Cherokey 4WD Mobile Platform (SKU:ROB0102)

Cherokey 4WD mobile platform

Contents

Introduction

The Cherokey 4WD is a versatile mobile robot that is compatible with popular microcontrollers such as the UNO, MEGA2560, Romeo, etc.

The Cherokey PCB is embedded with a L298P motor driver chip which allows it to drive two 6-12v DC motors with a maximum of 2A current. The integrated 2 way DC motor driver and XBee & APC220 socket allows you to start your project immediately without the need for an additional motor driver or wireless shield.

The expansion plate significantly increases the surface area of the Cherokey allowing you to easily connect a 9g micro servo or a standard sized servo in two different locations so that you can install a robotic arm or ultrasonic/IR sensors. The prototyping area makes it convenient to install sensors on the robot. Double sided solder pads in the middle of the top place can be populated with DIP or SMD components to extend the robot's functions.

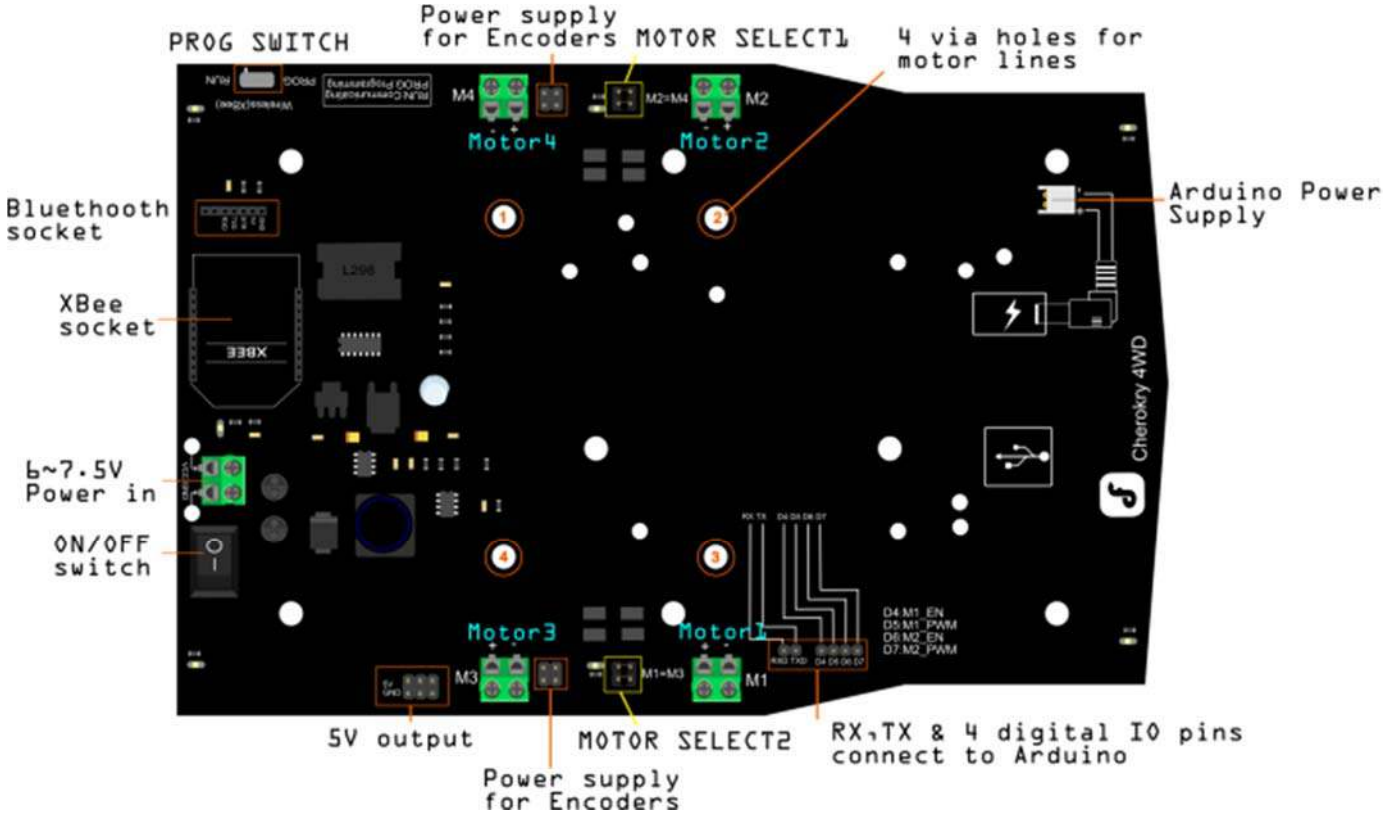
The high strength aluminium alloy chassis provides flexibilitiy in rapid movement particularly in outdoor enviornments, such as grass, gravel, sand or sloped surfaces.

