6V Ride-On Car with Modifications (Frozen Edition)

ENGR 202 students: Mohammed Bujbara & Alia Qasem

Service Learning Instructor: Samantha Corcoran

Technical Advisor: Nathan Smith

Spring 2016
Introduction

- Thank you for your interest in modifying a 6V car for sitting with Joystick activation!
- Before beginning, please read the following to ensure the safety of you and your child.
  - We are engineering students from Wichita State University taking ENGR 202 as a credit for engineering 2020. We had a very good opportunity to design a car for a child named Jocelyn. This project was especially for children with special needs. The GoBabyGo project consisted of a group of engineering and physical therapy students as well as advisors.
How to get the most out of your car

⚠️ DO NOT use your car for the first time without having it checked by an electrical engineer or other professional.

⚠️ DO NOT use a battery if the wires or casing shows any sign of damage or modification.

⚠️ DO NOT allow the child to use the car unsupervised.

⚠️ DO NOT use the car outside in any inclement weather or leave the car outside in such weather.

⚠️ DO NOT use the car if you smell or smell smoke.

⚠️ DO NOT leave the toggle switch in the ON position when the car is not in use.

⚠️ DO NOT TAKE ANY UNNECESSARY RISKS WHEN USING OR MODIFYING THE CAR!

⚠️ DO contact a clinician about your child’s use of a modified ride-on car and involve him/her in the modification process and later ride-on car use.
Before Modifying

⚠ Please always wear safety glasses and be sure to wear gloves to remove plastic burrs.

⚠ Always be aware of where your fingers and hand are during the modification process, especially when drilling or using PVC Cutters.

⚠ Do not use wire of a gauge higher than 16. A higher gauge means thinner wire, which may run this risk of causing overheating and wire melting.

⚠ When stripping wire, make sure to cut only through the plastic tubing and not the wire itself. Also, make sure to strip only enough wire so that you do not leave wire exposed when connecting wires together.
  - Cutting too deep into the wire or leaving unprotected wire may cause sparks, short circuits, and overheating. Always use wire strippers and not scissors.
  - Please note that you should never cut wires on the battery.

⚠ Please cover any wire nuts and crimp terminals with electrical tape.
Materials and Tools

**Tools:**
- Power drill
- ½”, ¼” and 1” drill bits
- Wire cutter
- 16 gauge wire stripper
- Wire crimper
- Screwdriver
- Flat head screwdriver
- PVC cutter
- Pliers
- Scissors

**Hardware:**
- Bolts
- Nuts
- (5) ft industrial-strength Velcro

**Electrical:**
- (2) Ring Crimp Connectors
- SPST Toggle Switch
- Wire Nuts
- 12-14 Gauge Wire
- Female Crimp Connectors
- Male Crimp Connectors
- Electrical Tape
- Arduino UNO and motor driver (Rated 5A or higher)
- 10K Ohm Potentiometer
- 6-12 Volt Dc Motor (powerful enough to turn a steering shaft)

**Other Materials:**
- (1) Pool Noodle
- Tape measure
- Kickboard
- Buckle
Install the SPST Switch (Kill Switch)

- **Step 1:** Under the seat, cut the red wire that connects the car to the battery.

- **Step 2:** Using ring crimp connectors, and attach 2 lengths of 14-16 gauge wire. Mount the 2 wires to the SPST switch. Then drill a hole in the back of the car with a battery operated drill (diameter will correspond to switch).

- **Step 3:** Route the SPST wires through the hole into the battery compartment, then mount the switch in the exterior drilled hole.

- **Step 4:** Connect the SPST switch “ON” wire to the cut wire (the side away from the battery connector). Connect the “OFF” wire to the side nearest the battery connector.
Steering Motor Mount

• **Step 1:** Remove the steering wheel, shaft, and front tire mount assembly. Discard the steering wheel and shaft. You will need to cut the “V-shaped” section (that the shaft connected to) completely off using a hacksaw or cut-off wheel. Next drill a hole (the size needed based on the DC motor used) on the left hand side of where the original steering shaft was located. Mount the motor as to leave its shaft facing towards the front of the car.

