|  |
| --- |
| Straight outta compe |
| Design Changes & Tuning |
| Configuring the Design for Optimal Performance |
|  |
| **Authors:**  **Zack Rauen**  **Juan Merchan**  **Jayesh Bokhiria** |
|  |

|  |
| --- |
|  |

# Revision History

|  |  |  |
| --- | --- | --- |
| Date | Rev. Author | Revision |
| Nov. 6, 2015 | Z. Rauen | Initial layout & content. |
| Nov. 6, 2015 | J. Bokhiria | Integration of test plan. |
| Nov. 6, 2015 | J. Merchan | Addition of H-Bridges noise cancelling components and integration of schematics |

Table of Contents

[Revision History II](#_Toc434579821)

[1. Introduction 1](#_Toc434579822)

[1.1. Identification 1](#_Toc434579823)

[1.2. Purpose 1](#_Toc434579824)

[1.3. Scope 1](#_Toc434579825)

[1.4. Definitions, Acronyms and Abbreviations 1](#_Toc434579826)

[1.4.1. Definitions 1](#_Toc434579827)

[1.4.2. Acronyms 1](#_Toc434579828)

[1.4.3. Abbreviations 1](#_Toc434579829)

[1.5. References 1](#_Toc434579830)

[1.5.1. Documents 2](#_Toc434579831)

[1.5.2. Standards 2](#_Toc434579832)

[2. System Design Changes 2](#_Toc434579833)

[2.1. H-Bridge 2](#_Toc434579834)

[2.1.1. Changes 2](#_Toc434579835)

[2.1.2. Additions 2](#_Toc434579836)

[2.2. User Interface 3](#_Toc434579837)

[2.2.1. Additions 3](#_Toc434579838)

[2.3. Line Camera 3](#_Toc434579839)

[2.3.1. Additions 3](#_Toc434579840)

[3. Tuning 3](#_Toc434579841)

[3.1. PID 3](#_Toc434579842)

[3.1.1. Coefficients 3](#_Toc434579843)

[3.2. Steering 4](#_Toc434579844)

[3.2.1. Angles 4](#_Toc434579845)

1. Introduction
   1. Identification

This document shall be known as “Final Design Changes & Tuning” and has an identifier of SOC-Tuning to be referenced in future documents.

* 1. Purpose

The purpose of this document is to define any changes to the original design that have occurred as a result of necessity during implementation. This includes additions and removals to and of components as well as editing items defined in previous design documents.

* 1. Scope

This document’s scope is for the overall system. The changes laid out in this document have been made in order for the overall functionality of the system to be accurate to the degree specified in *SOC-SysReq*.

* 1. Definitions, Acronyms and Abbreviations
     1. Definitions
        1. **Car:** The main product, the Intelligent Car Version 3.
        2. **Mode:** A constant state of the system, in this case of the car. This can be changed by input from the user (1.4.1.5).
        3. **Product:** The entire system (1.4.1.4) as a finished design.
        4. **System:** This refers to the entire system in use, in this case the car as well as the personal computer that the car connects to in order to use data logging. Additionally this will include the optional mobile phone that acts as a remote as well.
        5. **User:** The person(s) that will be using or interacting with the product.
        6. **We:** The members of Straight Outta CompE.
     2. Acronyms
        1. **I2C:** Optionally I2C. Inter-Integrated Circuit.
        2. **IEEE:** Institute of Electrical and Electronics Engineers.
        3. **PC:** Personal Computer.
        4. **LED:** Light emitting diode.
     3. Abbreviations
        1. **V:** volts.
        2. **°:** degree(s).
        3. **Ω:** Ohms
        4. **μ:** Micro ()
        5. **k:** Kilo ()
        6. **p:** Pico ()
  2. References
     1. Documents
        1. SOC-SysDes-1
        2. SOC-SysDes-2
        3. SOC-SysDes-3
        4. PID Autotune V1 - https://github.com/br3ttb/Arduino-PID-AutoTune-Library/blob/master/PID\_AutoTune\_v0/
        5. Materials from EE416/464, Fall 2015
     2. Standards
        1. I2C
        2. UART
  3. Overview

This document serves to display the changes that have been found to be necessary for the system. This will include “System Design Changes” where components are added, removed, or edited. It also includes “Tuning” where specific portions of a component are edited to increase performance. Therefore it is necessary to view *SOC-SysDes-3* before this document.

1. System Design Changes

This section goes over the changes to components that have been defined in previous design documents such as *SOC-SysDes-3.* Here, the components functionality has been changed in some manner, either via additions or removals, or even wiring changes.

* 1. H-Bridge

The H-Bridge wiring has changed to allow for more control over the cars driving.