The Cherokey 4WD mobile platform is also suitable for robot competitions and research-related projects.

Specification

- 2 channel DC motor driver
- Solder prototyping areas
- Servo sized holes
- Mounting holes compatible with Arduino UNO, MEGA, Romeo, etc.
- Incorporates a dual H-bridge for bi-directional motor control
- Easy to connect external modules such as XBee, DFRobot Bluetooth and APC220 wireless modules
- 7 white LEDS surrounding the board's edge for cool lighting effects
- Gearboxed motors for extra torque
- Motor Specification:
  Gear Ratio 1:120
  No-load speed(3V):100RPM
  No-load speed(6V):200RPM
  No-load current(3V):60mA
  No-load current(6V):71mA
  Stall current(3V):260mA
  Stall current(6V):470mA
  Torque (3V): 1.2Kgcm
  Torque (6V): 1.92Kgcm
  Size: 55mm x 48.3mm x 23mm
  Weight: 45g

Mainboard Pin Outs



In the above diagram the 4 digital I/O pins in the lower-right corner can connect to any digital pin of an Arduino or similar microcontroller. In order for it to work you must make sure the correct pins are assigned in your code.

Note: D5 and D6 are PWM pins

More Details:

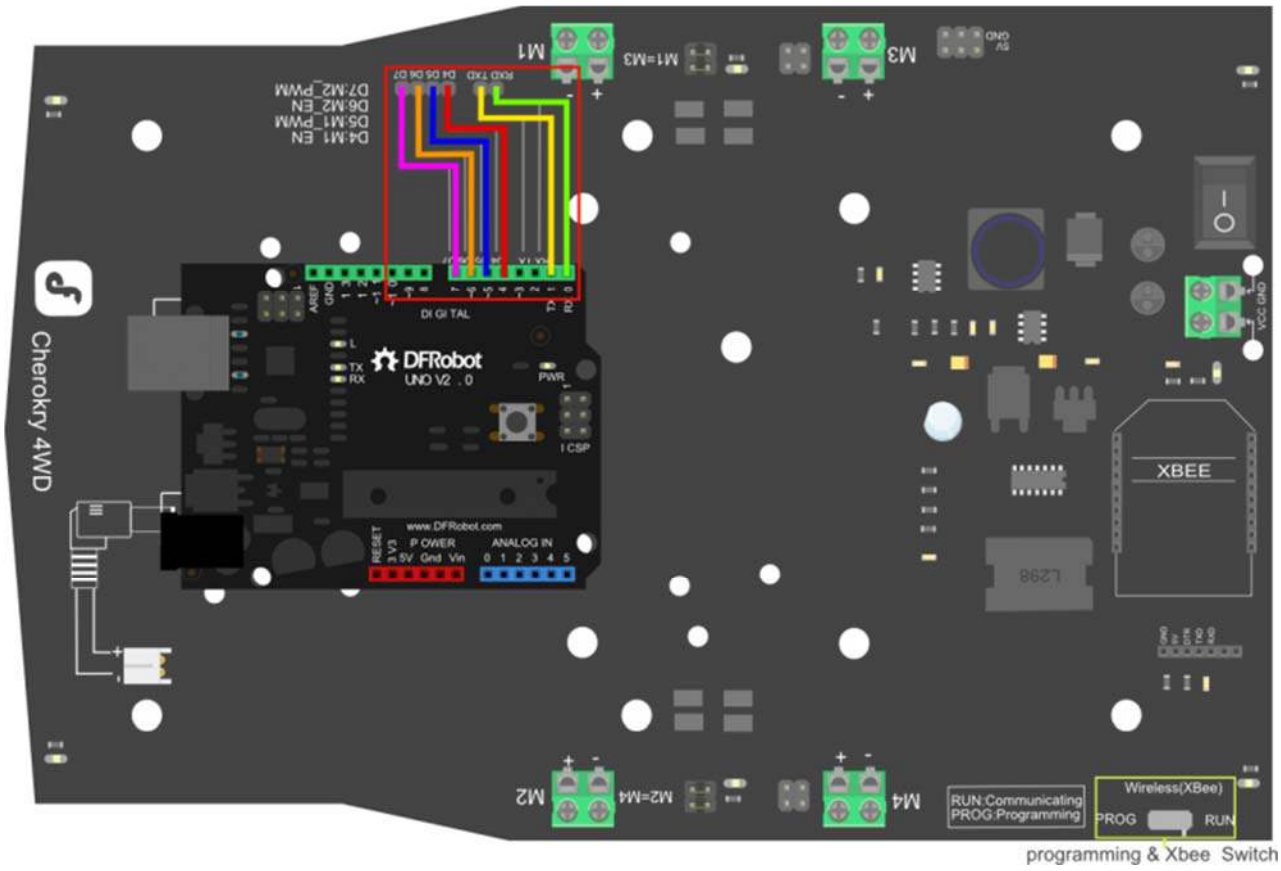
- PROG SWITCH:
  PROG:Set the switch in this position when uploading code if you plug in an XBee or Bluetooth module on to the Cherokey PCB
  RUN: Set the switch in this position to run the code after it has been uploaded
- MOTOR SELECT1: Short the pins with a jumpers to control Motor 2 and Motor 4 simultaneously
- MOTOR SELECT2: Short the pins with a jumper to control Motor 1 and Motor 3 simultaneously

Note: If you want to control the motors independently, remove the jumpers from these pins

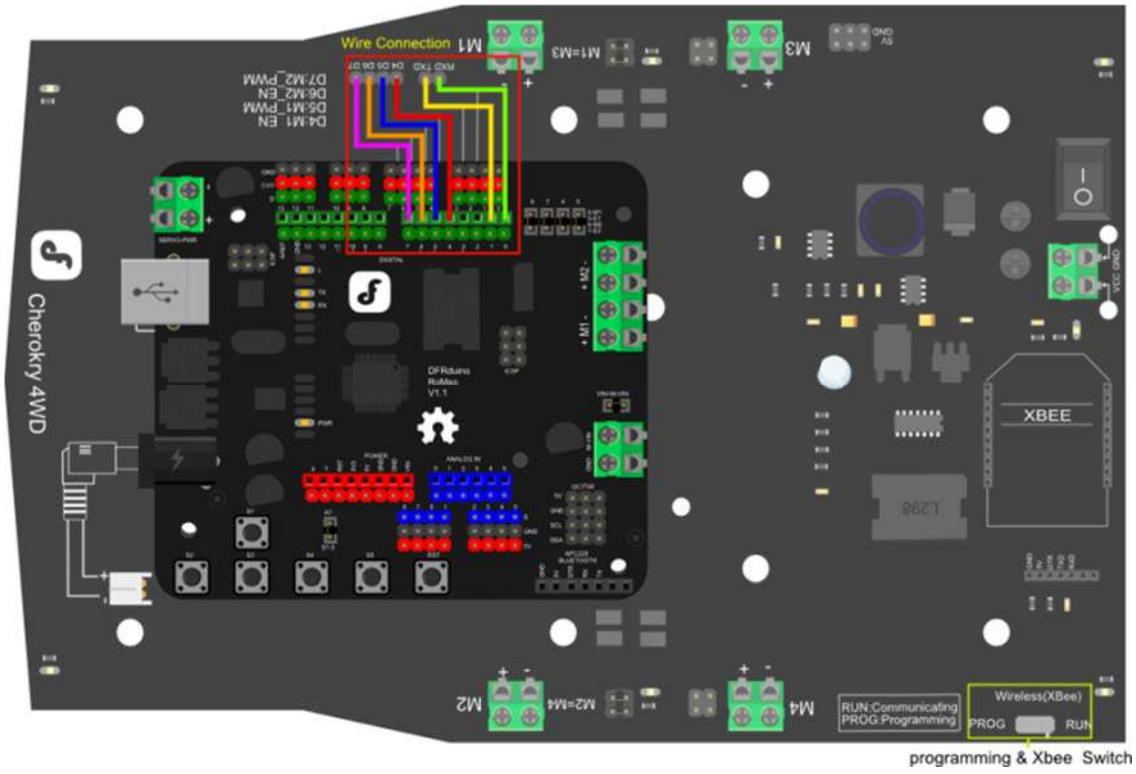
Correct and incorrect method of shorting the pins

