

Lafayette College | Electrical and Computer Engineering

Motor Controller Cooling System (MCCS) User Manual

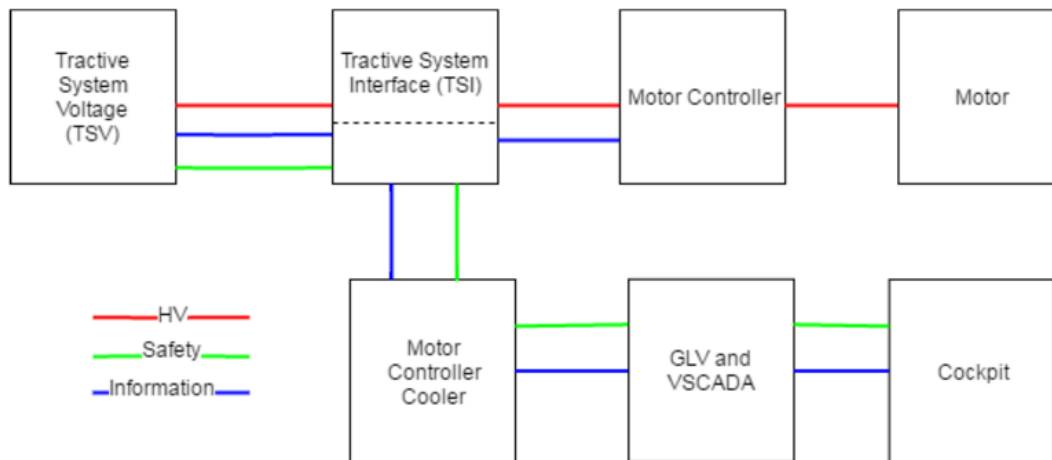
ECE 492 Spring 2017

Table of Contents

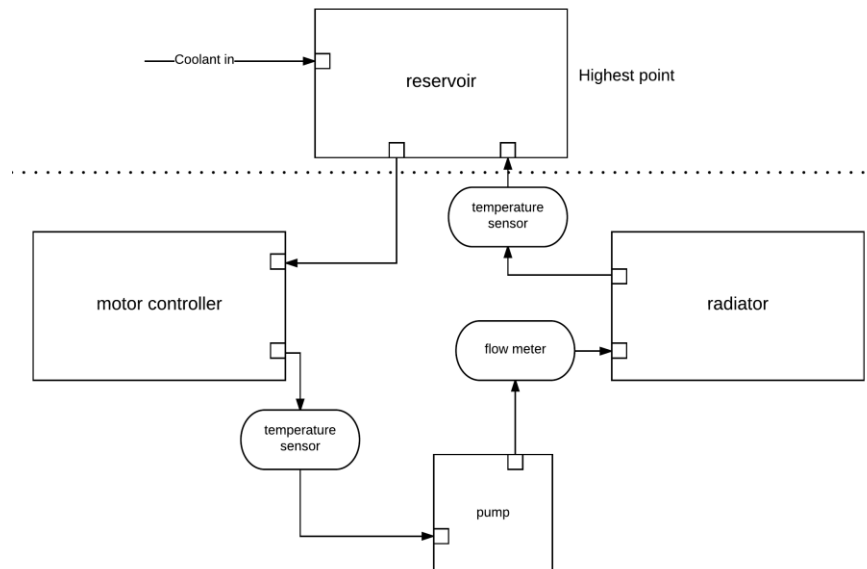
| | |
|---|----|
| Overview | 3 |
| Safety | 5 |
| Electrical System Components | 5 |
| Software Design Approach | 7 |
| Getting Started & System Operation | 8 |
| PANNEL 1: | 10 |
| SAFETY LOOP Panel (Left Side) – J1 | 10 |
| SAFETY LOOP Panel (Middle) – J3 | 10 |
| CAN and POWER Panel (Right Side) – J4 | 11 |
| PANNEL 2: | 11 |
| Pump and Sensors (left side) – J6 | 11 |
| Pump and Sensors (Middle) – J5 | 11 |
| CAN and POWER Panel (Right Side) – J2 | 12 |
| Troubleshooting | 12 |
| Maintenance | 13 |
| Current Status/Future Work | 14 |

Overview

This Motor Controller Cooling System (MCCS will be used in the rest of this document) User Manual documents detailed technical information of the MCCS within the LFEV 2017 Electric Car project. Functionalities, usage of the controller (Arduino) code, software design as well as connections (IC) to the outside system will all be covered.