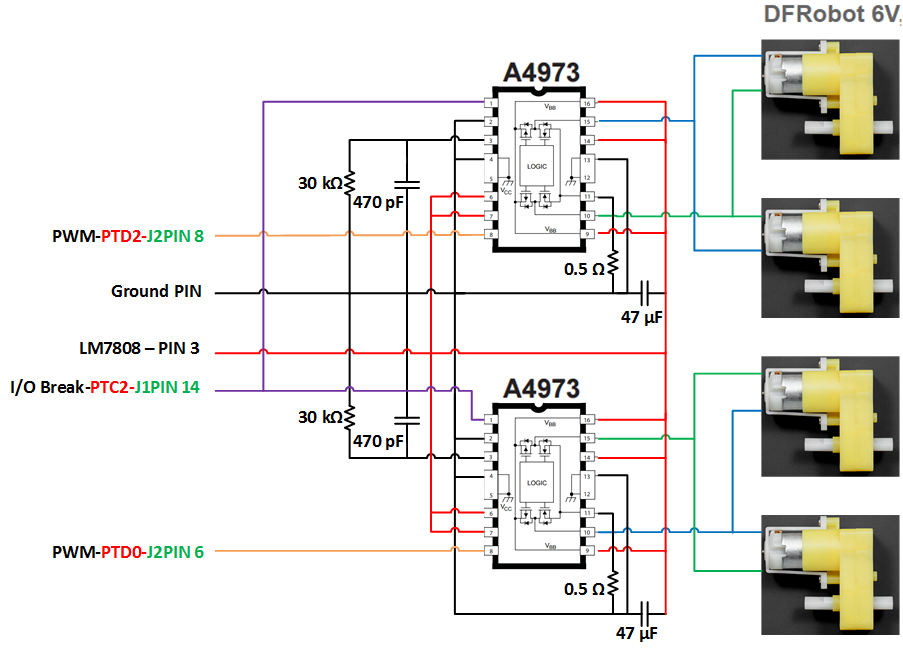


Figure 1 - H-Bridge Wiring

* + 1. Changes

Unless specified the changes list below are applied to both A4973 H-Bridges in Figure 1.

* + - 1. The ‘MODE’ pin (A4973 Pin 14) will be connected to VCC as this allows for fast current decay rather than slow current decay.
      2. The ‘PHASE’ pin (A4973 Pin 7) will also be connected to VCC allowing the car to move in the forward direction.
      3. The ‘Break’ pin (A4973 Pin 1) will also be connected to the microcontroller J1PIN 14 (I/O Pin), this pin enables the break mode of the H-Bridges. When the pin is high the car will be able to move and when low the motor will stop.
      4. The PWM (J2PIN8) output of the microcontroller is now connected to the ‘ENABLE’ pin (A4973 Pin8) of the right side H-Bridge (Top A4973 H-Bridge in Figure 1).
      5. The PWM (J2PIN6) output of the microcontroller is now connected to the ‘ENABLE’ pin (A4973 Pin8) of the left side H-Bridge (Bottom A4973 H-Bridge in Figure 1).
      6. The RC Pin (A4973 Pin 3) is connected in parallel to a 30kΩ resistor and a 470pF capacitor, then grounded to reduce electrical noise.
      7. The Sense Pin (A4973 Pin 11) is connected to a 0.5Ω resistor, then grounded to reduce electrical noise.
      8. The Load Supply Pins (A4973 Pins 9 and 16) are connected in parallel to the 6V load supply (LM7808-Pin 3) and a 47μF capacitor which is grounded to reduce electrical noise

Since the H-Bridge uses average DC power, a lower duty cycle corresponds to the motors spinning faster. This occurs because the ‘ENABLE’ is an active-low line.