Microcontroller Compatibility

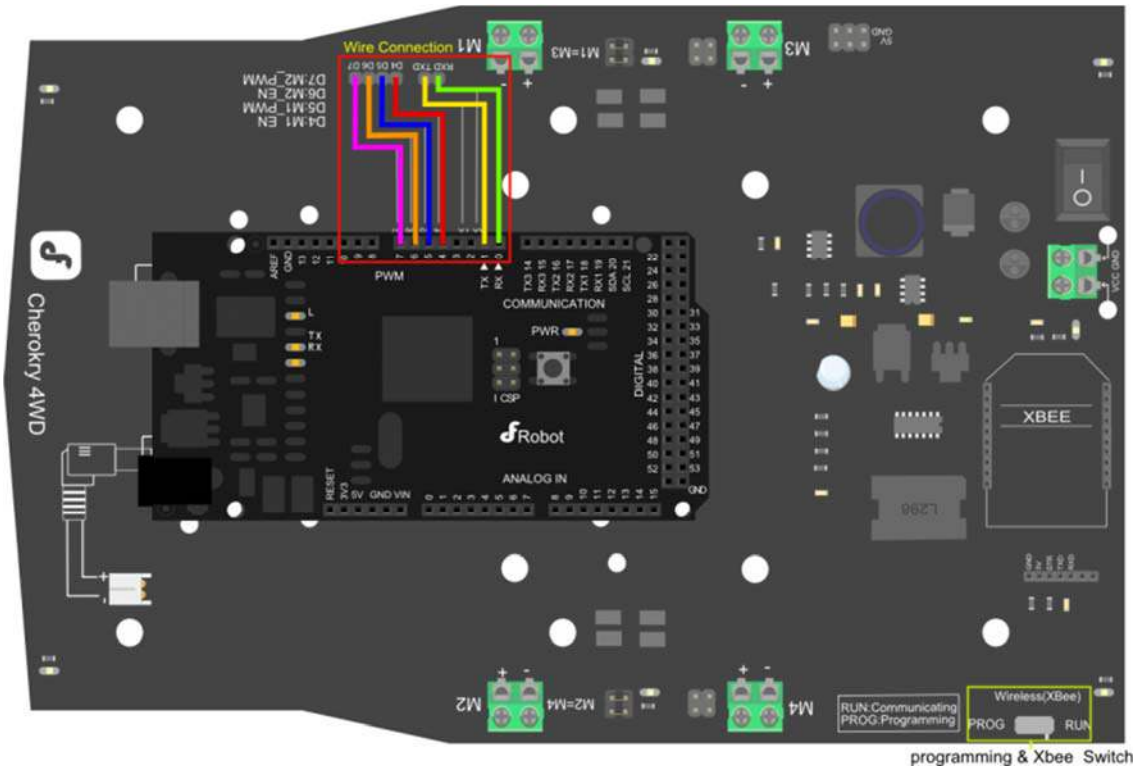
The Cherokey is compatible with most microcontrollers,such as Arduino UNO, Diecimila, Leonardo, Mega 1280,2560, ADK, Romeo etc.
If you use the RoMeo, you have complete control of 4 motors simultaneously.



