Feb 8, 2013

Team 19: Specification Document

**Laboratory 1: Risk Reduction Techniques**

**Morgan, Laura**

**Miaw, Jireh**

**Hauser, Steven**

**Dworak, Catherine**

**Bertoglio, David**

***Work Product***

**The Specification Document describes the behavior of the robot control system. It includes the glossary, mode definition, mode transition table, conditions, input and output data items, and event table.**

***Document Revision Information***

**Document Revision Information**

2/10/2013 – Template created

2/15/2013 – Mode definitions and events

2/18/2013 – Input and output data items

2/22/2013 – Glossary and event table

2/24/2013 - Completed

**Approval Sheet**

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

**Morgan, Laura**

**Miaw, Jireh**

**Hauser, Steven**

**Dworak, Catherine**

**Bertoglio, David**

**Pledge**

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

**Morgan, Laura**

**Miaw, Jireh**

**Hauser, Steven**

**Dworak, Catherine**

**Bertoglio, David**

Table of Contents

[Glossary 4](#_Toc349478321)

[Text Macros 4](#_Toc349478322)

[Input Data Items 4](#_Toc349478323)

[Output Data Items 5](#_Toc349478324)

[Conditions 5](#_Toc349478325)

[Set of Modes 6](#_Toc349478326)

[Mode Transition Table 6](#_Toc349478327)

[Define User Interface 7](#_Toc349478328)

Glossary  
  
Symbolic Constants

|  |  |  |
| --- | --- | --- |
| Name | Definition | Value |
| $max\_speed$ | max speed of motors | TO BE DETERMINED |
| $NoOp$ | no operation is taken by GUI | Null |
| $pressed$ | button on GUI is pressed down | True |
| $released$ | butoon on GUI was pressed and has been released | True |
| $arc\_radius$ | radius of arc taken by robot when multiple buttons pressed | TO BE DETERMINED |

## Text Macros

|  |  |
| --- | --- |
| Name | Definition |
| !connection! | connection between the robot and base station |
| !error\_message\_table! | listing of all error messages to error code |
| !reading! | decoded /input\_message/ to be displayed |
| !response! | message sent from robot to base station |

### Input Data Items

|  |  |
| --- | --- |
| Name | Definition |
| /button\_backwad/ | controls backward movement |
| /button\_forward/ | controls forward movment |
| /button\_left/ | controls movement left |
| /button\_right/ | controls  movement right |
| /button\_sensor\_light/ | displays light sensor information |
| /button\_sensor\_unltrasonic/ | displays ultrasonic sensor information |
| /input\_speed/ | input for new speed |
| /button\_get\_connection/ | get connection |
| /button\_end\_connection/ | end connection |
| /button\_change\_speed/ | changes speed of robot |
| /input\_message/ | message received from robot |

### Output Data Items

|  |  |
| --- | --- |
| Name | Definition |
| //data\_log// | display for messages from robot |
| //sensor\_light// | display for light sensor |
| //sensor\_touch// | display for touch sensor |
| //sensor\_sound// | display for sound sensor |
| //sensor\_ultrasonic// | display for ultrasonic sensor |
| //output message// | message sent to robot |

## Conditions

|  |  |
| --- | --- |
| Name | Definition |
| %connection\_received% | Whether a connection is created or not. |
| %get\_connection% | /button\_get\_connection/ = $released$ |
| %message\_recieved% | A message |
| %time\_out% | 10 seconds no response |
| %connected% | !connection! = True |
| %end\_connection% | /button\_end\_connection/ = $released$ |
| %is\_error\_message% | Whether /input\_message/ is error |