* + 1. Additions

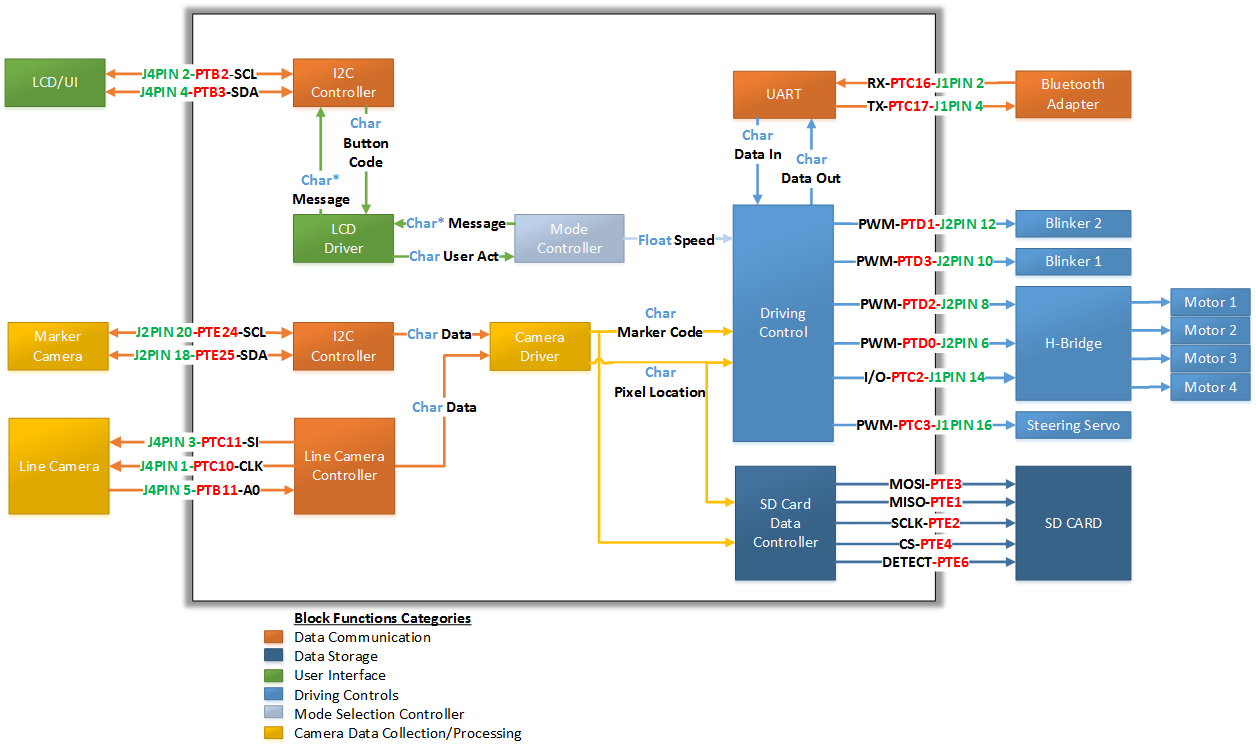


Figure 2 - Top Level Design Block Diagram

* + - 1. The ‘BRAKE’ input will now be connected to the microcontroller (PTC2-J1PIN14) in order to allow quick stopping for intersections. (Shown in Figure 2)
  1. User Interface

The UI has a new addition in order to prevent a failure to meet the requirements of the design.

* + 1. Additions
       1. The onboard button of the microcontroller (SW2) will also function to change the system mode.
       2. The LED on the PIXY camera will change corresponding to mode.

This additions prevents a failure due to I2C implementation trouble. Since the previous design for a UI used I2C, if I2C fails to be implemented this addition will still allow a changing of the mode for the user. The LED of the PIXY camera, then, acts as the signifier to the user in lieu of the LCD display.

* 1. Line Camera

The additions to the Line Camera module allow for increased performance, not just for the module but for the entire system.

* + 1. Additions
       1. Three high-intensity white LEDs will be pointing approximately the same angle as the Line Camera.

The LEDs added to the module allows the Line Camera to see the track more accurately. Not only does this help the Line Camera distinct black from white but it also reduces the cosine effect, allowing the edges of the signal to be interpreted properly.

1. Tuning

The Tuning section involving adjust small portions of individual components that adjust the performance of the overall system.

* 1. PID

The tuning of the PID is concerned with adjusted the parameters that lead to an output for the servo. More specifically, the coefficients of the calculated values as defined in *SOC-SysDes-3*. Also, in order to more easily tune these, they will all be multiplied by a new variable *Ka.*

* + 1. Coefficients
       1. KP will be equivalent to 0.6\*Ka.
       2. KI will be equivalent to 1.2\*Ka.
       3. KD will be equivalent to 0.075\*Ka.

During the final implementation phase, the variable *Ka* will be varied from 0 to 2 in increments of 0.1. The value that performs the best during testing will be the final value for all the coefficients. Performing the best for this test shall be defined as the value that allows the car to most closely follow the line.

* 1. Steering
     1. Angles
        1. The right servo angle will correspond to the wheels turning +30 degrees.
        2. The left servo angle will correspond to the wheels turning -30 degrees.

Since the steering system of the car is asymmetrical, the values of the steering servo will have to be determined by testing. In theory, with a 1:1 ratio of turning, the servo will need to turn -30 and +30 degrees for right and left respectively. To determine the values through testing, the servo will run through all angled (-90 to +90 degrees) and will be stopped by a button interrupt (SW2) on the microcontroller whenever the wheels are aligned to +30 degrees or -30 degrees. The value of the angle will be transmitted to the PC over serial communications. These values will then be recorded and used to steer the car.