Teams 19 & 20: Communications Protocol Specification Document

Laboratory #4: Development Tools and Communications Protocol

Morgan, Laura	Lenig, Tyler
Miaw, Jireh	Tang, Raymond
Hauser, Steven	Rupakhetee, Archit
Dworak, Catherine	McMillion, Andrew
Bertoglio, David	

Work Product

This document describes the communication protocol implemented by Teams 19 and 20 for communication between the base station control system, and the robot. This document describes the creation, and decoding process for messages.

Document Revision Information

2/15/2013 created
2/17/2013 designed base station to robot messages
2/22/2013 continued design
2/24/2013 designed robot to base station messages
3/17/2013 added commands and error detection

Approval Sheet

All group members whose names are listed below approve of the document and contributed fairly.

Member Names
Group 19 representative
Dworak, Catherine

Group 20 representative
Lenig, Tyler

Pledge

On my honor, as a student, I have neither given nor received unauthorized aid on this assignment.

Names
Group 19
Morgan, Laura
Miaw, Jireh
Hauser, Steven
Dworak, Catherine
Bertoglio, David

Group 20
Lenig, Tyler
Tang, Raymond
Rupakhetee, Archit
McMillion, Andrew

Table of Contents

Introduction

About

This document describes the protocol used to communicate between robot and base station system. This protocol allows the base station to control the robot, and allows the robot to send messages including errors to the base station.

Protocol Description

This protocol uses 11-character messages to communicate between the robot and the base station. The first 10-character of messages encode both commands from the base station to the robot and messages from the robot to the base station. The messages are structured such that the first two characters determine the type of command or message. The remaining characters are used for various parameters that are documented below. The 11th character holds the checksum used for error detection.

109 # Base Station to Robot Messages

110 ## Command Structure
111 Commands are 10-character messages, where the first two characters are the
112 command type. The remaining characters represent parameters to the command,
113 used by the robot to determine how to execute the command.

114 ## No-Op
115 **Message:** 0000000000
116 **Description:** This command is the no operation command can be used to test if
117 messages are being sent. This message is a "null" message.

118 ## Move Straight
119 **Command Type:** MS
120 **Parameters:** Forward/Backwards, and distance.
121   Character 2 is forward or backwards (F/B)
122   Characters 3-9 are distance (#), can be null (0s)
123 **Description:** This command moves the robot in a straight line. The
124 forward/backward parameter control the direction the robot will move in. The
125 distance allows for the robot to move a specified distance, this parameter can be
126 null. If distance is null, the robot will continually move
127 **Example Commands:**
128   MSF0000000 will move the robot forward continuously.
129   MSB0001000 will move the robot backwards 1000 units.

130 ## Move Arc
131 **Command Type:** MA
132 **Parameters:** Forward/Backwards, left/right, radius, distance
133   Character 2 is forward or backwards (F/B)
134   Character 3 is left or right (L/R)
135   Characters 4-6 are radius (# degrees)
136   Characters 7-9 are distance (#), can be null (0s)
137 **Description:** This command moves the robot in an arc. The forward/backward
138 parameter control the direction the robot will move along the arc. Left/Right will
139 control the direction the robot arcs to. Radius is the absolute value of the number of
140 degrees to move. The distance allows for the robot to move a specified distance, this
141 parameter can be null. If distance is null, the robot will continually move until
142 stopped.
143 **Example Commands:**
144   MAFL090000 will move the robot forward to the left along a 90 degree curve
145 continuously
146   MABR030100 will move the robot backwards along a 30 degree curve for
147 100 units.
148

149   **Turn**

150   **Command Type:** TN

151   **Parameters:** Left/Right, and radius

152   Character 2 is left or right (L/R)

153   Characters 3-9 are radius (# degrees), can be null (0s)

154   **Description:** This command turns the robot when stationary. The Left/Right

155   parameter determines the direction the robot turns. The Radius parameter is an

156   absolute value that determines how far the robot turns. If the radius is null, the

157   robot continually turns until stopped.

