**RoboTrike**

**Functional Specification**

Description: The system is a three-wheeled robotic car, called a RoboTrike that moves through

holonomic motion and fires a laser from a turret. The system consists of a control component with keypad and display for the user to control the movement of the car, as well as a motor component that moves depending on actions done to the control component and sends the current status to the control component. The two components are connected by a wireless serial interface using a defined protocol, and the serial port data will be buffered to be acted upon and displayed. The user can select any angle for the RoboTrike to move in (since the body of the RoboTrike itself cannot turn), as well as any speed to move at, and can fire the laser through the keypad.

Global Variables:None

Input: **From Control Unit:**

All input is through a 4 x 4 keypad. Depending on the keys pressed, the keypad can use different menus for the meanings of thekeys, as described below:

Table 1. Key name and description of function for each key in Main Menu

|  |  |  |
| --- | --- | --- |
| Position | Key Name | Description |
| 1, 1 | Fire | Fire Laser |
| 1, 2 | Laser off | Turn Laser Off |
| 1, 3 | Additional Menu | Move to additional menu |
| 1, 4 | Speed Menu | Move to speed menu |
| 2, 1 | Forward Left | Move at a 45 degree angle forwards and left |
| 2, 2 | Forward | Move straight ahead |
| 2, 3 | Forward Right | Move at a 45 degree angle forwards and right |
| 2, 4 | Angle Menu | Move to Angle Menu |
| 3, 1 | Left | Move left |
| 3, 2 | Stop | Stop moving |
| 3, 3 | Right | Scrolls down in the display |
| 3, 4 | Half Speed | Moves at half of the maximum speed |
| 4, 1 | Backward Left | Moves backwards and left |
| 4, 2 | Backward | Moves backwards |
| 4, 3 | Backwards Right | Moves backwards and right |
| 4, 4 | Full Speed | Moves at the maximum speed |

Table 2. Key name and description of function for each key in Speed Menu

|  |  |  |
| --- | --- | --- |
| Position | Key Name | Description |
| 1, 1 | 100% | Moves at 100% of max speed |
| 1, 2 | 90% | Moves at 90% of max speed |
| 1, 3 | 80% | Moves at 80% of max speed |
| 1, 4 | Main Menu | Moves to main menu |
| 2, 1 | 70% | Moves at 70% of max speed |
| 2, 2 | 60% | Moves at 60% of max speed |
| 2, 3 | 50% | Moves at 50% of max speed |
| 2, 4 | 40% | Moves at 40% of max speed |
| 3, 1 | 30% | Moves at 30% of max speed |
| 3, 2 | 20% | Moves at 20% of max speed |
| 3, 3 | 10% | Moves at 10% of max speed |
| 3, 4 | 0% | Moves at 0% of max speed |
| 4, 1 | +2% | Increments by 2% of max speed |
| 4, 2 | -2% | Decrements by 2% of max speed |
| 4, 3 | +20% | Increments by 20% of max speed |
| 4, 4 | -20% | Decrements by 20% of max speed |

Table 3. Key name and description of function for each key in Angle Menu

|  |  |  |
| --- | --- | --- |
| Position | Key Name | Description |
| 1, 1 | 320˚ | Move at 320 degrees |
| 1, 2 | 300˚ | Move at 300 degrees |
| 1, 3 | 270˚ | Move at 270 degrees |
| 1, 4 | 240˚ | Move at 240 degrees |
| 2, 1 | 210˚ | Move at 210 degrees |
| 2, 2 | 180˚ | Move at 180 degrees |
| 2, 3 | 150˚ | Move at 150 degrees |
| 2, 4 | Main Menu | Move to Main Menu |
| 3, 1 | 120˚ | Move at 120 degrees |
| 3, 2 | 90˚ | Move at 90 degrees |
| 3, 3 | 60˚ | Move at 60 degrees |
| 3, 4 | 30˚ | Move at 30 degrees |
| 4, 1 | +5˚ | Increment angle by 5 degrees |
| 4, 2 | -5˚ | Decrement angle by 5 degrees |
| 4, 3 | +50˚ | Increment angle by 50 degrees |
| 4, 4 | -50˚ | Decrement angle by 50 degrees |

