|  |
| --- |
| Straight outta compe |
| Final Documentation SOC |
| Analytics of Final Product |
|  |
| **Authors:**  **Zack Rauen**  **Juan Merchan**  **Jayesh Bokhiria** |
|  |

|  |
| --- |
|  |

# Revision History

|  |  |  |
| --- | --- | --- |
| Date | Rev. Author | Revision |
| Nov. 4, 2015 | J. Bokhiria | Initial layout & content. |
| Dec. 11, 2015 | J. Merchan | Addition of sections 2-3 and final editing. |
| Dec. 11, 2015 | Z. Rauen | Final edit. |

Table of Contents

[Revision History II](#_Toc437618209)

[1. Introduction 1](#_Toc437618210)

[1.1. Identification 1](#_Toc437618211)

[1.2. Purpose 1](#_Toc437618212)

[1.3. Scope 1](#_Toc437618213)

[1.4. Definitions, Acronyms and Abbreviations 1](#_Toc437618214)

[1.4.1. Definitions 1](#_Toc437618215)

[1.4.2. Acronyms 1](#_Toc437618216)

[1.4.3. Abbreviations 1](#_Toc437618217)

[1.5. References 1](#_Toc437618218)

[1.5.1. Documents 1](#_Toc437618219)

[1.5.2. Standards 2](#_Toc437618220)

[2. Evaluation and Analysis 2](#_Toc437618221)

[2.1 Specifications 2](#_Toc437618222)

[2.2 Unique Innovations 3](#_Toc437618223)

[2.3 Weaknesses 3](#_Toc437618224)

[3. Test Results 3](#_Toc437618225)

[4. Appendix 4](#_Toc437618226)

[4.1. Car System Design 4](#_Toc437618227)

[4.2. Software Source code 5](#_Toc437618228)

[4.3. Hardware Components 5](#_Toc437618229)

[4.3.1. H-Bridge 5](#_Toc437618230)

[4.3.2. Costume Steering design 7](#_Toc437618231)

[4.3.3. Steering Servo (HS-422 Servo) 7](#_Toc437618232)

[4.3.4. Power Regulator 8](#_Toc437618233)

[4.3.5. Line Scan Camera 8](#_Toc437618234)

[4.3.6. CMUCam5 Pixy Vision Sensor Camera 9](#_Toc437618235)

[4.3.7. SD Card 10](#_Toc437618236)

[4.3.8. BlueSMiRF Bluetooth 11](#_Toc437618237)

[4.3.9. Liquid Crystal Display (LCD) 11](#_Toc437618238)

[4.3.10. Blinkers and Brake Light 11](#_Toc437618239)

1. Introduction
   1. Identification

This document shall be known as “Final Product Documentation” and has an identifier of SOC-Final Doc to be referenced in future documents.

* 1. Purpose

The purpose of this document is wrapping all work which makes up the car system. This includes all components of the car at a hardware and software level and also contains the test results.

* 1. Scope

This document’s scope is for the overall system. The changes and modifications 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

**Car:** The main product, the Intelligent Car Version 3.

**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).

**Product:** The entire system (1.4.1.4) as a finished design.

**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.

**User:** The person(s) that will be using or interacting with the product.

**We:** The members of Straight Outta CompE.

* + 1. Acronyms

**I2C:** Optionally I2C. Inter-Integrated Circuit.

**IEEE:** Institute of Electrical and Electronics Engineers.

**PC:** Personal Computer.

**LED:** Light emitting diode.

* + 1. Abbreviations

**V:** volts.

**°:** degree(s).

**Ω:** Ohms

**μ:** Micro ()

**k:** Kilo ()

**p:** Pico ()

* 1. References
     1. Documents

1. SOC-SysDes-1
2. SOC-SysDes-2
3. SOC-SysDes-3
4. SOC-Tunning
5. Materials from EE416/464, Fall 2015
   * 1. Standards

I2C

UART

* 1. Overview

This document serves to display the evaluation and analysis, test results, and appendix which contains hardware and software level components of the car system.

1. Evaluation and Analysis

## Specifications

The basic functionality of the car can be broken down into three small sub-components. It must be able to follow the black line accurately and change its turning direction based on a color coded marker. Lastly, it must perform three operation modes and the user must be able to select any mode at any time.