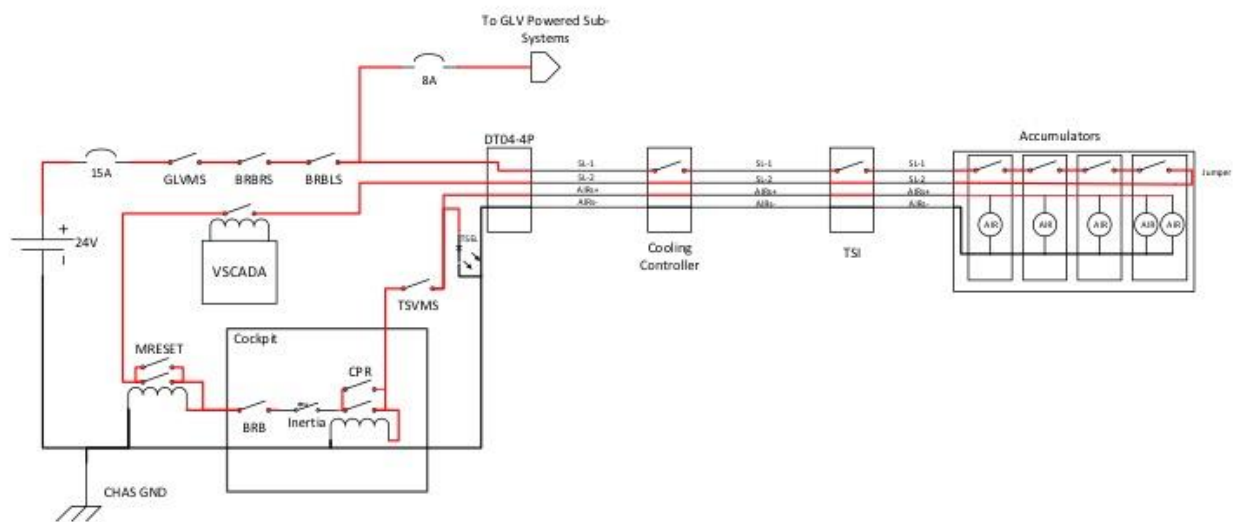
(Overall Simplified System Connection Diagram)



(Coolant Flow Diagram/Connection of Cooling Components)

MCCS was a new cooling system started in spring 2017, despite the existing commercial *Koolance* water cooling system was able to provide efficient cooling to the motor controller in the Dyno Room. In addition to the basic commercial cooling function, the new *MCCS* system is able to:

1. Use CAN bus interface to transfer crucial sensors' data to *VSCADA* (Vehicle Supervisory Control and Data Acquisition) system.
2. Interface with the *GLV* (Ground Low Voltage) safety loop by using a relay. The connection is as shown in the figure below. (Detailed description of the *GLV* safety loop can be found in *GLV* documentations.)



3. Display crucial sensors' data and system status (Low/High) on a SparkFun Serial Enabled 20x4 LCD connected to the central controller (Arduino Uno).
4. Automatic fan/pump speed control based on preset/customizable parameters.
5. Customizable temperature threshold, fan speed percentage and mode selection (High/Low) during run-time without user recompiling the Arduino C code.

Arduino code can be found on LFEV's GitHub [repo](#). While the other documents like *ICD* (Interface Control Document), maintenance manual, etc. can be found on the LFEV-Year-5 *Lafayette Sites* [website](#). (Which can also be easily accessed by scanning the QR code on any



smartphone below.

Safety

The 2017 LFEV team utilized the safety procedures and guidelines set forth by previous LFEV teams. In general, personal protective equipment (PPE; consisting of eye protection, cotton clothing, close-toed shoes, etc) was utilized in a designated workspace with a 1 m barrier distance with a safety watch procedure enacted each time a pack was opened.

