. 13-14)	Communications Committee
05/22/2009	2-E	Listed timers involved in diagrams (pp. 13-14) Add Update Morale -10 to Receive Mutiny (p.13) Velocity Timer duration now < 1 sec (p. 13) Moved Velocity Timer start to transitions (p. 13) Asserting Command always to Available (p. 13) Don’t check Available when want Result (p. 13) Add Send Mutiny message to Morale=-10 (p. 14) Added free commands to TOWRP (p. 14) Corrected spelling error (p. 14)	Communications Committee

Table of Contents

1 Communications Context

All TOWRP/COACH communications for the ME 218C 2009 Final Project will occur via a Zigbee network implemented using XBee radios in API mode. This protocol depends on two types of XBee API packets: outgoing data packets and incoming data packets.

1.1 The XBee API Data Packets

A standard Xbee packet of data is composed of 4 parts (in sequence):

Part:	Size:
1) Start Delimiter	1 byte
2) Length of frame	2 bytes
3) Frame Data	7 bytes
4) Checksum	1 byte

The Frame Data section is divided into two separate parts:

Part:	Size:
1) API identifier	1 byte
2) Command Data	6 Bytes

The Command Data is where network address and data is inserted. The break-down of this section differs for transmit and receive packets.

1.1.1 The XBee API Transmit Packet

For the transmit packet, the Command Data is broken down into four parts:

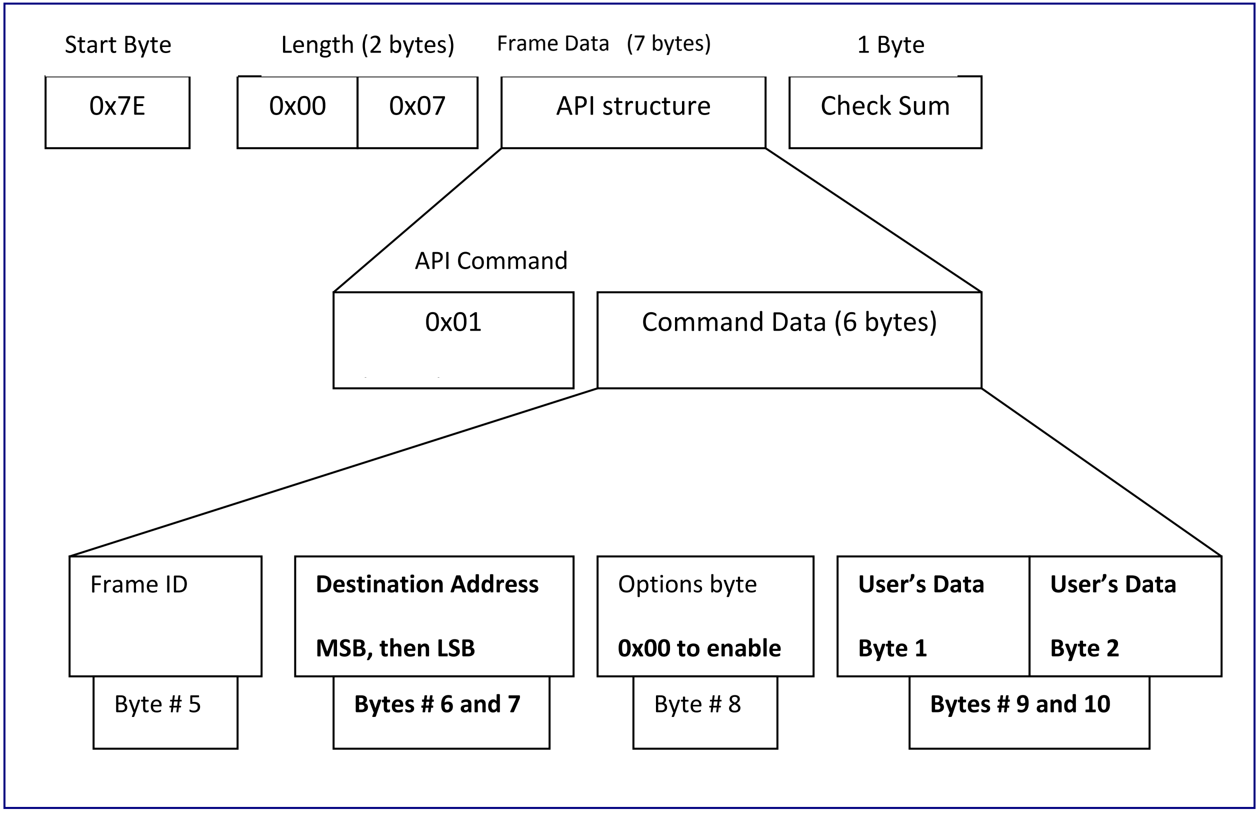
Part:	Size:
1) Frame ID	1 byte
2) Destination Address	2 bytes
3) Options	1 byte
4) RF Data (User's transmitted data)	2 bytes