Cherokey 4WD_DFRduino UNO



Cherokey 4WD_Romeo

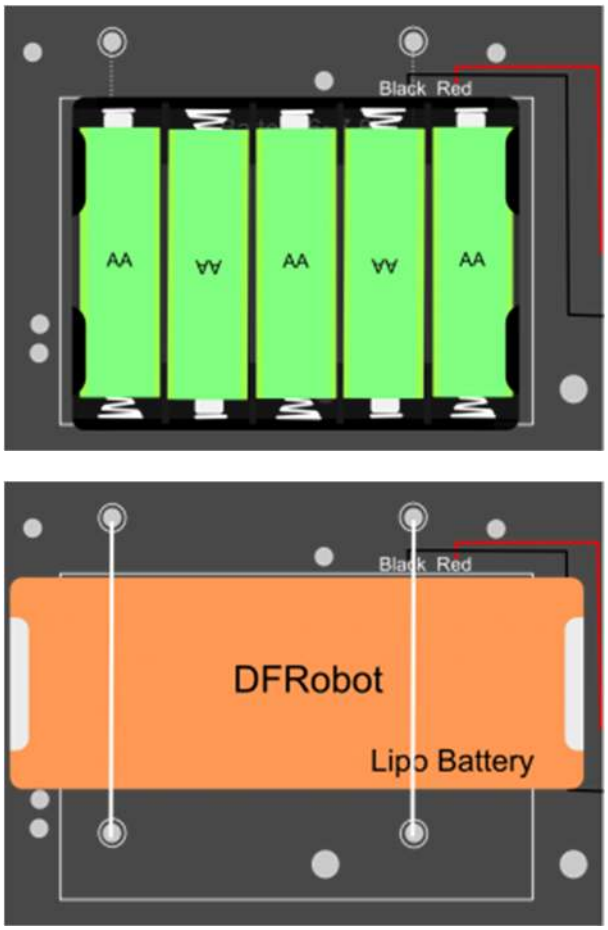


Cherokey 4WD_Mega

```
NOTE:

1 If you use Romeo or a Leonardo board where "RX0"&"TX0" are not exposed, you
need to change "Serial" to "Serial1".

2 Turn the "PROG/RUN" switch to the "PROG" position when you are going to upl
oad the sketch.

3 Turn the switch to "RUN" position when you are going to use Xbee,APC220 or
other modules.
```