• **Step 2:** You will need to fashion a steering block, using 2 pieces of block aluminum. The larger block will mount directly to the DC motor shaft, and will need another hole near the bottom to serve as a mounting point for a steering arm. The smaller block will need to mount on the larger block, with a hole in the center, laying directly on top of where the DC Motor shaft is (the purpose of the smaller block is to mount to the knob of a 10K Ohm Potentiometer).
Steering Motor Mount

• **Step 3:** Using the section off the original steering mount, cut off one leg of the “V,” and then twist the leg so that the ends are offset about 90 degrees. Drill a mounting hole at each end. Mount one end to the steering block you have created. Mount the other end to the topside of the original left-hand tire-mount section. Use lock-nuts for these connections, as you will need to leave them slight loose to not cause binding.

• **Step 4:** Using a 10K Ohm POT, insert and secure the knob to smaller aluminum block. Then using some tin strip (that can be slightly flexible), fashion a mounting strap the will hold the body of the POT stable. The purpose is to allow only the knob of the POT to move in relation to the movement of the wheel/DC motor. (We will explain the POT further later in the manual)
Steering Motor Mount

• Step 5: Using 3 lengths of 20-22 gauge wire, connect one end of the wiring harness to the three mounting posts on the POT. The POT harness consists of +5VDC, GND, and Analog out. Then route this wiring harness along the existing wiring trough to the battery compartment.

• Step 6: Using 2 lengths of 14-16 gauge wire, route the harness from the DC Steering Motor positive (+) and Negative (-) to the battery compartment.
Steering Motor Mount
Mounting the Analog Joystick

- **Step 1:** Drill hole of 1” on the right side, in a location that will be accessible to mount, as well as for the child to reach.

- **Step 2:** Create a wiring harness consisting of 4 separate lengths of 20-22 gauge wire that can be routed to the battery compartment. The harness will consist of a +5VDC, GND, Analog out (H), Analog out (V). Once this wiring harness is routed, mount the joystick in desired location, and connect wiring harness to joystick mounting pins.
Mounting the Analog Joystick
Removing the Original accelerator Switch

- **Step 1:** Using a flathead screwdriver remove the switch housing. Completely remove the physical switch. Create a wiring harness with 2 lengths of 14-16 gauge wires that will route back to the battery compartment. These will plug into the original plug in the Battery POS and MTR control spots. Now replace the empty switch housing.
Removing the Original accelerator Switch
Arduino Uno and Motor Driver Shield

- There are numerous connections. Please have an advisor on hand when installing connections.

- From component to Arduino:
  - 10K POT: 1 pin to +5VDC, 1 pin to GND, Center Pin to Analog 0.
  - Analog Joystick: 5V pin to +5VDC, GND to GND, H to Analog 3, V to Analog 5.
  - Motor Driver Board: GND to GND, D2 to Digital 7, M1PWM to Digital 3, M1DIR to Digital 8, M2PWM to Digital 5, M2DIR to Digital 10, and VDD to VDD.
  - Use ride-on car battery to power Arduino.
Arduino Uno and Motor Driver Shield

– From component to Motor Driver Board (Pololu Dual MC33926):
  • Battery POS (+) to VIN, Battery NEG (-) to GND
  • Steering motor (+) to M1A, Steering Motor (-) to M1B
  • Drive Motor (+) to M2A, Drive Motor (-) to M2B
Basic Wiring

*Please note: This is a basic reference only. The driver board we used was slightly different from this depiction.
* Please reference the specific driver board used. The programming may need to be changed dependent on the driver board. Pin assignments may need to change dependent on the driver board.
Arduino Code