Table 4. Key name and description of function for each key in Additional Menu

|  |  |  |
| --- | --- | --- |
| Position | Key Name | Description |
| 1, 1 | ScrollLeft | Scroll Display Left by one digit |
| 1, 2 | ScrollRight | Scroll Display Right by one digit |
| 1, 3 | Main Menu | Move to main menu |
| 1, 4 | BlinkChange | Change blinking |
| 2, 1 | StartAutoScroll | Start autoscrolling the display |
| 2, 2 | StopAutoScroll | Stop autoscrolling the display |
| 2, 3 | None | None |
| 2, 4 | None | None |
| 3, 1 | None | None |
| 3, 2 | None | None |
| 3, 3 | None | None |
| 3, 4 | None | None |
| 4, 1 | None | None |
| 4, 2 | None | None |
| 4, 3 | None | None |
| 4, 4 | None | None |

**From Motor Unit:**

Serial interface is through a standard serial port using a 16C450 UART. The serial interface is used to send input from the motor unit to the serial unit. This is in the form of status or error updates delimited by carriage returns. These updates will then cause different messages to be output to the display. The formats of these updates are:

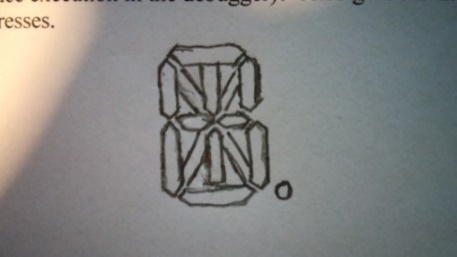
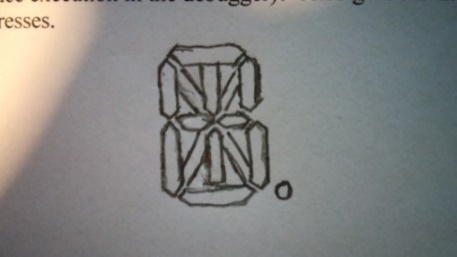
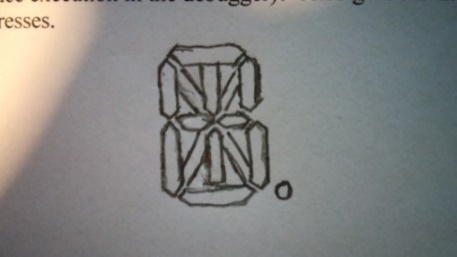
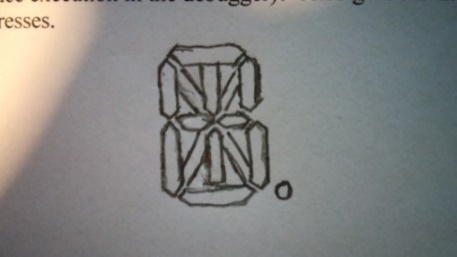
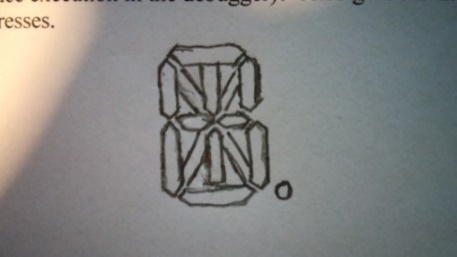
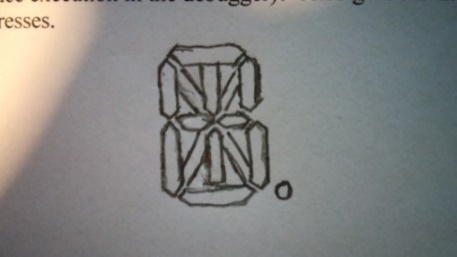
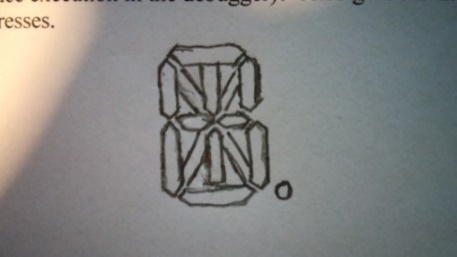
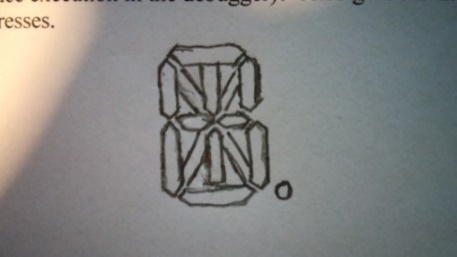
Table 5. Format of motor status/errors sent over serial channel to remote unit