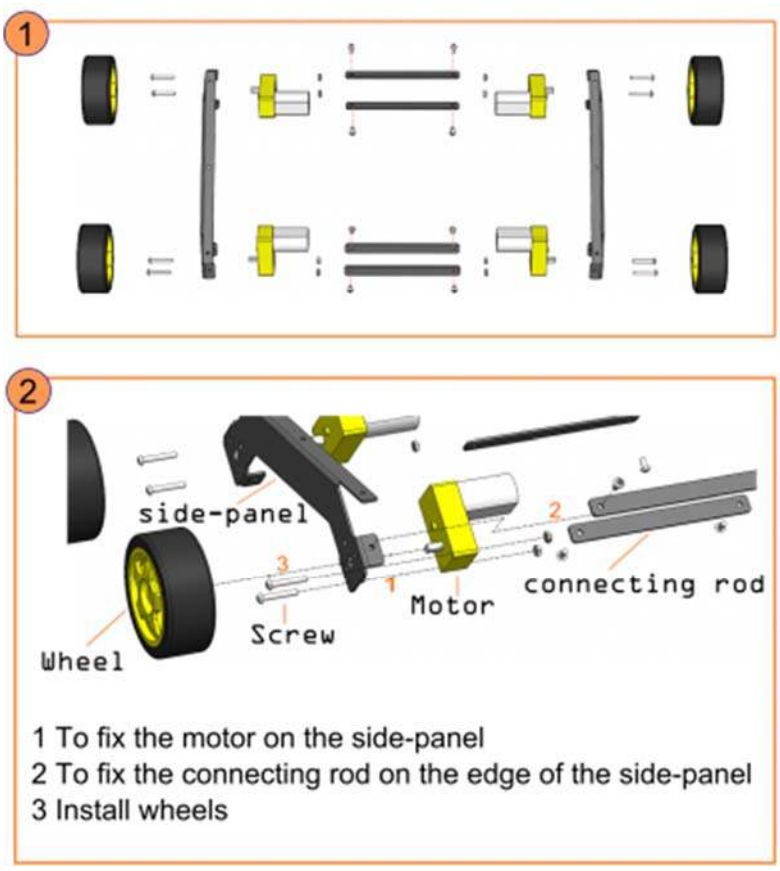
Power Supply

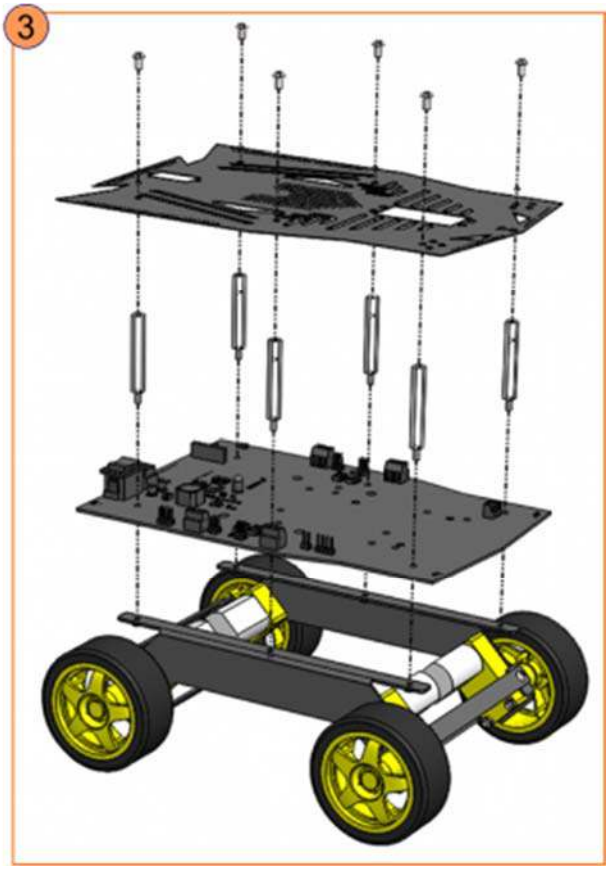
Install Battery on the back of the board.You can install 5xAA battery holder or lipo battery on the back of the board.