The car was able to perform as desired and a list of successful test results is listed below:

* + 1. The car tracks a black line against a white background
    2. The car navigates autonomously around on the track
    3. The car uses a line camera to detect a black line on the track
    4. The car uses vision sensor (CMUcam5 Pixy) to detect posts
    5. The car follows the commands from the posts which are located on the left and right sides of the track at the intersection on the track
    6. The car obeys the post closest to the intersection when more than one post is present
    7. The car moves when powered on
    8. The car changes the mode when the user interacts with the interface
    9. The car moves in the forward direction of the track
    10. The car follows the black line of the track as closely as possible
    11. The car stops while staying on the track within 1 meter of the stop line
    12. The data from the car can be logged and saved via an SD card or bluetooth.
    13. The car also has some optional features such as an LCD display for modes, blinkers, and brake lights. When the pixy camera recognizes a direction signature, the LED corresponding to that direction will turn on.

Additionally, the car is able to operate in the three operation modes after the user selects an operation mode. Some of the test results for every challenge are listed below.

* + 1. **Discovery Mode**
       1. The car is placed within 2 ft. of the edge of any part of the track at any orientation
       2. The car starts searching for the track when commanded
       3. The car can locate the track within 3 minutes
       4. The car automatically restarts if it fails to locate the track within 3 minutes
    2. **Accuracy Mode**
       1. The car is placed at the starting point on the track
       2. The car follows the black line of the track as closely as possible until it has completed two laps
       3. The car automatically stops after the completion of the second lap
    3. **Speed Mode**
       1. The car is placed at the starting point on the track
       2. The car follows the black line of the track as closely as possible until it has completed two laps
       3. The car automatically stops after the completion of the second lap
       4. The car has no more than two trials to complete this challenge with at least one valid lap

## Unique Innovations

The unique components of our car system make it possible for the car system to follow the black line with 95% accuracy at 85% speed setting with an average lap time of 1 minute and 10 seconds in accuracy mode while maintaining the car center with the line at any point. More unique system design are listed below:

* + 1. Our design uses a unique Hybrid driving controller that utilizes a front steering system and a standard differential. By implementing both of these components, the car is able to make sharper turns at higher speeds and turn on its on axis when an intersection is detected.
    2. LCD user Interface
    3. Signal Lights
    4. Our software design is very modular and very easy to read.
    5. A partial library was developed which acts similar to the HAL.
    6. Multiple *structs* and *typedefs* allow us to mimic classes.

## 2.3 Weaknesses

Below is a list of problems that we ran into while implementing the car system; however, these problems were fixed before the car system was delivered. However there were external problems outside of our control. A breakdown of the problems (weaknesses) are listed and described below:

* + 1. Electromagnetic Interference – The car system chassis deign was made to be compact and user friendly. To accomplish this the circuit board which holds the H-Bridge and Power regulator components is mounted inside the car chassis along with the power distribution board. As a result, the electromagnetic noise from the driving motors interferes with the grounding system of the distribution board.
    2. Under-steering – After the steering system was implemented, the car was able to make 90ᵒ turns, but it was unable to stay center with the line when turning in a circle or turning in an intersection. To solve this, the Hybrid drive system was implemented. (Descried in section 2.2.1)
    3. Physical switches sometimes caused the operation mode to change. In order to resolve this issue, we exerted caution when operating the physical switches. This problem can be caused by electrical noise in the ground line.
    4. Ambient light affects the detection of the intersection line or the start/stop line. This problem was encountered when testing during early hours in the morning. This can be solved by adjusting the angle of the line camera depending on the time of the day.

1. Test Results

For all software testing the results were printed on the PuTTY. For all hardware testing the results were captured using an oscilloscope, LeCroy, or a DMM meter.

During the pixy camera implementation, the camera was capturing only one signature even if there were two signatures. About ninety-percent of the time, this was the case. We used this test data and added a half function to fix this issue.

During the SD card implementation, used PuTTY to capture the generate results and modified the code for desired result.

During the bluetooth implementation, used PuTTY to capture the data results and modified the code for desired result.

For the calibration and trouble shooting of the line camera and driving controller, the collected data was printed out to Putty and also saved in the SD card. The printed data contained the location of the line with respect to the car and the turning direction set by the markers. The marker data is collected by the Pixy cam and processed by the I2C controller. This data was very useful to fix and tune the PID controllers (The Hybrid system contains 3 PID controllers) and Pixy cam angle and lens focus.

