Sophia Liu

RoboTrike Functional Specification

Remote unit main loop

Description:

This system allows an operator to control a three-wheeled robotic car (the RoboTrike) via a keypad and display over a serial interface. The system consists of two separate components, a remote unit with a keypad and display through which the user interacts with the system and the three-wheeled motor unit that can move around under user control. The keypad consists of various commands that allow the robot to move and fire a turret “laser.” The display shows current runtime information and errors. The motor unit can also send back status to be displayed. The two units communicate over a serial interface using a defined protocol.

Global Variables:

None.

Inputs:

Input is entered through an unlabeled 4x4 16 key keypad. Through this keypad, the user can increase or decrease the speed, turn

Outputs:

An 8-digit display consisting of 14 segments with right hand decimal point per digit is used to display messages while the system in running. These include status updates and error messages.

Three DC motors are used to move the RoboTrike via PWM (Pulse Width Modulation). These motor drivers are connected to port B of an 8255 chip. Each motor may be run clockwise or counterclockwise as determined by one bit of Port B for each motor. This is used to determine the direction of motion. One stepper motor is used to rotate the turret which is connected to port C of an 8255. It is configured as a unipolar drive and has four bits controlling it. The motor has a maximum step rate of 50 half-steps/sec. One servomotor is used to set the angle of elevation of the “laser.” This is controlled by a single bit of port C. All motors are controlled via 11 bits of parallel output of an 8255.

A serial interface is used to control the motor unit, and sends commands to the motor unit. The motor unit outputs its current status (speed, angle, turret position, etc) to the serial interface whenever it changes. The serial interface also receives commands from and sends status to the keypad and display unit.

There is also a turret “laser” which can be fired, or an LED that can be turned on. It is controlled via one bit of parallel output of an 8255.

User Interface:

|  |  |  |  |
| --- | --- | --- | --- |
| Increase speed (inc speed by amt) | Decrease speed (dec speed by amt) | Laser on | Laser off |
| Left (dec angle by deg) | Forward (forward at set speed) | Reverse (backward at set speed) | Right (inc angle by deg) |
|  |  |  |  |
|  |  |  |  |

|  |  |
| --- | --- |
| **Command** | **Action** |
| Increase Speed | Increases the RoboTrike speed by a set amount |
| Decrease Speed | Decreases the RoboTrike speed by a set amount |
| Laser On | Turns the turret laser on |
| Laser Off | Turns the turret laser off |
| Left | Turns the RoboTrike direction left 30 degrees |
| Right | Turns the RoboTrike direction right 30 degrees |
| Forward | Moves the RoboTrike forward at half speed. |
| Reverse | Sets the RoboTrike direction to 180 degrees (reverses the RoboTrike direction). |

|  |  |
| --- | --- |
| **Command** | **Display** |
| Increase Speed | XXXX (new speed in hexadecimal) |
| Decrease Speed | XXXX (new speed in hexadecimal) |
| Laser On | 00001 |
| Laser Off | 00000 |
| Left | 00XXX (new RoboTrike direction angle, 0 to 359) |
| Right | 00XXX (new RoboTrike direction angle, 0 to 359) |
| Forward | (new speed in hexadecimal) |
| Reverse | 00XXX (new RoboTrike direction angle, 0 to 359) |

The user moves the RoboTrike manually using the keypad, shown in “Inputs”. These are sent via the serial interface to the motor unit, which sends back status information which is displayed along with the current information about the RoboTrike movement.

Commands for the robot on the keypad are: Increase speed, decrease speed, laser on, laser off, left, right, forward, and reverse.

The display will show the current command being executed, and information such as the current speed and direction. It will also various errors from the serial interface or motor unit.

Error Handling:

The display also will show errors that occur on the serial interface, such as framing (ERR FRM), parity (ERR PAR), break (ERR BRK), overrun (ERR OR), and buffer overflow (ERR OF); or errors that are reported by the motor unit, such as bad command sent (ERR CMD). There is also a “debug mode” where incoming serial data is displayed on the display in hex.

Algorithms:

Movement: An algorithm will be used to move the vehicle in any angle. There are three wheels on the RoboTrike situated 120° from each other. They are controlled by three motors which can each run clockwise and counterclockwise. Varying direction and power given to each motor will allow the robot to maneuver directly in any direction without turning. Similarly, this allows the robot to travel at varying speeds.

Finite State Machine:

Data Structures:

Queues

Limitations:

Memory: There are 32K bytes of RAM and 32K bytes of ROM available. Serial EEROM can also store small amounts of data.

Known Bugs: None

Special Notes: None

EE/CS 51 RoboTrike Project

Sophia Liu

Table of Contents

Remote Unit Files

Remote Unit Main Loop

Display Functions

Display Functions Include File

14 Segment Display Tables

Conversion Functions

Conversion Functions Include File

Keypad Functions

Keypad Functions Include File

Display and Keypad Timer Event Handler

Display and Keypad Timer Event Handler Include File

Display and Keypad Timer Functions

Display and Keypad Timer Include File

Motor Unit Files

Motor Unit Main Loop

Motor Functions

Motor Functions Include File

Motor Timer Functions

Motor Timer Functions Include File

Motor Functions Trig Tables

Parser Functions

Parser Functions Include File

Remote and Motor Unit Shared Files

Main Loop Include File

Queue Functions

Queue Functions Include File

Event Queue

Event Queue Include File

Event Handler Functions

Event Handler Functions Include File

Event Constant Include File

Initialize Chip Select

Initialize Chip Select Include File

Interrupt Include file

Serial Functions

Serial Functions Include File

Serial Interrupt Functions