|  |  |  |  |
| --- | --- | --- | --- |
| Update Type | Format | Notes | Example |
| Speed Status | S-----<return> | Max 5 digits of speed | S2934<return> |
| Direction Status | D---<return> | Max 3 digits of direction | D394<return> |
| Motor Parsing Error | P<return> | Any extra characters will result in error | N/A |
| Motor Serial Error | M--<return> | Max 2 digits of error (value of line status register in 16C450) | M16<return> |

Output: **To User**

The control unit has an8 digit 14 segment horizontally scrollable LED display. The display scrolls right and left based on user input and at a fixed rate if set to do so and can display a maximum of 100 characters. In addition, the display blinks at a fixed rate if set to do so.

Figure 1. General 8 digit 14 segment LED Display



To Motor Unit

The display normally shows status updates of the speed and direction of the motor unit, as well as whether there was a parsing error on the remote side. The format of the display for a status update is:

NP Speed …... Direction …

or

PE Speed ….. Direction …

where each dot represents a spot where a digit can be shown based on current speed and direction. The display shows NP if there is no parsing error on the motor unit and PE if there is a parsing error. This display is updated as commands are sent and executed by the motor unit.

For example, the display could show (while scrolling through):

NP Speed 49303 Direction 234

In addition, if a serial error occurs in either the control or motor unit, the error is displayed indicating which unit is from and what the specific error was. Each error message contains either ‘M ‘ or ‘S ‘ at the start of the error to indicate that the error is from the motor or remote unit.

For example, an error message could be

‘M Framing’

Also, if there is some invalid status sent from the motor unit to the control unit, an error message ‘Communication Error’ is displayed.

**To Motor Unit**

Serial commands are sent to the motor unit based on what keys are pressed. The commands can fire/turn off a laser and change the speed and angle of movement of the RoboTrike. Specifying a speed and angle of movement causes the 3 DC motors to rotate the wheels to move in that direction and at that speed. The commands output to the serial channel from the command unit are in the following format (From Website’s Command Format):

‘S#’ set to the absolute speed specified by the parameter (#). ‘V#’ accelerates or decelerates by the relative speed specified by the parameter (#). ‘D#’ moves in the direction specified by the parameter (#) relative to the current direction of movement. ‘F’ fires the laser and ‘O’ turns the laser off.

For example commands output from the control unit via the serial channel could be (From Website’s Command Format):

S15 start moving forward at speed 15

V+20 speed up to a speed of 35

D+45 change direction to 45 degrees to the right

F turn on (fire) the laser

O turn off the laser

V-10 slow down to a speed of 25

S0 halt the RoboTrike

S32767 start moving forward at speed 32767 (half speed)

V+32767 speed up to speed 65534 (full speed)

User Interface: The main input is through a 4x4 keypad laid out as shown below.