Additional testing procedure and results can be found in the SOC-Final Test Plan located in redmine and in the SOC documentation binder.

1. Appendix
   1. Car System Design

FigureA shows the overall system logical block diagram for hardware and software. The blocks located inside the black rectangle represent software components to be implemented in the Freedom board. The other blocks located in the outside of the black rectangle represent Hardware components of the car system.

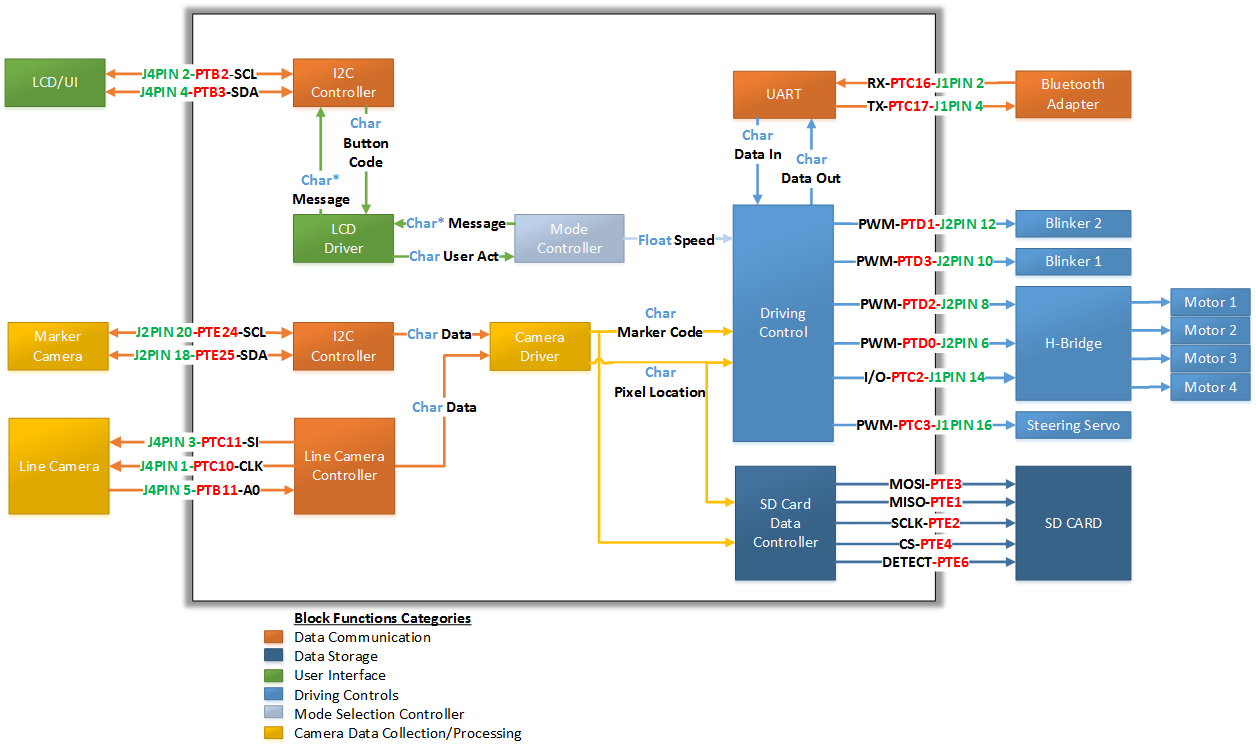


Figure A – System level block diagram

The car system block (FigureA) diagram is divided into sub-block categories represented by the different color patterns. Each block category will be described in detail in section 3.

* + - 1. The green blocks represent the user interface components
      2. The light blue Blocks represent the hardware and software components of the cars drive train
      3. The yellow blocks represent the data input and analysis from the two different cameras
      4. The orange blocks represent the different communication decoders and encoders
      5. The baby blue block represents the mode selection component
      6. The dark blue blocks represent the data collection and storage using a SD card
  1. Software Source code

The Source code for SOC system design is located in the SOC Redmind folder.

* 1. Hardware Components

A more in-depth description of the hard ware components can be found in the SOC System Documentation in Redmind or in the SOC Documentation binder.