In general, RF Data can be up to 100 bytes long, but in our case, the RF data is only 2 bytes long. Thus, ME218C users' transmitted data resides in bytes 9 and 10 of each message. Byte 9 contains the command/message "opcode", and byte 10 contains a parameter associated with that command/message (if any).

The standard message byte table for a transmit packet therefore looks like this:

Byte Number:	Description:	Value:
Byte 1)	Start Delimiter	0x7E
Byte 2)	Length of Frame 1	0x00
Byte 3)	Length of Frame 2	0x07
Byte 4)	Transmit API Identifier	0x01
Byte 5)	Frame ID	Reference Number
Byte 6)	Destination Address 1	See Addressing
Byte 7)	Destination Address 2	See Addressing
Byte 8)	Options	See XBee Doc
Byte 9)	Message Byte (Byte 1)	See RF Data
Byte 10)	Level Byte (Byte 2)	See RF Data
Byte 11)	Check Sum	0xFF – (Sum of Bytes 4-10)

In diagram form, this can be represented like this:



1.1.2 The XBee API Receive Packet

For the receive packet, the Command Data is broken down into four parts:

Part:	Size:
1) Source Address	2 bytes
2) Received Signal Strength Indicator	1 byte
3) Options	1 byte
4) RF Data (User's received data)	2 bytes

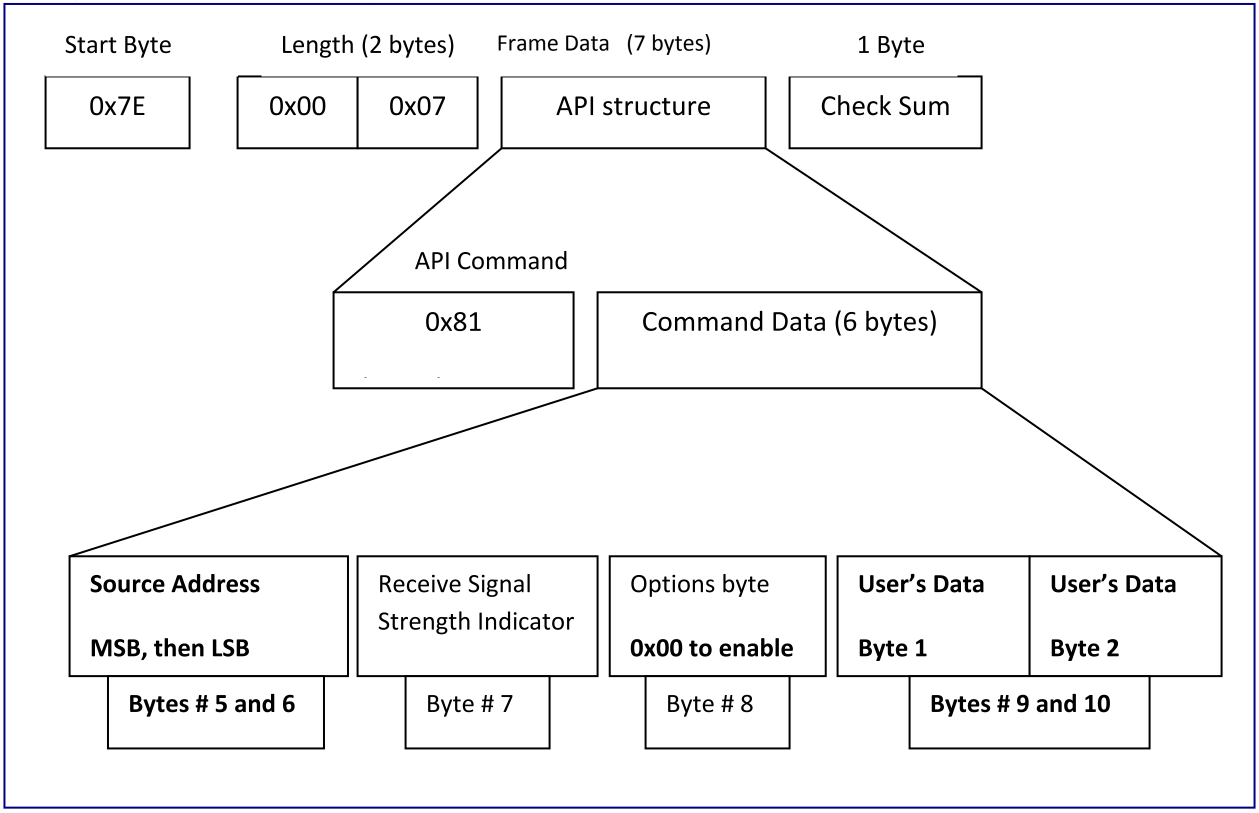
In general, RF Data can be up to 100 bytes long, but in our case, the RF data is only 2 bytes long. Thus, ME218C users' received data resides in bytes 9 and 10 of each message. Byte 9 contains the command/message "opcode", and byte 10 contains a parameter associated with that command/message (if any).