//Wichita State University GoBabyGo code
//Nathan Smith: Advising ENGR 202 students for their GoBabyGo build.
//Using a joystick to control drivemotor (Forward/Reverse)
//Using a DC motor with a custom built steering linkage and a 10K OHM potentiometer (for feedback)
//to control the steering.
//*For Public Use*

const int m1_pwm = 3 ; //initializing pin 3 as pwm steering motor
const int m2_pwm = 5 ; //initializing pin 5 as pwm drivemotor
const int m1_dir = 8 ; //steering output
const int m2_dir = 10 ; //drivemotor output
const int d2 = 7; // driver enable pin
int steer_link_Pot = analogRead(A0); //Reads value of steering linkage pot
int joystick_steerPot = analogRead(A3); //Reads value of joysticks H pot
int joystick_drivePot = analogRead(A5); //Reads value of joystick V pot
const int right_threshold = 700; //Sets beginning threshold on joystick H pot
const int left_threshold = 400; //Sets beginning threshold on joystick H pot
const int forward_threshold = 750; //Sets beginning threshold on joystick V pot
const int reverse_threshold = 400; //Sets beginning threshold on joystick V pot
const int pot_thresholdMin = 335; //Sets steering stop limits
const int pot_thresholdMax = 695; //Sets steering stop limits
const int steerpot_center = 504; //provides a center point to return to
const int steering_deadband = 35; //deadband to account for sensitivity
int center_window = 0;
int steerpot_window = 0;
int accel = 0;
void setup()
{
    Serial.begin(9600); // initialize serial communication at 9600 bits per second
}

void loop()
{
    steer_link_Pot = analogRead(A0); //Reads value of 10K pot attached to the steering linkage
    joystick_steerPot = analogRead(A3); //Reads value of joystick H pot
    joystick_drivePot = analogRead(A5); //Reads Value of joystick V pot
    delay (100); //delay set for stability
// For forward movement
if(joystick_drivePot > forward_threshold)
{
  if(accel <= 240) {
    accel = accel + 10 ;
  }
  //digitalWrite(m2_pwm,HIGH) ;
  digitalWrite(m2_dir,LOW) ;
  digitalWrite(d2, HIGH) ;
  analogWrite(m2_pwm,accel) ;
}
// For reverse movement
if(joystick_drivePot < reverse_threshold)
{
    if(accel <= 240) {
        accel = accel + 10;
    }

    // digitalWrite(m2_pwm,HIGH);
    digitalWrite(m2_dir,HIGH);
    digitalWrite(d2, HIGH);
    analogWrite(m2_pwm,accel);
}

//For no forward/reverse movement
if((joystick_drivePot > reverse_threshold)&&(joystick_drivePot < forward_threshold))
{
    accel = 120;
    digitalWrite(m2_pwm,LOW);
    digitalWrite(m2_dir,LOW);
    digitalWrite(d2, HIGH);
    // analogWrite(m2_pwm,100);
    digitalWrite(m2_pwm,100);
}
/For Righthand or CW motion: in_a = High, in_b = Low

if ((joystick_steerPot > right_threshold)&&(steer_link_Pot < pot_thresholdMax))
{
    //digitalWrite(m1_pwm,HIGH) ;
    digitalWrite(m1_dir,LOW) ;
    digitalWrite(d2, HIGH) ;
    analogWrite(m1_pwm,100) ;
    Serial.println("Turn Right");
}


//For Lefthand or CCW motion: in_a = LOW, in_b = HIGH

if ((joystick_steerPot < left_threshold) && (steer_link_Pot > pot_thresholdMin))
{
    //digitalWrite(m1_pwm, HIGH);
    digitalWrite(m1_dir, HIGH);
    digitalWrite(d2, HIGH);
    analogWrite(m1_pwm, 100);
    Serial.println("Turn Left");
}

// States that if joystick value has not met its threshold, then do nothing
if ((joystick_steerPot > left_threshold) & (joystick_steerPot < right_threshold))
{
    digitalWrite(m1_pwm, LOW);
    digitalWrite(m1_dir, LOW);
    digitalWrite(d2, HIGH);
    // analogWrite(m1_pwm, 255);
    Serial.println("Stop Turning 1");
}