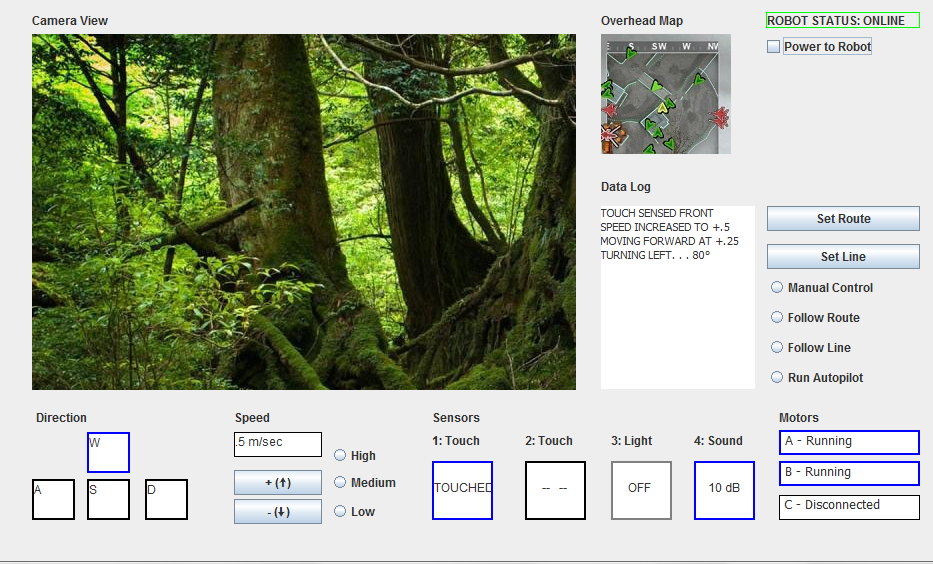
# Set of Modes

|  |  |
| --- | --- |
| Name | Definition |
| \*Normal Operation\* | %connected% |
| \*Awaiting Connection\* | !connection! = false /button\_get\_connection/ = $pressed$ |
| \*No Connection\* | !connection! = false |

# Mode Transition Table

|  |  |  |  |
| --- | --- | --- | --- |
|  | \*Normal Operation\* | \*Awaiting Connection\* | \*No Connection\* |
| \*Normal Operation\* |  | @T(%connection\_received%) | @T(%end\_connection%) |
| \*Awaiting Connection\* | @T(%connection\_received) |  | @T(%time\_out%) |
| \*No Connection\* |  | @T(%get\_connection%) |  |

Define User Interface

Figure 1. Robot GUI  
  
  
Inputs and Outputs  
  
Inputs  
  
Input data item:  Forward push button  
Acronym: /button\_forward/  
Hardware:  Switch, normally open  
Description: /button\_forward/  
            - controls forward movement

- while pressed move forward, when released stop

Input data item:  Backward push button  
Acronym: /button\_backward/  
Hardware:  Switch, normally open  
Description: /button\_backward/  
            - controls backwards movement

- while pressed move backward, when released stop

Input data item:  Right push button  
Acronym: /button\_right/  
Hardware:  Switch, normally open  
Description: /button\_right/  
            - controls movement right          
  
Input data item:  Left push button  
Acronym: /button\_left/  
Hardware:  Switch, normally open  
Description: /button\_left/  
            - controls movement left            
  
Input data item:  Light sensor button  
Acronym: /button\_sensor\_light/  
Hardware:  Switch, normally open  
Description: /button\_sensor\_light/  
            -data from light sensor will be displayed in light sensor display  
  
Input data item:  Ultrasonic sensor button  
Acronym: /button\_sensor\_ultrasonic/  
Hardware:  Switch, normally open  
Description: /button\_sensor\_ultrasonic/  
            -data from ultrasonic sensor will be displayed in ultrasonic sensor display  
  
Input data item:  Speed input  
Acronym: /input\_speed/  
Hardware:  Switch, normally open  
Description: /input\_speed/  
            -receive keyboard input of numbers to change speed  
  
Input data item:  Change speed button  
Acronym: /button\_change\_speed/  
Hardware:  Momentary switch, normally open  
Description: /button\_change\_speed/  
            -change robot speed to speed currently in /input\_speed/  
Data Representation:

Byte 3 is Motor/Motor combinations

Bytes 4-9 is the new speed

Input data item:  Get connection button  
Acronym: /button\_get\_connection/  
Hardware:  Momentary switch, normally open  
Description: /button\_get\_connection/  
            -transitions from \*No Connection\* to \*Awaiting Connection\*  
  
Input data item:  End connection button  
Acronym: /button\_end\_connection/  
Hardware:  Momentary switch, normally open  
Description: /button\_end\_connection/  
            -transitions from \*Normal Operation\* to \*No Connection\*  
  
Input data item:  Message received from robot  
Acronym: /input\_message/  
Hardware:  Communications link (bluetooth)  
Description: /input\_message/  
            -message sent from the robot  
  
Output  
  
Output data item: Message sent to the robot  
Acronym: //output\_message//  
Hardware: Communications link (bluetooth)  
Description: //output\_message//  
 - encodes commands for the robot to complete based on user input  
Characteristic of values: encoded based on communication specification; 10 character message  
  
Output data item: Light sensor output  
Acronym: //sensor\_light//  
Hardware: LCD monitor  
Description: //sensor\_light//  
 - display most recently read value from light sensor  
Characteristic of values: Strings  
  