The standard message byte table for a receive packet therefore looks like this:

Byte Number:	Description:	Value:
Byte 1)	Start Delimiter	0x7E
Byte 2)	Length of Frame 1	0x00
Byte 3)	Length of Frame 2	0x07
Byte 4)	Receive API Identifier	0x81
Byte 5)	Source Address 1	See Addressing
Byte 6)	Source Address 2	See Addressing
Byte 7)	RSSI	See XBee Doc
Byte 8)	Options	See XBee Doc
Byte 9)	Message Byte (Byte 1)	See RF Data
Byte 10)	Level Byte (Byte 2)	See RF Data
Byte 11)	Check Sum	0xFF – (Sum of Bytes 4-10)*

*Note: on the receive side, the use of this is to check that (Sum of Bytes 4-10) + CheckSum = 0xFF.

In diagram form, this can be represented like this:



1.2 Addressing

Each team has been provided with a pair of XBee radios, one for their TOWRP and one for their COACH. The upper byte on the pre-programmed address indicates which type of device the radio is for, 0xAF for the TOWRP and 0xBC for the COACH. The lower byte corresponds to the team's team number. **Teams should update their team number on the Team Formation Wiki to reflect the radio pair assigned to them, and use this number as the required label for their TOWRP.** This allows this protocol to never have to allow for team number to be passed as a parameter, even when it is needed for display on the COACH, since it can always be extracted from the XBee packet (bytes 5 and 6).

Anytime a message is received by a microprocessor connected to an XBee radio, you can be sure that the message was addressed to you (or broadcast); however, some error checking on the sender's address (again, available in the packet header) must still be done, and is the responsibility of each team. As a first pass, TOWRPs should only be receiving messages from COACHs and COACHs should only be receiving messages from TOWRPs. Additionally, if a TOWRP and COACH are currently paired with each other (see Conversations and Procedures for more on the term "Paired"), then they should only accept commands from each other. In this case, each device (TOWRP or COACH) will have to store the team number of the COACH/TOWRP with which it is currently paired.

1.3 RF Data

All transmissions will contain two bytes of ME218C data: one "opcode" byte (which command or message it is), followed by one "parameter" byte.

1.3.1 Opcode Byte

The first ME218C byte, the “opcode” byte, represents the message or command. These can either be directives from the COACH to the TOWRP or status updates from the TOWRP to the COACH. Each message has a unique value, or “opcode”, associated with it (see Available Commands and Messages). The “opcode” byte is the first user-sent byte in the XBee frame (see the XBee section).

1.3.2 Parameter Byte

The second ME218C byte, the “parameter” byte, represents a magnitude of the command or status. This allows for, among other things, the COACH to send varying magnitudes for movement commands to the TOWRP, and the TOWRP to send its morale status to the COACH. The parameter byte also enables the TOWRP to send status messages to the COACH, relaying important data to the individual piloting the TOWRP. If a parameter is not needed for the given opcode, any parameter byte may be sent and ignored by the recipient – this ensures all messages are two bytes long.