Figure 2. Various functions/numbers associated with keypad. Coordinates given to each key. Key functions in black are for the main menu. Key functions in green are for the angle menu. Key functions in purple are for the speed menu. Key functions in red are for the additional menu.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** |
| **1** | **Fire**  **100%**  **320°**  **Scroll Right** | **Laser Off**  **90%**  **300°**  **Scroll Left** | **Add’l Menu**  **80%**  **270°**  **Main Menu** | **Speed Menu**  **Main Menu**  **240°**  **Blink Change** |
| **2** | **Forward Left**  **70%**  **210°**  **AutoScroll** | **Forward**  **60%**  **180°**  **No Autoscroll** | **Forward Right**  **50%**  **150°** | **Angle Menu**  **40%**  **Main Menu** |
| **3** | **Left**  **30%**  **120°** | **STOP**  **20%**  **90°** | **Right**  **10%**  **60°** | **Half Speed**  **0%**  **30°** |
| **4** | **Backward Left**  **+2%**  **+5°** | **Backward**  **-2%**  **-5°** | **Backward Right**  **+20%**  **+50°** | **Max Speed**  **-20%**  **-50°** |

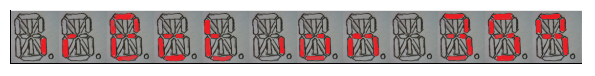
The RoboTrike moves based on the angle of movement and speed set by the user with the

keypad.

The angle can be set 2 ways. Either one of the 8 cardinal directions can be set directly from the menu itself using the Forward Left, Forward, Forward Right, Left, Right, Backward Left, Backward, or Backward Right buttons. In addition, more specific increments can be set by entering the angle menu (2, 4). From this menu, any of the keys other than the (2, 4) key will set specific angles, or increment or decrement the angle by some amount, as detailed by the green commands in the above keypad diagram. The main menu can be returned to by pressing the Angle Menu button (2, 4).

The speed can also be set 2 ways. The RoboTrike can be set to stop (zero speed), half of the maximum speed, or the maximum speed from the main menu using keys (3, 2), (3, 4), and (4, 4) respectively. To set more specific speeds, the speed menu can be entered by pressing (1, 4). From here, different absolute speeds can be set in the top 3 rows of the keypad, as shown in the purple commands of the keypad diagram. To increment or decrement the speed by some amount, the bottom row of the keypad can be used for a small and large increment/decrement. To return to the main menu, the speed menu button must be pressed again (1, 4).

The set angle and speed will also be displayed on the 8 digit 14 segment LED scrollable display. The angles are displayed from 0 to 360 degrees, while the speed is displayed from 0 to 65534 (where 65534 is max speed). After setting both speed and angle, if there were no parsing errors in the motor unit, the display will look like the following (expanding out of 8 digits of LEDs since the extra will be scrolled through):

Figure 3. Display of an example status output (scrolling is rolled out)



If there are parsing errors, the NP will be replaced with a PE.

Movement from will be sent via the serial interface to the motor unit. This will cause the 3 DC motors to rotate the wheels forward or backward such that the RoboTrike moves in the specified direction. This is done by connecting the 3 DC motor drivers to port B of an i8255 that stores the data written to it and holds it on the output lines. Three bits of port B correspond to whether each motor will run clockwise or counterclockwise (wheels rotate in either direction). Each DC motor will have its speed set using Pulse Width Modulation. The possible range for speed is from 0 to 100% of the highest possible speed. If a speed is not set when the RoboTrike is powered on, the RoboTrike will not move. When the RoboTrike is powered on, if an angle or direction of motion is specified but no speed is provided, then the RoboTrike will remain at rest until a speed is given. Once it is, the RoboTrike will move in the initially specified direction. no direction or angle is specified to move in, the RoboTrike will move in a default forward direction. Once the RoboTrike is moving at some speed, until a new speed is set, the RoboTrike will continue at the last set speed.

To fire the laser, press the Fire Laser key (1, 1). The laser will be fired until the Laser Off key (1, 2) is pressed to turn off the laser.

In addition, the keypad includes autorepeat functionality. If a key is continually pressed without removing the press, it will first start autorepeating at a slow rate of 0.5 Hz. If the key is pressed for 5 seconds, then the autorepeat rate speeds up to 1 Hz.

The display will also blink if the blink key (1, 4) is pressed when in the additional menu. If the display is already blinking, pressing this key will turn off the blinking.

