HIMALAYAN MAKERS GUILD

Self-Driving Car Project Definition and Plan

The self-driving car project will challenge students to make cars safer and more accessible using robotics. The project should take **~22 hours** to complete, with a proposed **6 month** schedule. It is intended for students aged 10 to 16 who have some background knowledge of electronics (e.g. they have completed the HMG Foundation Activities) but can easily be adapted for an older audience by adding complexity to the design challenge part of the project.

TABLE OF CONTENTS

THE PROBLEM – CAR ACCIDENTS

Millions of car accidents happen every year, harming people and damaging property. Driving while tired, distracted (talking on the phone), or intoxicated by drugs or alcohol increase the likelihood of accidents.

Car accidents harm people and damage property.

What are some ways to reduce car accidents, or the harm caused by them?

THE SOLUTION – SELF-DRIVING CARS

There are many possible solutions to the problem of car accidents, including reducing the use of cars (instead, travel by bike, or train); in-car safety technologies; and driving rules or regulations. We've chosen to explore self-driving cars as a possible way to reduce the harm caused by car accidents.

Self-driving cars use electronics and programming to drive more safely and efficiently. In some cases, they react more quickly than humans in emergencies. They can work together to be more energy and time efficient. Self-driving cars could also be more accessible to people with disabilities.

Unlike humans, self-driving cars would not have accidents due to exhaustion, intoxication, or distraction. They could also follow the rules of the road strictly to help prevent accidents.

[1]

Self-driving cars also present some challenges: they may be expensive to make, could be extremely dangerous if there is a technical error, and would have to make difficult ethical decisions in emergencies possibly deciding who should live and who should die.

We will be building a small self-driving car using a microcontroller, 2 motors, and a distance sensor.

[1] Car icon made by Smashicons from www.flaticon.com, licensed by CC 3.0 BY

NEEDS

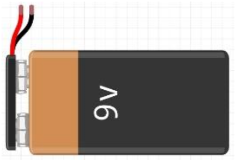
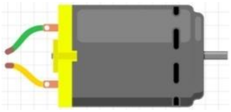
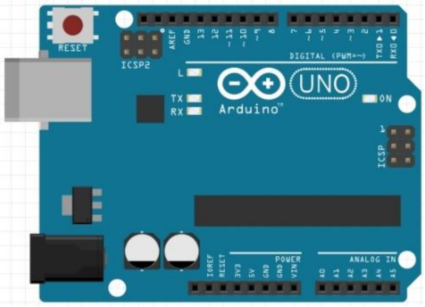
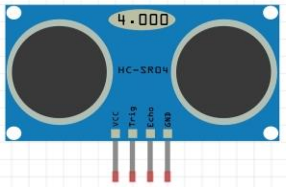
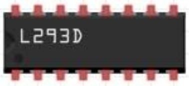
The car must:

1. Stop and turn around, 10cm – 20cm before hitting an object 5cm or taller in height, then continue moving forward.
2. Run continuously for at least 2 minutes on a fully charged battery.

CONSTRAINTS

Some limitations for our design include:

- Time: The car must be built in fewer than seven 1.5-hour activity sessions
- Materials:

1x	 [2]	9V Battery
2x		Geared DC Motor
1x	 RESET ICSP2 DIGITAL (PWM~) UNO Arduino ICSP POWER ANALOG IN	Arduino UNO Microcontroller
1x		Ultrasonic Sensor
1x		H-Bridge IC

Basic electronic components (wires, resistors, capacitors, LEDs, etc.), breadboards, and basic building supplies (cardboard, glue, tape, etc.) will also be available for building the car.

[2] Part Images from Fritzing

MATERIALS AND COSTS PER GROUP

These costs are for the basic prototype described in §Needs and Constraints and §Building the Example Prototype and do not include costs associated with the §Design Challenge part of the project (e.g. other sensors, servo motors etc.). Design challenge costs are expected to be $10 CAD per group.

Assuming one kit of parts per group of 3-4 students:

Item	Qty.	Cost per Group[1]	Expendable[2]	Supplier
LED Assorted 3mm 5mm	2	0.04	Yes	AliExpress
H Bridge Motor Driver L293D	1	0.31	Yes	AliExpress
Capacitor 100uF Electrolytic	2	0.08	Yes	AliExpress
Capacitor 0.1uF Ceramic	2	0.03	Yes	AliExpress
Resistors Assorted 2100pcs	2	0.01	Yes	AliExpress
9V Battery Snap	1	0.16		AliExpress
Prototype Board UNO Shield	1	1.79		AliExpress
Jumper cables MM MF FF 10cm	20	0.39	Yes	AliExpress
Arduino UNO with cable	1	6.62		AliExpress
9V Ni-Mh 450mAh	1	5.17		AliExpress
Motor, DC 6V, TT	2	3.85		Ason, Kathmandu
Castor Wheel, Small	1	1.28		Himalayan Solution
Total Cost per Group		**$18.56 CAD**		