All movement commands have 8 possible levels. These 8 levels have been broken down into convenient percentage-of-max values – 0% (stop), 15%, 29%, 45%, 59%, 75%, 89%, 100%. Only 8 levels are provided so that a fourth sign bit can indicate direction (forward/backward and left/right), and so that speed (longitudinal velocity) and direction (rotational velocity) can compose the upper and lower nibble, respectively, of a single parameter byte. More information, including sign conventions, can be found in the Available Commands and Messages section.

2 Communications Protocol

2.1 Ground Rules

Only one command/message can be sent per XBee packet. Thus, each XBee packet should be 11 bytes long total, including the checksum byte. Packets cannot be sent at a rate greater than 5 Hz. It is the responsibility of each COACH/TOWRP to time this internally and not send messages too quickly. All communications will be made at 9600 baud.

When parameters are signed numbers, the standard is to use 2's complement representation. This way, those programming in C will see the numbers properly and the refresh rate is as fast as possible. (Note that the parameter for the velocity command is not strictly a number, but rather a more complex construction, and so does not abide by this rule; this rule applies unless otherwise specified, however.)

2.2 Available Commands and Messages

From COACH to TOWRP

Command	*Opcode*	*Parameter*	*Context*	*Broadcast*	*Notes*
Suppress Mutiny	0x00	None (Anything)	Superstate "Paired"; used in response to a "Mutiny in Progress"; expected response is an updated morale value of -5; resend/time-out after 1/5 seconds	No	Can only be sent after the COACH-specific series of actions has been completed successfully; the TOWRP's morale value will immediately change to -5, which will be indicated in the "Update Morale" response sent by the TOWRP.
Assertion of Command	0x01	COACH's Team Color (0 = Red, 1 = Blue)	Superstate "Not Paired"; used in response to "I Am Available"; expected response is "Assertion of Command Response"; resend/time-out after 1/3 seconds	No	
Jettison the Crew	0x02	None (Anything)	Superstate "Paired"	No	Causes a transition to superstate "Not Paired" and resets TOWRP's morale value to be zero.

Velocity	0x03	Upper Nibble: Percent of Maximum Forward Velocity Desired (Positive is Forward);
		Lower Nibble: Percent of Maximum Rotational Velocity Desired (Positive is Counterclockwise);
		In each nibble, the MSB is a sign bit, 0 for positive, 1 for negative. The other three bits form a code for magnitude: 000 = 0% (stop), 001 = 15%, 010 = 29%, 011 = 45%, 100 = 59%, 101 = 75%, 110 = 89%, 111 = 100%
			Superstate "Paired"; then must be sent at least once per second; expected response is an updated morale value; resend/time-out after 1/5 seconds	No	Velocities are valid for one second; after one second, if a new velocity command has not been received, the TOWRP should default to 0% of maximum velocity, both forward and rotational.

Free Digital	0x04	None (Anything)	Superstate "Paired"	No	COACH must provide a means for triggering this command.
Free Analog	0x05	Magnitude (-100% to +100%)	Superstate "Paired"	No	COACH must provide an analog input for this command, in addition to a means for triggering the command.

From TOWRP to COACH

Command	*Opcode*	*Parameter*	*Context*	*Broadcast*	*Notes*
Mutiny in Progress	0x20	None (Anything)	Superstate "Paired"; then must be sent if morale reaches mutiny level; resend/time-out after 0.2/3 seconds	No	If a Suppress Mutiny command is not received soon enough, TOWRP transitions to superstate "Not Paired"
I Am Available	0x21	None (Anything)	Superstate "Not Paired"; then must be sent at least once per second; expected response of “Assertion of Command”; resend after 1 seconds	Yes	Morale stays constant at zero while broadcasting this, so that the initial value for a new COACH will always be zero.
Assertion of Command Result	0x22	New Controlling COACH (network address)	Superstate "Not Paired"	Yes	Leads to a transition to superstate "Paired"
Update Morale	0x23	Current Morale Value	Superstate "Paired"; then only in response to a Velocity or Suppress Mutiny command.	No	Used to acknowledge receipt of both Velocity and Suppress Mutiny commands.