The display will also scroll from left to right. If the autoscroll is enabled by pressing key (2, 1) when in the additional menu, then the display will be scrolled right once every second. If the autoscroll is disabled, the display will not automatically scroll right, but can be scrolled manually through the keypad. Whether autoscrolling is enabled or not, pressing the scroll right and left keys (1, 1) and (1, 2) in the additional menu will allow for scrolling the display by one digit left or right for each press. As explained before, if the scroll key is continually pressed, it will autorepeat to allow for faster scrolling rates.

Error Handling: If an error occurs in the serial interface or is reported by the motor unit, a description of it will be

shown on the display. The following table details the errors that can be reported by the display. The error will be prefixed by ‘M’ or ‘S’ depending on whether the serial error occurred in the motor or remote unit. The errors are detailed below.

Table 6. Strings displayed for different serial errors. Would be prefixed by ‘M ‘ or ‘S ‘ depending on whether the serial error was from the motor or remote unit, respectively.:

|  |  |
| --- | --- |
| ‘Nothing’ | No serial error |
| ‘Overrun’ | Overrun error |
| ‘Parity’ | Parity error |
| ‘Par/Over’ | Parity error and overrun error |
| ‘Framing’ | Framing Error |
| ‘Fram/Ove’ | Framing and overrun error |
| ‘Fr/Pari’ | Framing and Parity Error |
| ‘Fr/Pa/Ov’ | Framing and parity and overrun error |
| ‘BreakInt’ | Break interrupt |
| ‘Break/Ov’ | Break interrupt and overrun error |
| ‘Break/Pa’’ | Break interrupt and parity error |
| ‘Break/Pa/Ov’ | Break interrupt and parity and overrun error |
| ‘Bre/Fra’ | Break interrupt and framing error |
| ‘Br/Fr/Ov’ | Break interrupt and framing and overrun errors |
| ‘Br/Fr/Pa’ | Break interrupt and framing and parity errors |
| ‘B/F/P/O’ | Break interrupt, framing, parity, and overrun errors |

In addition, if there is an invalid command sent from the control to motor unit, there will be a parsing error reported by ‘PE’ at the start of the status display. If there is an invalid status sent from the motor to control unit, then there will be a ‘Communication Error’ message displayed instead of any status. When another valid status is sent, the communication error message will be replaced by the new status.

Algorithms: The holonomic motion algorithm is used to allow the RoboTrike to move in any direction as well

as to allow it to move without needing feedback to help determine its position. The holonomic motion calculates the pulse widths to drive each of the motors with by using the fact that s = F●v, which gives how long each motor should be on for and how long it should be off for to move in a certain direction with some speed. In addition, the serial parsing algorithm is used to transmit data from the motor unit to the control unit and back through a serial interface. This is done through a Mealy finite state machine that connects an action with each transition between different states, where the state entered depends on the type of character received. This allows the parsers to do something meaningful (display message/run motors) with the data they transmit between each other.

Data Structure: Tables are used for the User Interface in multiple ways. They are used for each of the keys in the

keypad to detail the function to be performed when the key is pressed along with which menu to move to. Also, tables are used for the different error messages to be displayed to the user. Finally, tables are mainly used to implement the finite state machines on both the motor and control unit for the tokens as well as the main transition table.

In addition, queues are used to process data that arrives in large bursts when there is not enough time to handle it immediately. This is done through an EventQueue which holds a value and type for each element which together specify what to do with the event.

Limitations: The RoboTrike can never turn in a specific direction, or rotate about its axis. In addition, the RoboTrike cannot travel along a pre-selected path, and must be controlled step by step to execute any path. Also, multiple key presses are not handled, and the rightmost key in a row, or the first row scanned is the key which is handled. Other key presses at the same time are ignored. The scrolling does not also allow for autoscrolling back left when the end of the buffer is reached, and instead stops at the last scroll frame of the buffer unless another message to display resets it to the start or if the user manually scrolls back left. Also, the turret is not controlled at all, with no way to elevate or rotate the turret.

Known Bugs: The display buffer is sometimes overwritten in parts by error messages, though this does not show up generally and can be reset by outputting another status update.

Special Notes: None