* + 1. H-Bridge

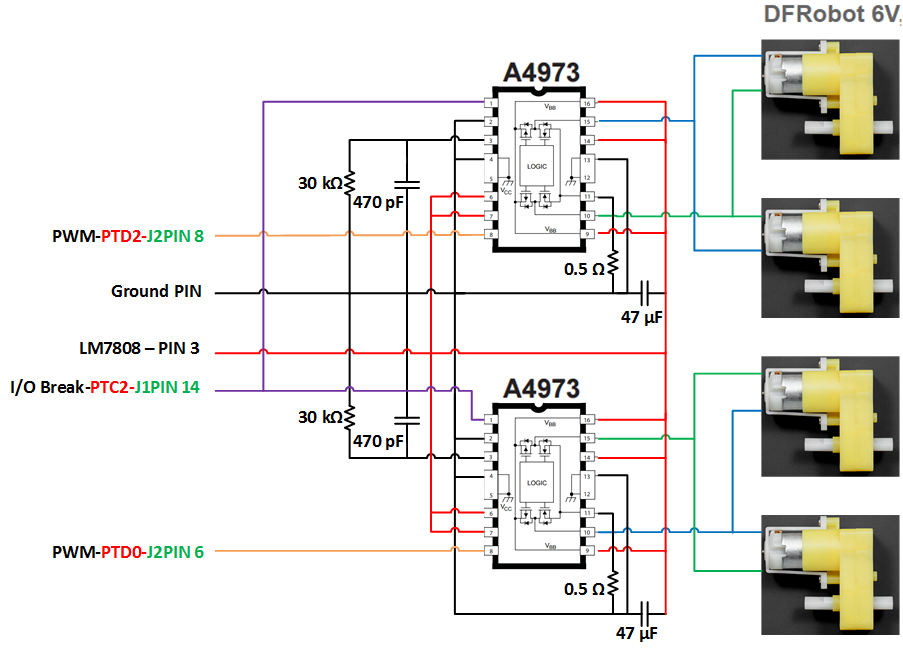


Figure 1 - H-Bridge Wiring

* + - 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. Costume Steering design

|  |
| --- |
|  |

*Figure 4.3.2.1**Customized Steering System*

Figure 4.3.2.1 illustrates the different parts of the steering system. The parts label A, B, C, D, and E are components of the car alignment which keeps both front wheels at the same turning angle. The parts label F and G are the brackets that will hold the front motors and wheel. To make installation easier the holes for the bold (screws) are label from 1 to 8 (The each part has a matching number). The two un-label holes in parts B and C will be use to attach the Steering System to the Car Mobile Platform Described in section 3.2.1.1.

* + 1. Steering Servo (HS-422 Servo)

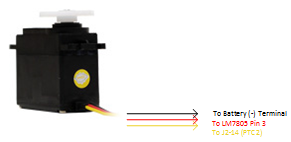


Figure 2: Schematic of the connection of the servo.[3]

As seen in the above schematic, the first pin of the servo is used as a ground. In the scope of this project, this will be connected to the negative terminal of the battery. The second wire will be connected to +5V coming from voltage regulator LM7805 on pin 3. The last wire will be connected to the microcontroller on jumper 2 pin 14.