1. *Currency is CAD, 2017-06-10. Assuming one set of parts per student.*
2. *Likely to be broken or lost during the activity.*

Additional items that should be provided include building materials for the body and wheels of the car such as cardboard, thin wood, paper, old pens, recycled plastic containers, straws, popsicle sticks, tape, glue, wire coat-hangers, and beads. Also, tools such as scissors, box-cutters, wire cutters, and rulers should be available for use.

DESIGN CHALLENGE

The students will work in groups to create the same basic self-driving car prototype as described in §Needs and Constraints and §Building the Example Prototype. Each group will then be challenged to improve their design by adding extra functionality. Each group will try to implement a specific improvement. The groups should be encouraged to come up with their own improvements after testing their basic prototypes. However, if they need some extra guidance possible improvements include:

- Car stops before falling off an edge
- Car follows a black line on the floor
- Car follows a wall continuously (around corners)
- Car direction or speed controlled remotely

6 MONTH PLAN

This plan is structured assuming three 1.5-hour project sessions per month. The plan is guided by the Engineering Design Process[3] and refers to it as follows:

- EDP1: **Define** the problem, needs, and constraints
- EDP2: **Research** the problem and related technology
- EDP3: **Imagine** possible solutions
- EDP4: **Select** a promising solution
- EDP5: **Build** a prototype
- EDP6: **Test** the prototype
- EDP7: **Improve** the solution

Before building the basic prototype the students should be introduced to key concepts through hands-on activities (see §Topic Activity Resources for more information):

- DC motors (how they work, electromagnetic induction)
- DC motors (forward/backwards control... H-Bridge concept)
- H-Bridge IC (use in a breadboard to control a motor)
- Arduino Programming (basic introduction to program structure, input, output)
- Ultrasonic sensor (how it works, basic application... serial monitor print-out)

Where necessary, extra topic activities should be arranged to match the design challenges, for example:

- Servo motors (how they work)
- Using light sensors to detect edges (hardware and programming)
- Object avoidance (basics of programming to get around things)

[3] https://www.teachengineering.org/k12engineering/designprocess

MONTH 1 – INTRODUCTION

1. Introduce the problem (EDP1), form the groups, and discuss possible solutions (EDP3)
2. Choose the best solution (EDP4) and present the example prototype of the car students will build (see §Building the Example Prototype). Discuss the needs and constraints of the car (EDP1)
3. Topic activities (EDP2, EDP5): DC Motors 1 – Electromagnetic Induction

MONTH 2 – TOPIC ACTIVITIES

4. Topic activities (EDP2, EDP5): DC Motors 2 – Controlling Motor direction with an H-Bridge
5. Topic activities (EDP2, EDP5): DC Motors 3 – H-Bridge IC
6. Topic activities (EDP2, EDP5): Programming an Arduino to Control an H-Bridge (basic introduction to program structure, input, output)

MONTH 3 – BUILD THE PROTOTYPE

7. Topic activities (EDP2, EDP5): Ultrasonic sensor (how it works, basic application, serial monitor print-out of distance)
8. Build the prototype (EDP5)

MONTH 4 – TEST THE PROTOTYPE

9. Build the prototype, continued (EDP5)
10. Test the prototype (EDP6)
11. Improve the prototype (EDP7)

MONTH 5 – DESIGN CHALLENGE

12. Improve the prototype (EDP7)
13. Prepare for exhibition of project

MONTH 6 – PROJECT EXHIBITION

14. Exhibit the project

BUILDING THE EXAMPLE PROTOTYPE

An example prototype of the project should be built for demonstration to the students in Month 1 of the project. This design can also be used as a reference helping the students troubleshoot their prototypes.

To be done: Diagram, breadboard, photos, etc.

TOPIC ACTIVITY RESOURCES

The following resources can be used when preparing for the topic activities:

- DC Motors 1 – Electromagnetic Induction
- DC Motors 2 – Controlling Motor direction with an H-Bridge
- DC Motors 3 – H-Bridge IC
- Programming an Arduino to Control an H-Bridge (basic introduction to program structure, input, output)
- Text based programming for the Arduino (transition from using block programming tools to editing an Arduino sketch directly)
- Ultrasonic sensor (how it works, basic application, serial monitor print-out of distance)

DC MOTORS 1 – ELECTROMAGNETIC INDUCTION

Activity Goal: Explore how DC motors use electricity to make mechanical motion.