Installation Steps

Refer to the Instruction manual for detailed assembly steps





Sample Code

Simple Test Program

Use Arduino IDE to upload the following sketch to the microcontroller via the USB port. In Arduino IDE's board settings, you can use "Arduino UNO".
Under COM settings, select the microcontrollers COM port (the COM port will vary on your computer).
(Make sure your COM port is correctly assigned to your microcontroller in the IDE or the program will not upload!)

```
int speedPin_M1 = 5;      //M1 Speed Control
int speedPin_M2 = 6;      //M2 Speed Control
int directionPin_M1 = 4;     //M1 Direction Control
int directionPin_M2 = 7;     //M1 Direction Control
```

```
void setup(){
}

void loop(){
    carAdvance(100,100);
    delay(1000);
    carBack(100,100);
    delay(1000);
    carTurnLeft(250,250);
    delay(1000);
    carTurnRight(250,250);
    delay(1000);
}

void carStop(){                 //  Motor Stop
  digitalWrite(speedPin_M2,0);
  digitalWrite(directionPin_M1,LOW);
  digitalWrite(speedPin_M1,0);
  digitalWrite(directionPin_M2,LOW);
}

void carBack(int leftSpeed,int rightSpeed){         //Move backward
  analogWrite (speedPin_M2,leftSpeed);              //PWM Speed Control
  digitalWrite(directionPin_M1,HIGH);
  analogWrite (speedPin_M1,rightSpeed);
  digitalWrite(directionPin_M2,HIGH);
}

void carAdvance(int leftSpeed,int rightSpeed){       //Move forward
  analogWrite (speedPin_M2,leftSpeed);
  digitalWrite(directionPin_M1,LOW);
  analogWrite (speedPin_M1,rightSpeed);
  digitalWrite(directionPin_M2,LOW);
```

```
}


void carTurnLeft(int leftSpeed,int rightSpeed){      //Turn Left
  analogWrite (speedPin_M2,leftSpeed);
  digitalWrite(directionPin_M1,LOW);
  analogWrite (speedPin_M1,rightSpeed);
  digitalWrite(directionPin_M2,HIGH);
}
void carTurnRight(int leftSpeed,int rightSpeed){      //Turn Right
  analogWrite (speedPin_M2,leftSpeed);
  digitalWrite(directionPin_M1,HIGH);
  analogWrite (speedPin_M1,rightSpeed);
  digitalWrite(directionPin_M2,LOW);
}
```

First Test

After code has been uploaded to the microcontroller, unplug the USB cable from the board.
Place the Cherokey on a flat surface and at ground level for safety.
Turn the Cherokey on using the switch at the rear.

It should go backwards, forwards, turn 90 degrees to the left and turn 90 degrees to the right.

TROUBLESHOOTING TIPS:

Batteries must be connected to make the motors move! If the Cherokey is only plugged in with USB power through the microcontroller, the motors will be under powered and **will not** work!
If batteries are installed but the motors are not moving, make sure the switch at the rear of the Cherokey PCB is turned on
The Cherokey's direction may vary depending on the wiring of the motors. If you think the directions are wrong, try switching the positive and negative wires
If a problem persists, try editing the code to change the motor direction - covered in the **Changing the Motor Direction in Code** section below

Control Test Program

Now we can try another program that will give us keyboard control over the Cherokey.
The advantage of this program is that we will be given feedback to predefined directions. By observing the robot's motion, you can debug the motor directions so that each motor is spinning the correct way.
Upload the following code as before:

```
/*
 # Edited by:  Matt
 # Date:       2015.09.06
 # Version:    1.1
 # Product:    Cherokey 4WD Mobile Platform
 # SKU:        ROB0102/ROB0117

 # Description:
 # Drive 2 motors with this Cherokey 4WD Mobile Platform
 # Connect D4,D5,D6,D7,GND to UNO digital 4,5,6,7,GND

*/
//Motor Definitions
int E1 = 5;     //M1 Speed Control
int E2 = 6;     //M2 Speed Control
int M1 = 4;     //M1 Direction Control
int M2 = 7;     //M2 Direction Control

//DIRECTIONS

//STOP
void stop(void)
{
  digitalWrite(E1, 0);
  digitalWrite(M1, LOW);
  digitalWrite(E2, 0);
  digitalWrite(M2, LOW);
}

//ADVANCE
void advance(char a, char b)
{
  analogWrite (E1, a);
  digitalWrite(M1, HIGH);
```

```
  analogWrite (E2, b);
  digitalWrite(M2, HIGH);
}

//MOVE BACKWARDS void back_off (char a, char b)
{
  analogWrite (E1, a);
  digitalWrite(M1, LOW);
  analogWrite (E2, b);
  digitalWrite(M2, LOW);
}


//TURN LEFT
void turn_L (char a, char b)
{
  analogWrite (E1, a);
  digitalWrite(M1, LOW);
  analogWrite (E2, b);
  digitalWrite(M2, HIGH);
}

//TURN RIGHT
void turn_R (char a, char b)
{
  analogWrite (E1, a);
  digitalWrite(M1, HIGH);
  analogWrite (E2, b);
  digitalWrite(M2, LOW);
}

void setup(void) {
  int i;
  for (i = 4; i <= 7; i++)
```

```
    pinMode(i, OUTPUT);
  Serial.begin(9600);      //Set Baud Rate
  Serial.println("hello. w = forward, d = turn right, a = turn left, s = back
ward, x = stop, z = hello world"); //Display instructions in the serial monit
or
  digitalWrite(E1, LOW);
  digitalWrite(E2, LOW);
}


void loop(void) {
  if (Serial.available()) {
    char val = Serial.read();
    if (val != -1)
    {
      switch (val)
      {
        case 'w'://Move Forward
          Serial.println("going forward");
          advance (255, 255);  //move forward at max speed
          delay (1000);
          stop();
          break;
        case 's'://Move Backward
          Serial.println("going backward");
          back_off (255, 255);  //move backwards at max speed
          delay (1000);
          stop();
          break;
        case 'a'://Turn Left
          Serial.println("turning left");
          turn_L (255, 255);
          delay (1000);
          stop();
          break;
```

```
        case 'd'://Turn Right
          Serial.println("turning right");
          turn_R (255, 255);
          delay (1000);
          stop();
          break;
        case 'z':
          Serial.println("hello world!");
          break;
        case 'x':
          Serial.println("stopping");
          stop();
          break;
      }
    }
    else stop();
  }
}
```

