

RoboTrike Functional Specifications

Description The system is a three-wheeled robotic car which can be controlled via keypad over a serial interface. The RoboTrike moves via holonomic motion, and its three motors control its three wheels in order to move in all directions at different speeds. It also has a laser that can "fire".

The RoboTrike comes in two components: the robot itself and the remote controller board. The robot contains the motors and laser as mentioned before, as well as a serial port, and the controller board contains a keypad to control the robot. Each key on the keypad maps to a different command and controls a different part of the robot, such as its speed or direction.

The controller board also contains a 14-segment LED display, which can display the RoboTrike's status, including its speed, direction, and laser state, as well as any errors that may have occurred from the serial channel or on the RoboTrike itself. It also contains a serial port used for communication with the motor board.

Global Variables None.

Input The RoboTrike controller board receives input from the 16-key keypad, in which each key maps to a different command (as described in more detail in *User Interface*). It also receives feedback from the RoboTrike motor board via standard serial interface using a 16C450 serial chip.

The RoboTrike motor board also receives input via standard serial interface using a 16C450 serial chip.

Output The 14-segment LED display will display the status of the RoboTrike, as well as any errors that may have occurred and the current command that is being executed.

The controller board outputs commands via serial channel to the motor board, which control the speed at which the DC motors rotate as well as whether or not to turn the laser on. By having the wheels move at different rates, the RoboTrike allows for holonomic motion.

User
Interface

Controlling the RoboTrike

The RoboTrike can be controlled via a 16-key keypad, in which each key maps to a different command. There are two menus to choose from, each which maps keys to different commands. They are as follows:

Main Menu

↑ Speed	Fun!	↑ Angle	Main Menu
↓ Speed	Stop	↓ Angle	Baud Menu
Laser On	Go/ Forward	Laser Off	Insanity
Left	Reverse	Right	Display Status

The *Main Menu* contains commands to control the RoboTrike's speed, direction, and laser state. To switch to the *Baud Menu*, simply press on the *Baud Menu* key.

Baud Menu

			Main Menu
Baud 150	Baud 1200	Baud 2400	Baud Menu
Baud 4800	Baud 9600	Baud 19200	Insanity
			Display Status

The *Baud Menu* contains commands to control the baud rate of the serial channel between the control board and the motor board. The desired baud rate can be selected by pressing one of the six available keys mapping to different baud rates.

A description of each key is below. Keys are split into three categories: *Functional Keys* (the keys highlighted in blue), *Main Menu* keys, and *Baud Rate* keys.

Functional Keys

These are various functional keys.

Key	Description
Main Menu	Changes the keypad layout so that it is in the <i>Main Menu</i> format.
Baud Menu	Changes the keypad layout so that it is in the <i>Baud Menu</i> format.
Insanity	Causes the RoboTrike to violently spin around quickly for a few seconds.
Display Status	Displays the status of the RoboTrike. More details can be found in the <i>LED Display</i> description.

Main Menu Keys

The main menu contains keys to control the motion of the RoboTrike and its lasers. For example, by pressing the **↑ Speed** key, the speed will increase and the RoboTrike will move faster.

Key	Description
↑ Speed	Increases the speed of the RoboTrike.
↓ Speed	Decreases the speed of the RoboTrike.
Fun!	Displays a fun message to the LED display.
↑ Angle	Increases the angle of direction that the RoboTrike is going in. Angle is measured clockwise from the current direction of motion.

↓ Angle	Decreases the angle of direction that the RoboTrike is going in. Angle is measured clockwise from the current direction of motion.
Stop	Stops the RoboTrike.
Laser On	Turns the RoboTrike's laser on.
Laser Off	Turns the RoboTrike's laser off.
Go/Forward	Tells the RoboTrike to go forward at its previous speed before it was stopped. If it wasn't stopped, doesn't do anything.
Left	Tells the RoboTrike to go in the direction 90 degrees to the left.
Reverse	Tells the RoboTrike to go in the reverse direction.
Right	Tells the RoboTrike to go in the direction 90 degrees to the right.

