**Team Member Responsibilities:**

Christian Alcalde:

* Proximity sensor drivers
* Obstacle detection and avoidance programs
* Power

Abdullah Wardak:

* IR sensor drivers
* Line/path following program
* CPU

Eric Fong:

* Motor control hardware and drivers
* Accelerometer drivers
* Positional feedback mapping w/ accelerometer

**Project Name:** Autonomous Car

Our main goal is to create a car that follows a drawn path on the ground. It would also have a method for avoiding obstacles and moving pass them. Power will be provided to the car through a battery holder connected to the processor. Standard batteries will be used, although we do not know which size. A switch on the car will turn the car on and off.

A simple project car chassis will be the base of our vehicle. It comes with four wheels and four DC motors. If the motors fail to support the weight of all peripherals, we will replace them with stronger motors. Motor speed and direction control will be implemented using PWM to adjust motor speed, H-bridge switching, and associated driver software.

Line detection with infrared sensors will work by monitoring reflective values of three spots in front of the car, one in the front-middle and one each to either side. If the middle sensor and one of the side sensors detect a change in reflection, the robot will readjust.

Steering can be approached one of two ways. The first implementation would use a servo to angle the two front wheels in the direction to turn in. The second implementation would turn the car by running the left and right side motors at different speeds (i.e. left motor in reverse with right motor in forward produces counter-clockwise turning); the differing motor speeds would be facilitated by expanding off of the motor speed and direction control hardware and software.

To detect obstacles, we will implement several ultrasonic distance sensors that will detect any objects in the way of the car. By connecting the sensors through GPIO serial ports on our processor, the car will react accordingly. If the sensors detect an object within three inches, the car will be programmed to reverse and move past the obstacle. Collision avoidance will be triggered by obstacle detection, and will be implemented using accelerometers and proximity sensors. The proximity sensors will feed data to the CPU to determine a possible route around the obstruction. For a base goal we may assume that most obstacles require no more than about a two-meter-long detour and can be circumvented in a circular arc, more complex pathing algorithms would be a stretch goal. Accelerometers would be used to obtain movement data that the CPU can calculate some rudimentary mapping data from to then be used in adjusting course back onto the original path after executing collision avoidance maneuvers.

Base Objectives:

* car - controls motors and steering
* variable speed, reverse gear motor control
* path detection and reaction
  + obstacle detection
  + collision avoidance
* travel logging (how far did I travel, where is my destination relative to my start, etc.)
  + travel statistics (average speed,)

Stretch Goals:

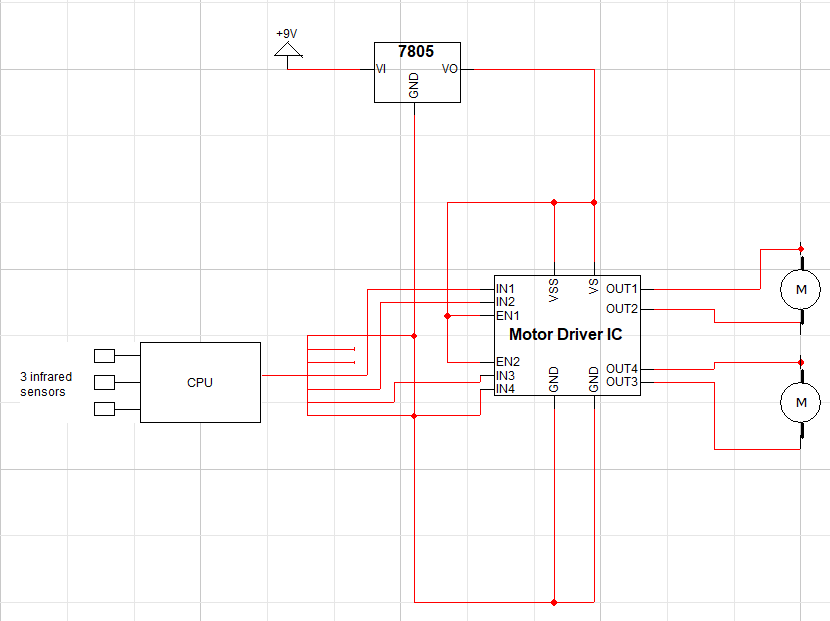
● remote control

● obstacle reaction

● reading road signs

Parts to look for:

* Accelerometer
* IR sensor (line detection)
* Chassis
* Distance sensor (obstacle detection)
  + Ultrasonic
  + Radar?
  + IR could possibly work too?
* Motors (DC brushed)
  + H-bridge for reversing car



For motor driver IC, which drives the motors we will be using (DC brushed) H-bridge.

7805 is the voltage input and output. Since in this class we are not allowed to use arduino, we must build it.

Possible parts:

* <https://www.amazon.com/wheel-layer-Chassis-Encoder-Arduino/dp/B06VTP8XBQ/ref=sr_1_2?crid=159RPZ9FC7T3L&keywords=car+chassis&qid=1567805053&s=gateway&sprefix=car+chas%2Caps%2C202&sr=8-2>
* <https://www.sparkfun.com/products/11769>