StarL Interface Control Document

Last updated 5/3/12

Position system:

A PC will be connected to the OptiTrack system, using a MATLAB program to acquire the locations of each robot. This PC will be connected to IllinoisNet and will have a list of each robot's IP address. This PC will periodically broadcast the locations of each robot to all robots on port 4000. This will be done with an ASCII UDP message containing entries with the following format:

Line header	Line ID	X Position	Y Position	Yaw	End of line

The fields will be delineated by the pipe character: "|"

Line header: This is a single character header describing the type of entity being tracked in this line. "#" indicates a robot, "@" indicates a waypoint.

Line ID: This is the name of the entity, for example "Carlos", or "Diana". Without this line, the robots receiving the message wouldn't be able to determine their own position from the list of positions. In the case of waypoints, the waypoint name and waypoint room are combined, separated by a comma.

X Position: The X position, in millimeters, of the entity from the coordinate system origin.

Y Position: The Y position, in millimeters, of the entity from the coordinate system origin.

Yaw: The rotation of the entity, in degrees, from its initial state.

End of line: A single character, "&" followed by either a new line character "\n" indicating the end of the entry.

An example position message:

#|Carlos|1523|1200|-15.5|&

#|Diana|140|90|125|&

@|Waypoint1,A|500|500|0|&

@|Waypoint2,B|1000|1000|0|&

W|Wall1|

Defining walls and waypoints:

A WPT file is parsed by the MATLAB program or StarL simulator. This file contains the endpoints of walls and the locations and names of waypoints to be sent to the robots. Walls are only used by the MATLAB display, they are not sent to the robots and aren't involved in any calculations/path finding (currently). In fact, as of 5/3/12, walls haven't been used and will likely be removed from the WPT specification. Names of waypoints MUST BE UNIQUE. Sending two waypoints with a duplicate name will cause the first waypoint received to be overwritten with the second. The WPT file is an ASCII file with the following format:

Comments: Any line beginning with "%" is ignored and treated as a comment
Walls: "WALL,X0,Y0,X1,Y1", where X# and Y# are the coordinates of the wall ends.
Waypoint: "WAY,X,Y,A,NAME", where X and Y are the local coordinates of the waypoint, A is the angle it faces, NAME is its name.
Each line ends with a new line character "\n"

Robot message passing:

Messages are sent from robot to robot via an acknowledging UDP protocol on port 4001. The format of a message is as follows:

Header	Sequence Number	From	To	Message ID	Contents	End

Fields are delineated with the pipe character, "|".

Header: The character "M".

Sequence Number: A unique integer message identifier. Initialized to a random number between 0 and Integer.MAX_VALUE for each robot, incremented by 1 with each message sent.

From: The name of the sending robot

To: The name of the receiving robot

Message ID: An integer used to identify the contents and destination thread of the message, for example, MID 0 = leader election, MID 1 = collision alert, etc.

Contents: The contents of the message, may be blank.

End: The character "&".

To demonstrate, assume two robots exist, **RA** and **RB**. Suppose **RA** sends a message **M** to **RB**. Upon receiving the **M**, **RB** must acknowledge receipt by sending an ACK message (detailed below) back to **RA**. If an ACK message is not received by **RA** with 250 milliseconds, the **M** is sent again.

If the original **M** was lost in transmission, it will be received by **RB** in a subsequent transmission. If the original ACK from **RB** to **RA** was lost, **RB** will know to resend the ACK when a duplicate **M** is received from **RA**. This is accomplished by storing all received messages in a queue. If a received message is a duplicate, assume that the ACK was lost and flag it to be resent. Duplicate ACKs are simply ignored. Messages older than 20 seconds are removed from the queue.

The ACK message has the same sequence number, from, and to fields as the original message. All ACK messages have message ID 0. The contents of ACK messages are ignored, some versions of the StarL framework use contents "ACK", but the latest version leaves these messages blank. All messages with ID 0 will be treated as ACK messages. This is the only reserved message ID.

Example messages:

M|777|Alice|Bob|55|Hello|&
M|777|Alice|Bob|0|ACK|&

M|123|Carlos|Diana|1||&
M|123|Carlos|Diana|0|ACK|&