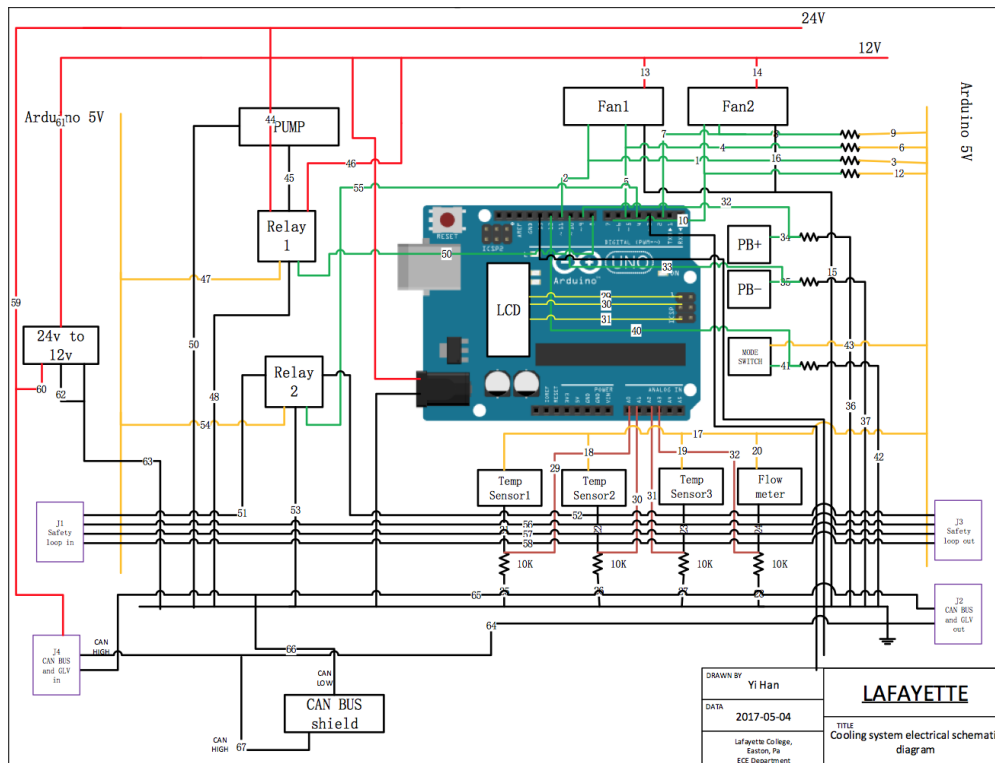
Safety documents detailing procedures, expectations and guidelines produced by previous LFEV teams and to which the 2017 LFEV team adhered can be found here:

<http://sites.lafayette.edu/ece492-sp15/files/2015/02/SafetyPlan.pdf>

<https://sites.lafayette.edu/ece492-sp16/files/2016/03/2016SafetyPlan.pdf>

Electrical System Components

This diagram shows all the electrical connections on the central controller, one Arduino UNO R3. The detailed explanations of how internally/externally the pins are connected to various parts can be found in ICD documents of MCCS.



Explanation of some important parts:

1. Pump:

- A *Koolance* 24V BLDC (brushless DC) pump is used since the overall system should be running on 24V DC GLV power. Due to the lack of control pin on this specific BLDC pump, conventional motor shield on Arduinos cannot be used; thus, in order to control the speed of the pump, a 24V relay is used. Depending on LOW/HIGH mode, the relay will direct 12V/24V to the pump, so speed control is possible.

2. Fan/Radiator:

- Fans are working on 12V DC power, so a DC-DC 24V-12V converter is being used to convert the overall 24V GLV power to 12V.

3. Relay(s):

- The [*KEYES 5V Relay Module*](#) is powered by 5V output power from Arduino UNO R3 and controlled by a digital pin from Arduino UNO R3. The maximum DC current and voltage are 10A and 30VDC, so it can be used to switch 24V/12V DC voltage and be used to tap into the 24V GLV safety loop.