Hands-On: Demonstration of electromagnet (wire wrapped around a nail), and a simple DC coil motor.

Resources:

- https://learn.sparkfun.com/tutorials/motors-and-selecting-the-right-one
- https://www.wired.com/2016/01/how-to-build-a-super-simple-electric-motor-out-of-stuff-you-already-have/

DC MOTORS 2 – CONTROLLING MOTOR DIRECTION WITH AN H-BRIDGE

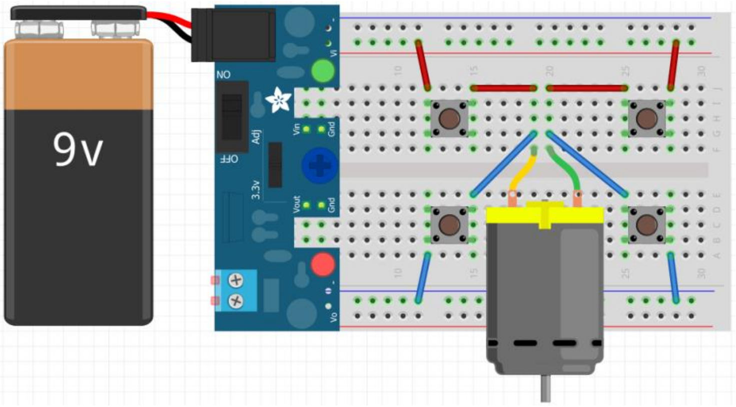
Activity Goal: Understand how to control the rotation direction of a DC motor.

Hands-On: Change direction of DC motor by changing terminals between 5V and GND. Build an H-Bridge control circuit using push buttons on a breadboard.

Resources:

- http://www.mcmanis.com/chuck/robotics/tutorial/h-bridge/
- https://www.rakeshmondal.info/L293D-Motor-Driver

- An example of the hands-on circuit, an H-Bridge built using push-buttons.



DC MOTORS 3 – H-BRIDGE IC

Activity Goal: Relate the H-Bridge concept to an H-Bridge IC and use the IC to control motor direction.

Hands-On: Build a breadboard circuit using the H-Bridge IC to control motor direction.

Resources:

- http://www.mcmanis.com/chuck/robotics/tutorial/h-bridge/
- https://www.rakeshmondal.info/L293D-Motor-Driver
- https://www.elprocus.com/h-bridge-motor-control-circuit-using-l293d-ic/

PROGRAMMING AN ARDUINO TO CONTROL AN H-BRIDGE

Activity Goal: Control the H-Bridge IC using an Arduino Uno.

Hands-On: Program an Arduino UNO using BlocklyDuino[4] to control the direction of two motors, using the H-Bridge IC circuit build on the breadboard during the previous activity.

Resources:

- See the documentation for Foundation Activity “FA4 - Blinking an LED” for review of programming the Arduino.
- https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/
- https://www.bc-robotics.com/tutorials/controlling-dc-motor-arduino/
- https://www.maxphi.com/dc-motor-interfacing-arduino-tutorial
- http://www.learningaboutelectronics.com/Articles/Arduino-microcontroller-H-bridge-circuit.php

TEXT-BASED PROGRAMMING FOR THE ARDUINO

Activity Goal: Transition from using block-based programming tools to writing programs directly in the Arduino IDE.

[4] https://blocklyduino.github.io/BlocklyDuino/blockly/apps/blocklyduino/
Or a similar block-based programming tool.

Hands-On: Take the program from the previous activity (driving an H-Bridge with the Arduino), and look at the text code produced. Define key parts of the code, and alter it to include and LED that turns on when the motor is rotating in one direction, and off when it is rotating in the other direction.

Resources:

- Topics to cover:
    - Structure: setup(), loop(), curly brackets {}, what does "void" mean?, if/else statements, semicolon line endings, comments
    - Variables: int, float
    - Digital IO: pinMode(), digitalWrite()
    - Delay()
- For functions like digitalWrite(), see the documentation on the Arduino Language Reference: https://www.arduino.cc/reference/en/#page-title

ULTRASONIC SENSOR

Activity Goal: Introduce how the SR04 ultrasonic sensor works and program the Arduino to use the sensor to control an LED. Print the current measured distance to the computer screen.

Hands-On: Program an Arduino UNO, using the ultrasonic sensor as an input and calculating distance. Print distance result to the serial monitor for debugging. Turn on an LED when an object comes closer than 20cm using if/else statements

Resources:

- Putting a 1uF capacitor between 5V and GND, close to the sensor, can help stabilize the readings.
- https://arduinobasics.blogspot.com/2012/11/hc-sr04-ultrasonic-sensor.html