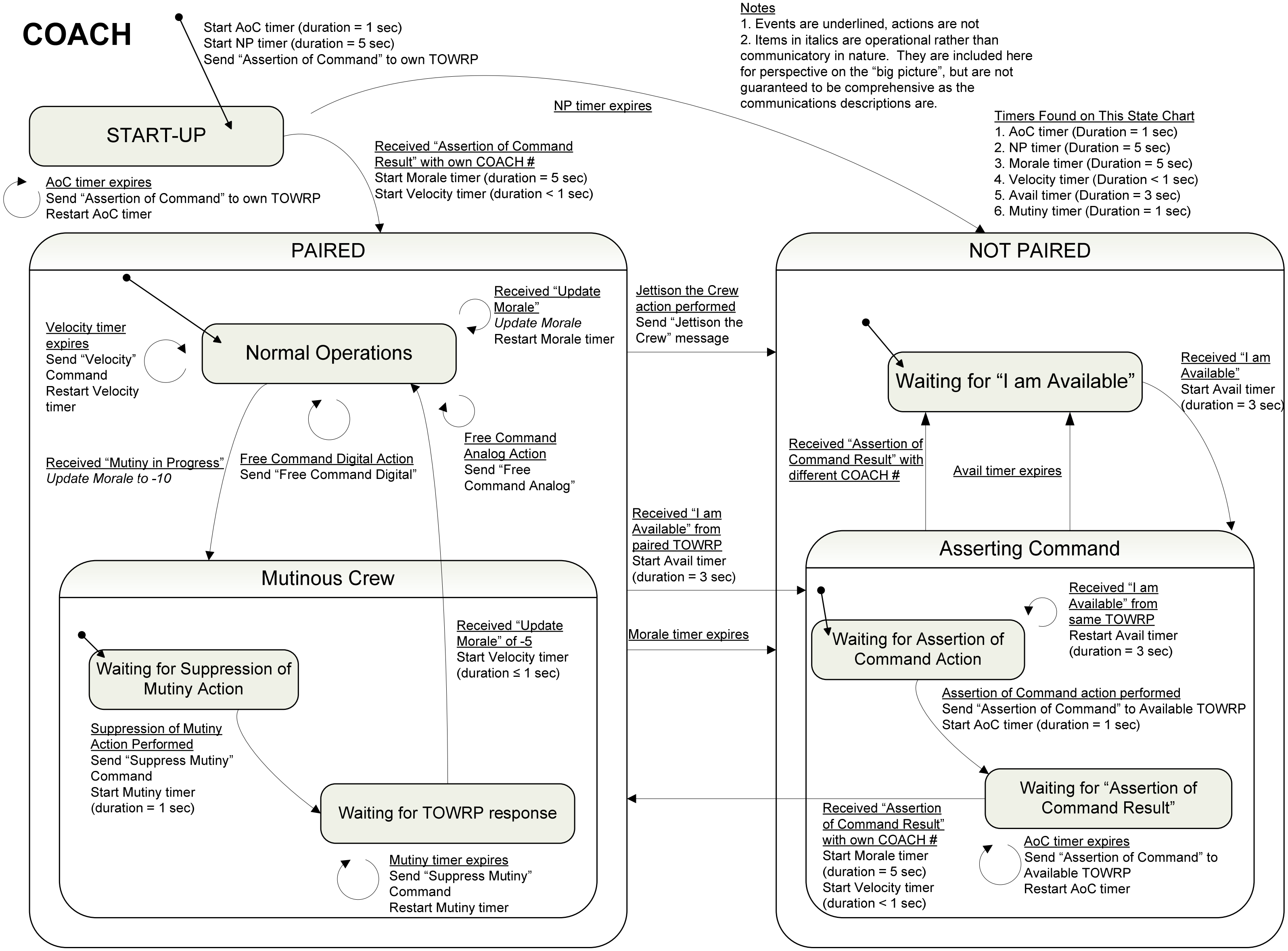
2.3 Conversations and Procedures

Both the TOWRP's and the COACH's communications can be thought of as a two-level hierarchical state chart. Both devices' superstates include "Paired" (with a COACH/TOWRP) and "Not Paired" (with a COACH/TOWRP); the COACH's state chart also has a third superstate, "Startup". Transitions from "Not Paired" to "Paired" are achieved through the "Assertion of Command" conversation, which consists of a series of messages from the tables above (see below for more on individual conversations). Transitions from "Paired" to "Not Paired" occur automatically after five seconds a) for the TOWRP if it has not received any velocity commands, and b) for the COACH if it has not received any update morale messages. The "Paired" superstate is dominated by the COACH; for the most part, only the COACH can initiate conversations in this superstate (the exception being the case of mutiny); the "Not Paired" superstate is dominated by the TOWRP; only the TOWRP can initiate conversations in this superstate. In the "Paired" superstate, the COACH is required to send velocity commands at least once every second. In the "Not Paired" superstate, the TOWRP is required to send "I Am Available" messages at least once every second.

Upon reset (and therefore startup), the TOWRP awakes into the "Not Paired" superstate, while the COACH awakes into the "Startup" superstate. (This means that the TOWRP should be turned on before the COACH.) In the "Startup" superstate, the COACH is required to send an "Assertion of Command" to its own team's TOWRP (hard-coded) at least once every second. (Note that when used in the "Not Paired" superstate, "Assertion of Command" may only be sent in response to an "I