* + 1. Power Regulator

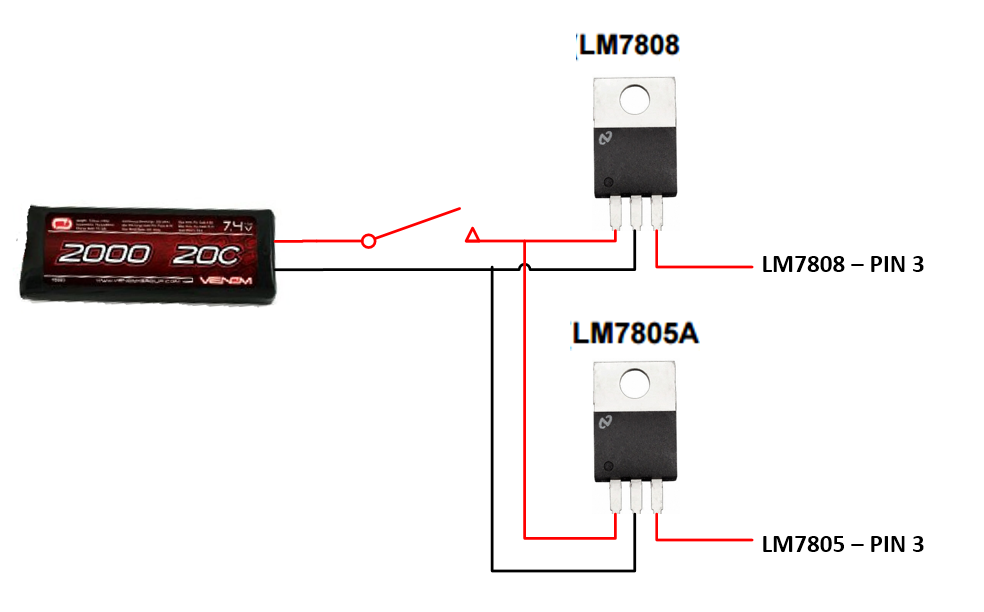


Figure 3: Voltage regulator schematic.[3]

Figure 3 illustrates the voltage regulator wiring scheme. The LM7808 will regulate the voltage to 6.6 volts 1 amp, this voltage will be outputted using the LM7808 Pin 3 to then be used by the H-Bridge load supply. The LM7805 will regulate the voltage to 5 volts 1 amp, this voltage is outputted using the LM7805 Pin 3 to then be used by the steering servo and the microcontroller.

* + 1. Line Scan Camera

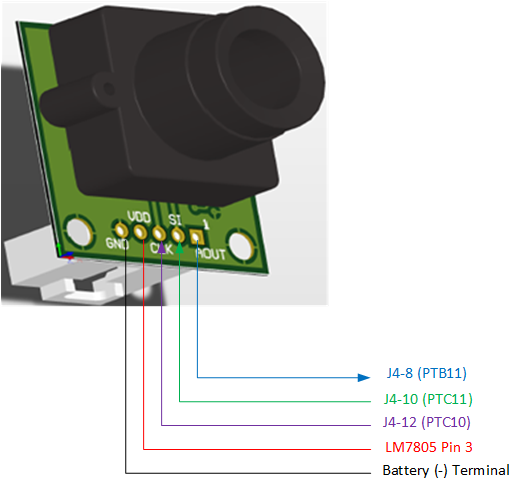


Figure 4: Line scan camera shape and pinout. [3]

Above, the pinouts of the camera are shown, with their corresponding labels. The analog output (blue) will go to the microcontroller on J4-8 to be sampled by the ADC. The SI input (green) is the pulse signal and is connected to J4-10. The CLK input (purple) will be connected to J4-12. The last two pins will come from the voltage regulator (see section 5) and the negative terminal respectively.

The line camera will be mounted on top of the steering brackets using a 30 degree metal sheet plate from the camera mounting kit. This camera bracket will be placed 0.25" from the front of the car chassis (Where the steering bracket is located) and 0.5" from the pivot point of each wheel in the steering bracket. The steering bracket was described in detail in SOC-SysDes-2.

* + 1. CMUCam5 Pixy Vision Sensor Camera

Below *Figure* 11 shows how the marker camera is connected to the microcontroller. I2C serial clock (SCL), pin 5 of pixy camera is connected to the J2PIN 20 of microcontroller. I2C serial data (SDA), pin 6 of pixy camera is connected to the J2PIN 18 of microcontroller. The pin 2, on the pixy camera accepts 5V for powering the camera which is connected to the J3PIN 10 of microcontroller to power the camera through the microcontroller. Note, if the microcontroller is not powered via micro-USB this powering method will not work. Alternate would be to power the pixy through voltage supply. Pins 6, 8, and 10 of marker camera are grounded using J3PIN 14.

|  |  |
| --- | --- |
|  | del  Figure 5: I/O pins and wire connection between marker camera and microcontroller. [3] |

The Marker camera will be mounted on the top level (Second raised plate) of the car chassis 1” from each side and 0.25” from the front side using pan-tilt mechanism.

* + 1. SD Card

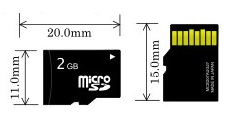


Figure 6: Schematic and sizing of a microSD memory card. [3]

The diagram above shows the size specification of a microSD card. As shown, it is considerably small in size. Also shown are 8 connectors on the right half of the figure. The two elongated pins, 4 and 6 counted from the left, are used for the primary and reference voltage (ground). More specifications of the microSD card connections see reference 3, SOC-SysDes-3.