Once the code has uploaded, **keep the USB cable plugged in**. Make sure the Cherokey's switch is ON and that a power supply is connected - e.g.: a lipo battery.

Open the Arduno IDE serial monitor. Set the bottom panels to "No line ending" and the baud rate to 9600. This is important as the microcontroller needs to communicate with your computer for this program to work properly.

If it is working correctly, the following line should appear when the serial monitor is opened:

```
hello. w = forward, d = turn right, a = turn left, s = backward, x = stop, z
= hello world
```

These are basic instructions for this program. Using the W, D, A, S, and Z keys on your keyboard, try moving the Cherokey.

When you press "W", the Cherokey should move forward for 1 second, and then stop
When you press "D", the Cherokey should turn to the right 90 degrees and then stop
When you press "A", the Cherokey should turn to the left 90 degrees and then stop

When you press "S", the Cherokey should move backwards for 1 second, and then stop
When you press "Z", the Arduino IDE serial monitor should print: "hello world!"

If motors are spinning incorrectly, there are two methods to change this:

- Changing the polarity of the motors by changing the positive and negative wiring. (which works, but isn't an elegant solution)
- Changing lines in the code, covered in the section below

Changing the Motor Direction in Code

Let's examine some of the global variables in the program:

```
int M1 = 4;     //M1 Direction Control
int M2 = 7;     //M2 Direction Control
```

Digital pins 4 and 7 have been assigned as the motor direction control pins. By setting each either HIGH or LOW (i.e. on or off), we can control which way the motor will turn.

Let's examine another section of code:

```
//TURN LEFT
void turn_L (char a, char b)
{
  analogWrite (E1, a);
  digitalWrite(M1, LOW);
  analogWrite (E2, b);
  digitalWrite(M2, HIGH);
}
```

This is a function that tells the Cherokey to turn left. M1 and M2 are set as LOW and HIGH respectively. This means that the left-side wheels will turn backwards and the right-side wheels turn forwards.

Conversely, turning right has the motor pins set like so:

```
//TURN RIGHT
void turn_R (char a, char b)
{
  analogWrite (E1, a);
  digitalWrite(M1, HIGH);
  analogWrite (E2, b);
  digitalWrite(M2, LOW);
}
```

Therefore, if you find that your Cherokey's wheels are going in a direction you don't intend them to, try changing the motor pins signal. If they are going the wrong way and the direction pin is set to HIGH, try changing it to LOW, and vice versa. You can use the keyboard control program to debug and verify these settings.