// States that if the steering linkage potentiometer meets its limit, then do nothing (keeps it from turning too far in either direction)
else if (steer_link_Pot < pot_thresholdMin || steer_link_Pot > pot_thresholdMax)
{
    digitalWrite(m1_pwm, LOW);
    digitalWrite(m1_dir, LOW);
    digitalWrite(d2, HIGH);
    // analogWrite(m1_pwm, 255);
    Serial.println("Stop Turning 2");
}
Arduino Code

// Moves steering to center
center_window = 1;
steerpot_window = 1;
if (joystick_steerPot > right_threshold) center_window = 0;
if (joystick_steerPot < left_threshold) center_window = 0;

if (center_window == 1)
{
    if (steer_link_Pot > steerpot_center + steering_deadband)
    {
        steerpot_window = 0;
        // digitalWrite(m1_pwm,HIGH);
        digitalWrite(m1_dir,LOW);
        digitalWrite(d2, HIGH);
        analogWrite(m1_pwm, 100);
        Serial.println("Turn Left to Center");
    }
}
if (steer_link_Pot < steerpot_center - steering_deadband) // - steering_deadband
{
    steerpot_window = 0;
    //digitalWrite(m1_pwm,HIGH);
    digitalWrite(m1_dir,HIGH);
    digitalWrite(d2, HIGH);
    analogWrite(m1_pwm,100);
    Serial.println("Turn Right to Center");
}
if (steerpot_window == 1)
{
    digitalWrite(m1_pwm,LOW);
    digitalWrite(m1_dir,LOW);
    digitalWrite(d2, HIGH);
    // analogWrite(m1_pwm,100);
    Serial.println("Stop Centered");
}
// prints the analog values on the serial monitor

Serial.print("Steering Pot Values: ");
Serial.print(steer_link_Pot);
Serial.println(" ");
Serial.print("Joystick H Values: ");
Serial.print(joystick_steerPot);
Serial.println(" ");
Serial.print("Joystick V Values: ");
Serial.print(joystick_drivePot);
Serial.println(" ");
delay(100);   // delay between reads for stability
}


Theory

• The 10K POT is being used to provide an analog value to Arduino. The value is from 0 to 1023, and the number displayed in the serial monitor should reflect which direction and how far the knob is turned. The reason for this is to set a center value, maximum left value, and maximum right value. These values are then used to drop to logic signal to the corresponding pin, when the value is met. This way the motor does not just keep spinning.

  – To find the threshold values: connect to your Laptop and manually turn the wheels center, right, and left. The values you get are what needs to be entered into the code. The values will be different every build.

  » Now that the values are known, this will be used to convert the Analog in to a digital out. Thus sending the logic level needed to the motor driver board.
Theory

• The analog joystick is being used as a means of forward/reverse and directional control. The joystick uses 2 10K Ohm POTS to read the x (Horizontal) and y (Vertical) values. The Analog inputs will be between 0-1023.

  - We enter threshold values to ensure that the joystick is not extremely sensitive. These values should always be the same if using a similar joystick module.

    » Now that the values are known, this will be used to convert the Analog in to a digital out. Thus sending the logic level needed to the motor driver board.
Installation of fitting materials

- Using any webbing of your choice and some plastic buckles, fashion a very simple seatbelt.
- Attach the seatbelt ends to each side of the car's removable seat.
- Using the foam kickboard and an X-acto knife, fashion a seat and back cushion, and secure to removable seat with industrial strength Velcro.
- Using a tape measure, determine the interior sections of the car that need cushioning to avoid any falls or bumps.
  » Then using an X-acto knife and the pool noodles, section the noodles down the center, and secure to all needed areas of the interior with industrial strength Velcro.
Warning

- If you smell smoke or notice any burning/melting, UNPLUG IMMEDIATELY!

- Pay close attention for any weird smells, actions from the motor etc.
The End

CONGRATULATIONS!
YOU’VE FINISHED YOUR FROZEN RIDE-ON CAR MODIFICATION!