Am Available"; during "Startup" however, this restriction is lifted and in fact the COACH must send "Assertion of Command".) If the COACH successfully completes the "Assertion of Command" conversation with its TOWRP within five seconds of reset, it transitions to the "Paired" superstate; otherwise, it transitions to the "Not Paired" superstate. Thus, the determination of which COACH in a given match begins "marauding" is accomplished strictly via the startup protocol. Once communications are established with a TOWRP, however, this protocol does nothing to enforce any waiting for the game play to start; it is up to the human operator to only send velocity commands with zero magnitude until allowed to do otherwise.

Various conversations (diagrammed below) can be constructed from the individual commands and messages listed in the tables above. However, as the tables indicate, only certain commands and messages are available in a given superstate. (For example, the COACH cannot send an "Assertion of Command" message while in the "Paired" superstate.) If packets are lost, it is quite possible for entire conversations to be corrupted, leading to temporary discrepancies between superstates. (For example, if a COACH never receives the "Mutiny in Progress" message, it will think it is still "Paired" with its TOWRP when in fact it is not.) Nothing is done to prevent this; however, the automatic transitions from "Paired" to "Not Paired" after five seconds will ensure that any such discrepancies are resolved by that time. (In the case of the COACH being unaware of the mutiny, it would stop receiving morale updates from the TOWRP since the TOWRP is ignoring its velocity commands, and, after five seconds, would conclude it must be "Not Paired".)