4. Low Power Components:

- Parts like pushbuttons, switches and different types of sensors can be directly powered by 5V Arduino output power.

5. Arduino:
 - Arduino UNO R3 can only source a total of 200mA; operating voltage should not exceed 12V.
6. SparkFun CAN Bus Shield:
 - Uses the Microchip MCP2515 CAN controller with the MCP2551 CAN transceiver. CAN connection is via a standard 9-way sub-D for use with OBD-II cable. The shield also has a serial LCD connector.
7. SerLCD:
 - Communication with SerLCD requires 5V TTL serial at a default baud rate of 9600bps (8-N-1). You can adjust the baud to any standard rate between 2400 and 38400 bps. The power (VDD), ground (GND) and RX pins are all broken out to both a 0.1" pitch header as well as a 3-pin JST connector. The SerLCD should be directly connected to the LCD ports on the SparkFun CAN Bus Shield.

Software Design Approach

The idea is to integrate all sensors/PWM control pins/digital pins/analog pins/CAN communication/LCD interface on the same board, also methods (functions if you like to call them) should be running in parallel by using *millis()*, in other words, default *delay()* function by Arduino should not be used except during Arduino *setup()* period.

The code is pretty detailed commented, but further questions/bug reports can be opened as an *Issue* on LFEV Year 5 GitHub Cooling Repo. Here is a list of the major functions:

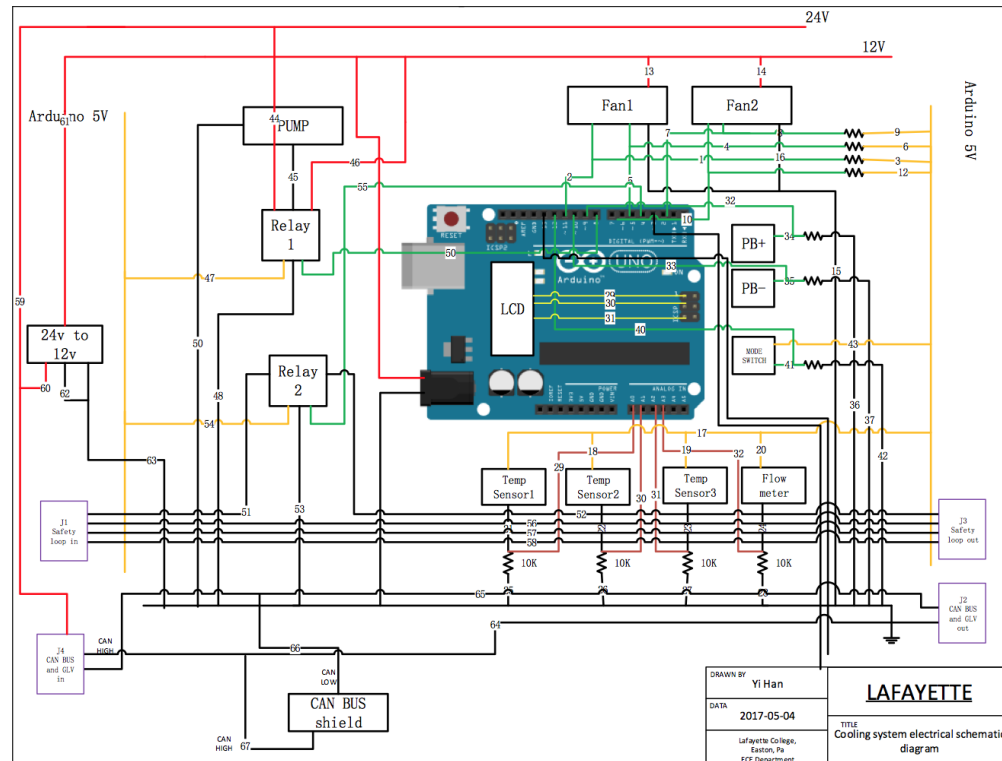
1. *detectFlowmeterRotation(FLOW_PIN) & calculateFlowmeterFrequency()*:
 - Work together to read in calculate flow frequency
2. *FlowThermistorReadAndPrint() & FlatThermistorReadAndPrint()*:
 - Read in and calculate temperature readings of thermistors.
3. *FanModeControl()*:
 - Control operating mode of fans.
4. *FanReader()*:
 - Read in fan speed.
5. *ShowOnLCD()*:
 - Print out crucial data on SerLCD.
6. *CANwrite()*:
 - Using MCP2515 CAN chip to output formatted data following CAN ICD document with specific head IDs and transmission rate.
7. *SafetyLoopController()*:
 - Open safety loop when necessary.