These connections take no space on the jumpers of the microcontroller. Instead, the microcontroller has an onboard connector for a microSD card. The schematic of which is shown below.



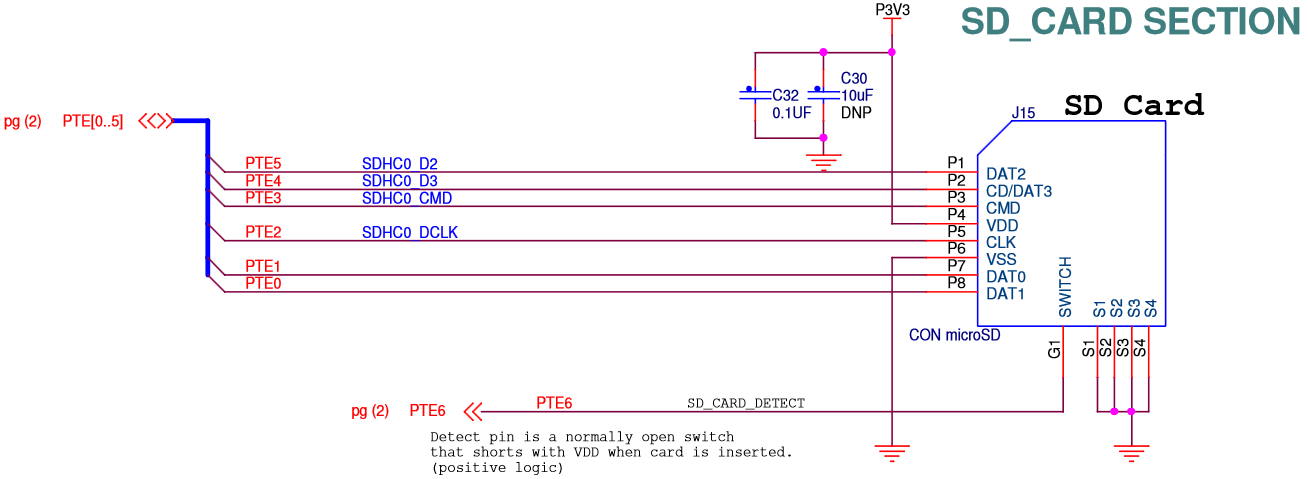


Figure7: Schematic of the connector for the microSD.[3]

All the grounds here are connected internally to the microcontroller so again, no positions on the jumpers of the microcontroller are taken up for the microSD card. Also, the pullup and ground for Vdd and Vss are already connected by default so that means there is less setup for the microSD when doing the implementation.

* + 1. BlueSMiRF Bluetooth

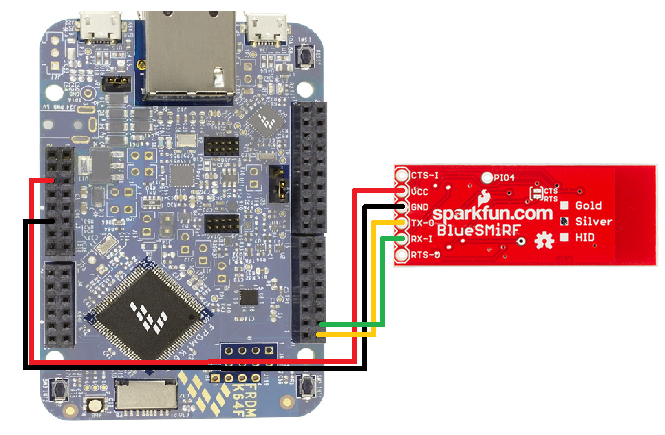


Figure 8: I/O pins and wire connection between bluetooth dongle and microcontroller. [3]

Above figure shows the connection between microcontroller and bluetooth. It has four wires VCC, ground, transmit (Tx), and receive (Rx). The Tx on bluetooth is connected to Rx (PTC16) of microcontroller. The Rx on bluetooth is connected to Tx (PTC17) of microcontroller.

* + 1. Blinkers and Brake Light

The LED’s used in the car are standard Yellow and Red color LED’s powered from the microcontroller. They are operated by setting a JPIN to a logical 1. They Output a logical 1 in a JPIN is 3.3V.