Output data item: Sound sensor output  
Acronym: //sensor\_sound//  
Hardware: LCD monitor  
Description: //sensor\_sound//  
 - display most recently read value from sound sensor  
Characteristic of values: Strings  
  
Output data item: Touch sensor output  
Acronym: //sensor\_touch//  
Hardware: LCD monitor  
Description: //sensor\_touch//  
 - display most recently read value from touch sensor  
Characteristic of values: Strings  
  
Output data item: Ultrasonic sensor output  
Acronym: //sensor\_ultrasonic//  
Hardware: LCD monitor  
Description: //sensor\_ultasonic//  
 - display most recently read value from ultrasonic sensor  
Characteristic of values: Strings  
  
Output data item: Display for messages from robot  
Acronym: //data\_log//  
Hardware: LCD monitor  
Description: //data\_log//  
 - displays messages from robot  
 - displays error message from robot  
Characteristic of values: Strings/sentences in textbox  
  
Define Set of Events

|  |  |  |  |
| --- | --- | --- | --- |
| Mode | Event   |  | | --- | |  | | Action |
| \*Normal Operation\* | @T(/button\_forward/ = $pressed$)   |  | | --- | |  | | //output\_message// = “MSF0000000” is sent |
|  | @T(/button\_backward/ = $pressed$) | //output\_message// = “MSB0000000” |
|  | @T(/button\_left/ = $pressed$) | //output\_message// = “TNL0000000” |
|  | @T(/button\_right/ = $pressed$) | //output\_message// = “TNR0000000” |
|  | @T(/button\_left/ = $pressed$ AND /button\_forward/ = $pressed$) | //output\_message// = “MAFL000000” |
|  | @T(/button\_right/ = $pressed$ AND /button\_forward/ = $pressed$) | //output\_message// = “MAFR000000” |
|  | @T(/button\_left/ = $pressed$ AND /button\_backward/ = $pressed$) | //output\_message// = “MABL000000” |
|  | @T(/button\_right/ = $pressed$ AND /button\_backward/ = $pressed$) | //output\_message// = “MABR000000” |
|  | @T(/button\_right/ = $pressed$ AND /button\_left/ = $pressed$) | $NoOp$ |
|  | @T(/button\_forward/ = $pressed$ AND /button\_backward/ = $pressed$) | $NoOp$ |
|  | @T(/button\_forward/ = $released$) | //output\_message// = “ST00000000” |
|  | @T(/button\_backward/ = $released$) | //output\_message// = “ST00000000” |
|  | @T(/button\_left/ = $released$) | //output\_message// = “ST00000000” |
|  | @T(/button\_right/ = $released$) | //output\_message// = “ST00000000” |
|  | @T(/button\_left/ = $released$ AND /button\_forward/ = $released$) | //output\_message// = “ST00000000” |
|  | @T(/button\_right/ = $released$ AND /button\_forward/ = $released$) | //output\_message// = “ST00000000” |
|  | @T(/button\_left/ = $released$ AND /button\_backward/ = $released$) | //output\_message// = “ST00000000” |
|  | @T(/button\_right/ = $released$ AND /button\_backward/ = $released$) | //output\_message// = “ST00000000” |
|  | @T(/button\_right/ = $released$ AND /button\_left/ = $released$) | $NoOp$ |
|  | @T(/button\_forward/ = $released$ AND /button\_backward/ = $released$) | $NoOp$ |
|  | @T(/button\_change\_speed/ = $released$) | Send //output\_message// based on /input\_speed/ |
|  | @T(/button\_sensor\_ultrasonic/ = $released$) | //output\_message// = “RS30000000” |
|  | @T(/button\_sensor\_light/ = $released$) | //output\_message// = “RS40000000” |
|  | @T(%get\_connection%) | $NoOp$ |
|  | @T(%end\_connection%) | Go to \*No Connection\* |
|  | @T(/input\_message/ = “RS1~”) | //sensor\_touch// = !reading! |
|  | @T(/input\_message/ = “RS2~”) | //sensor\_sound// = !reading! |
|  | @T(/input\_message/ = “RS3~””) | //sensor\_ultrasonic// = !reading! |
|  | @T(/input\_message/ = “RS4~”) | //sensor\_light// = !reading! |
|  | @T(%is\_error\_message%) | lookup  error in !error\_message\_table! and display message on //data\_log// |
| \*Awaiting Connection\* | @T(%connection\_received%) | Go to \*Normal Operation\* |
|  | @T(%time\_out%) | Go to \*No Connection\* |
| \*No Connection\* | @T(%get\_connection%) | Go to \*Awaiting Connection\* |