8. PumpShifter():

- Shift power supply to pump between 24V and 12V.

Getting Started & System Operation

- Pre-requisite/System Setup:
 1. Arduino UNO R3, Arduino USB connector and Sparkfun CAN Bus Shield (with stackable headers) / other CAN Bus shield with MCP2515 or similar CAN communication chip, but the digital pins usage may be different.
 2. Arduino IDE and any operating system can run Arduino IDE.
- Internal System Connection:
 - Can be found in MCCS Internal ICD. But here is a schematic diagram:



| Wire Number | Conn A | Conn B | Voltage |
|-------------|-----------------|--------------|---------|
| 1 | FAN1 PWM | 10K Resistor | N/A |
| 2 | FAN1 PWM | DIGITAL 11 | N/A |
| 3 | 10K Resistor | Arduino 5V | 5V |
| 4 | FAN1 Tachometer | 10K Resistor | N/A |
| 5 | FAN1 Tachometer | DIGITAL 5 | N/A |

| | | | |
|----|-------------------|------------------|-----|
| 6 | 10K Resistor | Arduino 5V | 5V |
| 7 | FAN2 Tachometer | 10K Resistor | N/A |
| 8 | FAN2 Tachometer | DIGITAL 2 | N/A |
| 9 | 10K Resistor | Arduino 5V | 5V |
| 10 | FAN2 PWM | DIGITAL 6 | N/A |
| 11 | FAN2 PWM | 10K Resistor | N/A |
| 12 | 10K Resistor | Arduino 5V | 5V |
| 13 | FAN1 Power | 12V | 12V |
| 14 | FAN2 Power | 12V | 12V |
| 15 | FAN1 GND | GND | GND |
| 16 | FAN2 GND | GND | GND |
| 17 | TEMP SENSOR1 PIN1 | Arduino 5V | 5V |
| 18 | TEMP SENSOR2 PIN1 | Arduino 5V | 5V |
| 19 | TEMP SENSOR3 PIN1 | Arduino 5V | 5V |
| 20 | FLOW METER PIN1 | Arduino 5V | 5V |
| 21 | TEMP SENSOR1 PIN2 | 10K Resistor | N/A |
| 22 | TEMP SENSOR2 PIN2 | 10K Resistor | N/A |
| 23 | TEMP SENSOR3 PIN2 | 10K Resistor | N/A |
| 24 | FLOW METER PIN2 | 10K Resistor | N/A |
| 25 | TEMP SENSOR1 PIN2 | A0 | N/A |
| 26 | TEMP SENSOR2 PIN2 | A1 | N/A |
| 27 | TEMP SENSOR3 PIN2 | A2 | N/A |
| 28 | FLOW METER PIN2 | A3 | N/A |
| 29 | LCD TX | RX | N/A |
| 30 | LCD + | LCD + on Arduino | N/A |
| 31 | LCD - | LCD - on Arduino | N/A |
| 32 | PB1 GND | DIGITAL 9 | N/A |
| 33 | PB2 GND | DIGITAL 10 | N/A |
| 34 | PB1 GND | 10K Resistor | N/A |
| 35 | PB2 GND | 10K Resistor | N/A |
| 36 | 10K Resistor | GND | GND |
| 37 | 10K Resistor | GND | GND |
| 38 | PB1 POWER | Arduino 5V | 5V |
| 39 | PB2 POWER | Arduino 5V | 5V |
| 40 | MODE SWICH GND | DIGITAL 12 | N/A |
| 41 | MODE SWICH GND | 10K Resistor | N/A |
| 42 | 10K Resistor | GND | GND |
| 43 | MODE SWICH POWER | Arduino 5V | 5V |
| 44 | RELAY1 PIN1 | 24V | 24V |
| 45 | RELAY1 PIN2 | PUMP POWER | N/A |
| 46 | RELAY1 PIN3 | 12V | 12V |