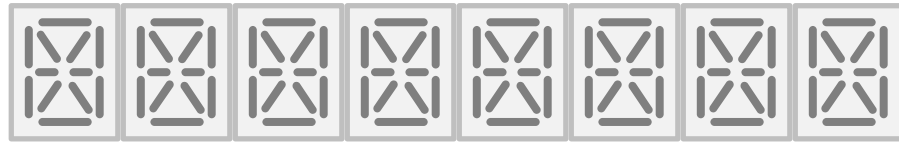
Baud Menu Keys

Note that when the below keys are pressed, no other key can be pressed until the controller board gets a confirmation from the motor board that the baud rate has been changed.

Key	Description
Baud 150	Changes the baud rate to 150.
Baud 1200	Changes the baud rate to 1200.
Baud 2400	Changes the baud rate to 2400.
Baud 4800	Changes the baud rate to 4800.
Baud 9600	Changes the baud rate to 9600.
Baud 19200	Changes the baud rate to 19200.

LED Display

The LED display is a display that can show up to 8 characters at a time, and each character has 14 segments like so:



It can display strings of varying length. If the string does not fit within 8 characters, the display will scroll the characters from left to right until it reaches the end of the string. Once the end of the string is reached, the display will restart from displaying from the first character again.

Status Messages

Depending on what key is pressed, the LED display will show different messages. If the *Display Status* key is pressed, a string containing the RoboTrike status is shown. The string is of the format:

***** S+##### A+#####**

Where **#** refers to the laser status (**L** if it is on, **_** if it isn't), **#####** refers to the speed of the RoboTrike (ranges from 00000 to 32767), and **###** refers to the angle of direction of the RoboTrike (ranges from 000 to 359).

Error Messages

If any error occurs, the LED display will show a string of the format:

ERROR ##

Where **##** refers to the error code of the error that occurred (more details in *Error Handling*).

Command Display

The LED display also displays the command that is being executed once a key is pressed. The keys are mapped as follows:

Main Menu

Inc SPEEd	HI KArEN PLEASE GIVE ME AN A+	Inc Angle	MAIN MENU
dEc SPEEd	StOP	dEc Angle	bAUd MENU
LASER On	FOrWARD	LASER OFF	INSANITY
LEFt	rEVERSE	rIght	

Baud Menu

			MAIN MENU
bAUd150	bAUd 1200	bAUd 2400	bAUd MENU
bAUd 4800	bAUd 9600	bAUd 19200	INSANITY

Error Handling

If any errors occur on the serial interface (i.e., the character could not be sent, etc.), or if there is a parsing error from the motor unit (i.e. a bad command was sent), then the error will be displayed onto the LED display. The error is a two-digit hex code, and can be understood when converted to binary. Each bit of the code represents a different error that occurred; more than one error can occur at once.

Bit	Description
0	Overflow of the serial channel's receive queue. A character had to be dropped.
1	Overrun error. Occurs when the character in the serial receive buffer is overwritten before it is read.
2	Parity error. Occurs when the parity of the received character in the serial channel doesn't match the parity selected.
3	Framing error. The received character does not have a valid stop bit.
4	Break interrupt handler. The received data input from the serial channel was held clear for longer than a full-word transmission time.
5	Parsing error. Some kind of error occurred when the motor board tried parsing the command from the serial channel.

The RoboTrike will not handle external sources of error. For example, if the RoboTrike attempts to move forward but is stuck at a wall, there is no feedback from the RoboTrike to the control board that it is unable to move forward.

Algorithms	Holonomic motion calculations, to figure out the rate at which to run the three motors in order to make the RoboTrike go in the desired speed and direction.
Data Structures	None (no global ones). For more details about local and shared variables, please see the documentation in the code.
Limitation	The RoboTrike cannot give any details about its whereabouts or external status. If a command to move is issued, it does not know if it actually moved or not.