158   **Example Commands:**

159   TNR0000090 will turn the robot right 90 degrees

160   TNL0000000 will turn the robot left continuously

161   **Stop**

162   **Message:** ST00000000

163   **Description:** This command stops any actions that the robot is currently doing. This

164   will end any movement actions.

165   **Read Sensor**

166   **Command Type:** RS

167   **Parameters:** Sensor Port

168   Character 2 is sensor type (U for Ultrasonic, T for touch, M for sound, L for

169   light)

170   Characters 3-9 are 0

171   **Description:** This command will read a specified sensor. The Sensor Port parameter

172   will determine which sensor to read the value of.

173   **Example Commands:**

174   RSU0000000 will cause the robot to read the value of the sensor, and send

175   the data to the base station.

176   **Set Speed**

177   **Command Type:** SS

178   **Parameters:** Motor/Motor Combination, and new speed.

179   Character 2 is Motor/Motor combination (A for Motor A, B for Motor B, C for

180   Motor C, D for Drive Motors)

181   Characters 3-9 are the new speed

182   **Description:** This command will change the speed of the motors. The combination

183   will determine which motors or combinations of motors to change the speed for.

184   **Example Commands:**

185

186   **Read All Sensors**

187   **Command Message:** RA00000000

188   **Description:** This command tells the robot to read all sensors and send the data.

189   Each sensor's data will be sent to the base station in a separate message.

190   **End Connection**

191   **Command Message:** EC0000000

192 **Description:** This command instructs the robot to end connection with the base
193 station.

194 ## Robot to Base Station Messages

195 ### Acknowledgment
196 **Description:** This message is sent to the base station as acknowledgment of
197 receiving a command.
198 **Message:** AK00000000

199 ### Error Messages

200 ### Sensor Error Messages
201 **Message Type:** ERS
202 **Parameters:** Message number
203   Characters 3-9 are message number
204 **Description:** This message will tell the base station that an error with a sensor has
205 occurred. The message number maps to a more specific description, that the base
206 station will have stored locally for reference. Available messages can be seen in a
207 table below, which will have additions added as required.
208

Message Number	Description
0000001	Error with sensor in port 1
0000002	Error with sensor in port 2
0000003	Error with sensor in port 3
0000004	Error with sensor in port 4

209

210 ### Motor Error Messages
211 **Message Type:** ERM
212 **Parameters:** Message Number
213   Characters 3-9 are message number
214 **Description:** These messages will tell the base station that an error with a motor
215 has occurred. The message number correlates to a specific description, which the
216 base station has stored locally. Available messages can be seen in a table below,
217 which will have additions added as required.
218

Message Number	Description
0000001	Error with motor in port A
0000002	Error with motor in port B
0000003	Error with motor in port C

219

220 ### Sensor Data Messages
221 **Message Type:** SD
222 **Parameters:** Sensor Type, and Data

223   Character 2 is sensor type (U for Ultrasonic, T for Touch, M for Sound, or L for
224 Light)
225   Characters 3-9 sensor data
226 **Description:** These messages allow for the robot to send data to the base station
227 based on the values of the sensor.

228 Error Detection

229 Introduction

230 The Error detection this protocol utilizes is a checksum for detecting errors in
231 packets, and a timeout on acknowledgments. The checksum is calculated using only
232 the first 10 characters of the message, then checked against the character that is
233 sent in the packet. The timeouts will be 10 seconds before the sender will assume
234 the packet was lost and needs to be retransmitted.

235 Checksum Function

236 The function for calculating the checksum is
237 $(checksum = \sum_{i}^{10}(byte)message[i]) \; mod \; 256$. This function allows for no matter
238 the value of the checksum it will fit in a one-byte character. The function is the sum
239 of the byte value of each character in the message modulo 256.