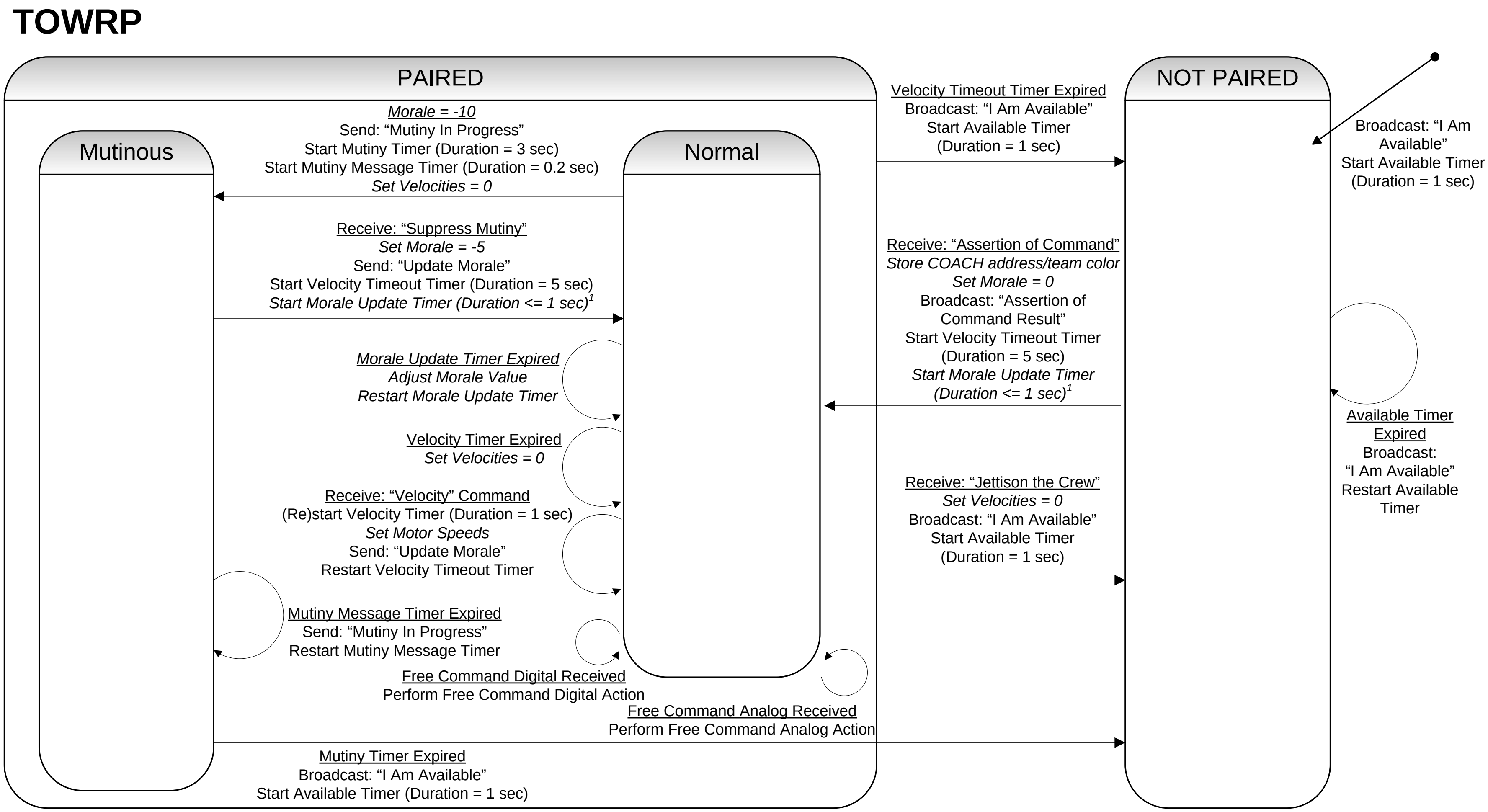
2.3.1 Communications State Chart for the COACH


COACH
Start AoC timer (duration = 1 sec)
Start NP timer (duration = 5 sec)
Send "Assertion of Command" to own TOWRP
START-UP
AoC timer expires
Send "Assertion of Command" to own TOWRP
Restart AoC timer
NP timer expires
Received "Assertion of Command Result" with own COACH #
Start Morale timer (duration = 5 sec)
Start Velocity timer (duration < 1 sec)
PAIRED
Normal Operations
Velocity timer expires
Send "Velocity" Command
Restart Velocity timer
Received "Update Morale"
Update Morale
Restart Morale timer
Received "Mutiny in Progress"
Update Morale to -10
Free Command Digital Action
Send "Free Command Digital"
Free Command Analog Action
Send "Free Command Analog"
Mutinous Crew
Waiting for Suppression of Mutiny Action
Suppression of Mutiny Action Performed
Send "Suppress Mutiny" Command
Start Mutiny timer (duration = 1 sec)
Waiting for TOWRP response
Mutiny timer expires
Send "Suppress Mutiny" Command
Restart Mutiny timer
Received "Update Morale" of -5
Start Velocity timer (duration ≤ 1 sec)
Jettison the Crew action performed
Send "Jettison the Crew" message
Received "I am Available" from paired TOWRP
Start Avail timer (duration = 3 sec)
Morale timer expires
NOT PAIRED
Waiting for "I am Available"
Received "I am Available"
Start Avail timer (duration = 3 sec)
Received "Assertion of Command Result" with different COACH #
Avail timer expires
Asserting Command
Waiting for Assertion of Command Action
Received "I am Available" from same TOWRP
Restart Avail timer (duration = 3 sec)
Assertion of Command action performed
Send "Assertion of Command" to Available TOWRP
Start AoC timer (duration = 1 sec)
Waiting for "Assertion of Command Result"
Received "Assertion of Command Result" with own COACH #
Start Morale timer (duration = 5 sec)
Start Velocity timer (duration < 1 sec)
AoC timer expires
Send "Assertion of Command" to Available TOWRP
Restart AoC timer

Notes
1. Events are underlined, actions are not
2. Items in italics are operational rather than communicatory in nature. They are included here for perspective on the "big picture", but are not guaranteed to be comprehensive as the communications descriptions are.