| | | | |
|----|--------------------|--------------------|-----|
| 47 | RELAY1 POWER | Arduino 5V | 5V |
| 48 | RELAY1 GND | GND | GND |
| 49 | RELAY1 S | DIGITAL 8 | N/A |
| 50 | PUMP GND | GND | GND |
| 51 | RELAY2 PIN1 | SAFETY LOOP1 IN | N/A |
| 52 | RELAY2 PIN2 | SAFETY LOOP1 OUT | N/A |
| 53 | RELAY2 GND | GND | GND |
| 54 | RELAY2 POWER | Arduino 5V | 5V |
| 55 | RELAY2 S | DIGITAL 4 | N/A |
| 56 | J1 | J5 | N/A |
| 57 | J1 | J5 | N/A |
| 58 | J1 | J5 | N/A |
| 59 | J4 | 24V | N/A |
| 60 | 24V | DC TO DC CONVERTOR | 24V |
| 61 | DC TO DC CONVERTOR | 12V | 12V |
| 62 | DC TO DC CONVERTOR | GND | GND |
| 63 | DC TO DC CONVERTOR | GND | GND |
| 64 | J4 | J2 | N/A |
| 65 | J4 | J2 | N/A |
| 66 | CAN BUS SHILED | J2 | N/A |
| 67 | CAN BUS SHILED | J2 | N/A |

- External System Connection:

PANNEL 1:

SAFETY LOOP Panel (Left Side) – J1

Connector Type: DT04-4P Panel Mount

Pin 1: Safety loop input wire 1

Pin 2: Safety loop input wire 2

Pin 3: Safety loop input wire 3

Pin 4: Safety loop input wire 4

SAFETY LOOP Panel (Middle) – J3

Connector Type: DT04-4P Panel Mount

Pin 1: Safety loop output wire 1

Pin 2: Safety loop output wire 2

Pin 3: Safety loop output wire 3

Pin 4: Safety loop output wire 4

CAN and POWER Panel (Right Side) – J4

Connector Type: DT04-6P Panel Mount

Pin 1: CAN H

Pin 2: CAN L

Pin 3: SHIELD

Pin 4: CHGND

Pin 5: 24VDC

Pin 6: 24V RTN

PANNEL 2:

Pump and Sensors (left side) – J6

Connector Type: DT04-8P Panel Mount

Pin 1: Pump POWER

Pin 2: Pump GND

Pin 3: Sensors 5V

Pin 4: TEMP SENSOR 1 INPUT

Pin 5: TEMP SENSOR 2 INPUT

Pin 6: TEMP SENSOR 3 INPUT

Pin 7: FLOW METER INPUT

Pin 8: N/A

Pump and Sensors (Middle) – J5

Connector Type: DT04-8P Panel Mount

Pin 1: FANs POWER

Pin 2: FANs GND

Pin 3: FAN1 PWM

Pin 4: FAN2 PWM

Pin 5: FAN3 tachometer

Pin 6: FAN4 tachometer

Pin 7: N/A

Pin 8: N/A

CAN and POWER Panel (Right Side) – J2

Connector Type: DT04-6P Panel Mount

Pin 1: CAN H

Pin 2: CAN L

Pin 3: SHIELD

Pin 4: CHGND

Pin 5: 24VDC

Pin 6: 24V RTN

Troubleshooting

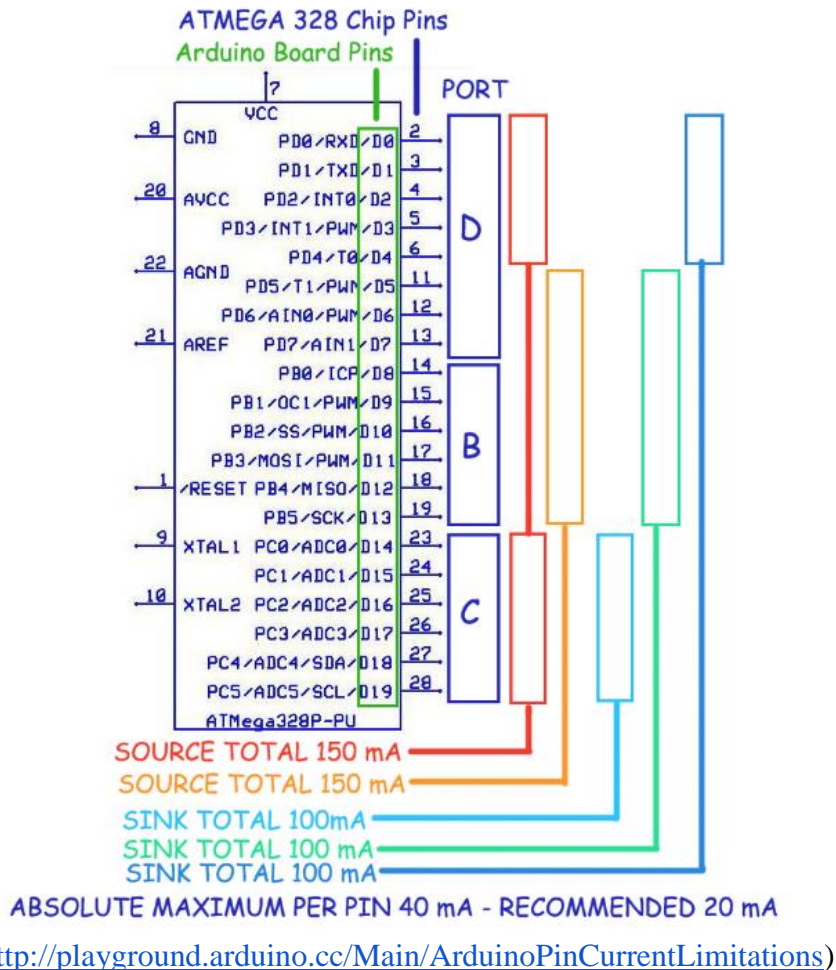
Arduino:

1. Check whether Arduino driver is successfully installed.
2. Check whether COM Port is set to the actual Arduino COM Port.
3. Make sure have the SparkFun CAN libray integrated with Arduino.

Board:

1. Resistors should be used when connect Arduino with sensors. Detailed connection construction can be found in *ICD*, which can be found on LFEV website also.
2. Note: for Arduinos with 1 VCC pin can source a total of 200mA; Arduino UNO is one of the ones with only 1 VCC pin.

ARDUINO UNO / Duemilanove CURRENT SOURCE/SINK LIMITS



Maintenance

1. Pump:
 - Should be connected to 24V DC power.
2. Fan/Radiator:
 - Fans are working on 12V DC power, so a DC-DC 24V-12V converter is being used to convert the overall 24V GLV power to 12V.
3. Relay(s):
 - The [KEYES 5V Relay Module](#) is powered by 5V output power from Arduino UNO R3 and controlled by a digital pin from Arduino UNO R3. The maximum DC current and voltage are 10A and 30VDC, so it can be used to switch 24V/12V DC voltage and be used to tap into the 24V GLV safety loop.
4. Low Power Components:
 - Parts like pushbuttons, switches and different types of sensors can be directly powered by 5V Arduino output power.

5. Arduino:
 - Arduino UNO R3 can only source a total of 200mA; operating voltage should not exceed 12V.
6. SparkFun CAN Bus Shield:
 - Directly stacked on Arduino UNO R3, no other installation needed.
7. SerLCD:
 - Powered by CAN Bus Shield, do not need extra power.

Current Status/Future Work

Even though all individual functions were completed, there wasn't a change for MCCS team to fully integrate all the parts on one soldered board/PCB board and integrate with the "car" downstairs. In the future work, the suggestion is to put all parts together and test them and then if possible, make a PCB board and connect everything to the actual vehicle.