Timers Found on This State Chart
1. AoC timer (Duration = 1 sec)
2. NP timer (Duration = 5 sec)
3. Morale timer (Duration = 5 sec)
4. Velocity timer (Duration < 1 sec)
5. Avail timer (Duration = 3 sec)
6. Mutiny timer (Duration = 1 sec)

2.3.2 Communications State Chart for the TOWRP


TOWRP
PAIRED
NOT PAIRED
Mutinous
Normal
Morale = -10
Send: "Mutiny In Progress"
Start Mutiny Timer (Duration = 3 sec)
Start Mutiny Message Timer (Duration = 0.2 sec)
Set Velocities = 0
Receive: "Suppress Mutiny"
Set Morale = -5
Send: "Update Morale"
Start Velocity Timeout Timer (Duration = 5 sec)
Start Morale Update Timer (Duration <= 1 sec)1
Morale Update Timer Expired
Adjust Morale Value
Restart Morale Update Timer
Velocity Timer Expired
Set Velocities = 0
Receive: "Velocity" Command
(Re)start Velocity Timer (Duration = 1 sec)
Set Motor Speeds
Send: "Update Morale"
Restart Velocity Timeout Timer
Mutiny Message Timer Expired
Send: "Mutiny In Progress"
Restart Mutiny Message Timer
Free Command Digital Received
Perform Free Command Digital Action
Free Command Analog Received
Perform Free Command Analog Action
Mutiny Timer Expired
Broadcast: "I Am Available"
Start Available Timer (Duration = 1 sec)
Velocity Timeout Timer Expired
Broadcast: "I Am Available"
Start Available Timer
(Duration = 1 sec)
Receive: "Assertion of Command"
Store COACH address/team color
Set Morale = 0
Broadcast: "Assertion of Command Result"
Start Velocity Timeout Timer
(Duration = 5 sec)
Start Morale Update Timer
(Duration <= 1 sec)1
Receive: "Jettison the Crew"
Set Velocities = 0
Broadcast: "I Am Available"
Start Available Timer
(Duration = 1 sec)
Broadcast: "I Am Available"
Start Available Timer
(Duration = 1 sec)
Available Timer Expired
Broadcast:
"I Am Available"
Restart Available Timer

Notes
1. Morale must be updated at least once per second, however, it can be updated more frequently if desired.
2. Events are underlined, actions are not
3. Items in italics are operational rather than communicatory in nature. They are included here for perspective on the "big picture", but are not guaranteed to be comprehensive as the communications descriptions are.

Timers Found on This State Chart
1. Mutiny Timer (Duration = 3 sec)
2. Mutiny Message Timer (Duration = 0.2 sec)
3. Velocity Timer (Duration = 1 sec)
4. Velocity Timeout Timer (Duration = 5 sec)
5. Morale Update Timer (Duration <= 1 sec)
6. Available Timer (Duration = 1 sec)

3 Contact Information

For any questions about the communications protocol, please contact the member of the Communications Committee specifically responsible for that section. Clarifications requested before Tuesday, May 12 may be able to be amended to this document before the final draft is due.

Overall/General Communications: David Hoffert, hoffert@stanford.edu
Packet Structure: Egan Gans, egangans@stanford.edu
COACH to TOWRP Commands: Nathan Fenner, nfenner@stanford.edu
TOWRP to COACH Commands: Michael McDaniels, mmcdan@stanford.edu
Superstates: Ruddick Lawrence, ruddick@stanford.edu
Mutiny Conversation: Matt Montgomery, matt123@stanford.edu
Assertion of Command Conversation: Ellis Garai, egarai@stanford.edu
Jettison Conversation: Joey Darragh, darragjm@stanford.edu
Velocity/Morale Conversation: Sam Early, saearly@stanford.edu
Free Command Conversations: Peling Lee, lings@stanford.edu