Due: 03/8/2023

Rover Mission #1 – Rolling Chassis

a. Drive in a ~10ft square pattern, stop at original location and place marker. The marker should be within 3ft of the rover's original location and the mission should be completed within 60 seconds.
b. Drive straight for 15-30ft and place physical marker. The rover will read the specific intended distance from an RF transmitter beacon at the starting location. The marker should be within +/- 10% of the specified drive distance and the task should be completed in less than 30 seconds.
    - The distance to drive will be transmitted as a 12-character message. The distance will be two digits preceded by an 'X'. The remainder of the message will be padded with 'Z' characters. For example, the message for the rover to travel 15 feet would be "X15ZZZZZZZZZ".
    - The distance to be traveled will be continuously transmitted from the RF beacon at 5 second intervals.

*Requires two consecutive successful runs

Results

PASSED

Rover V.1 was used to attempt this mission which included a DC motor, Arduino uno, L298P shield R3 motor driver module, Futaba S3003 standard servo, 9.6V 2000mAH NiMH battery pack, MG90S micro servo and 433 MHz RF receiver.

Arduino sketch

Mission 1A

```
//Mission 1.A arduino code
//This sketch allows the rover to drive in an approximate 10' square and drop a payload
//Purpose to calibrate servo motor for steering and payload drop
//as well as drive power ratio for distance/time


#include <Servo.h>  //include Servo library

//Servo
Servo Steer;                  //create Steer servo
Servo Dump;                   //create Dump servo for payload
#define SERVO_PIN_STEER A3    //attach steer servo to pin A3
#define FRONT 100             //Initial position for Steer servo
#define SERVO_PIN_DUMP A2     //attach Dump servo to A2
#define DUMP 140              //initial position for dump servo
```

```
//create variables for servos
int RIGHT=FRONT+20;
int LEFT=FRONT-20;
int PAYLOAD=DUMP-125;

//DC motor settings
int POWER=100;                //Power rating
int directionPin = 12;        //Diriction pin on 12
int pwmPin = 3;               //power pin on 3
int brakePin = 9;             //brake pin on 9

void setup() {

//pin output layouts for DC motor
pinMode(directionPin, OUTPUT);
pinMode(pwmPin, OUTPUT);
pinMode(brakePin, OUTPUT);

//Attach servos
Steer.attach(SERVO_PIN_STEER);
Dump.attach(SERVO_PIN_DUMP);

Dump.write(DUMP);                 //insures payload is in origional position
digitalWrite(directionPin, HIGH); //DC motor HIGH->forward
digitalWrite(brakePin, LOW);      //release breaks
analogWrite(pwmPin, POWER);       //turn on DC motor
delay(3500);                      //keep DC motor on for time,
                                  // this extra time takes acceleration into consideration

Steer.write(RIGHT);               //turn right
delay(1700);                      //keep servo in position for time
Steer.write(FRONT);               //return to FRONT/forward posiiton
analogWrite(pwmPin,POWER);
delay(1000);

 for (int i = 0; i <=1; i++) {   //loop for final 2 turns at full speed

    Steer.write(RIGHT);
    delay(1500);
    Steer.write(FRONT);
     analogWrite(pwmPin,POWER);
    delay(1000);
 }

digitalWrite(brakePin, HIGH);  //activate breaks
analogWrite(pwmPin, 0);       //set work duty for the motor to 0 (off)
delay(2000);
Dump.write(PAYLOAD);          //turns servo for payload to drop it
delay(2000);
Dump.write(DUMP);             //return payload mechanism to origional spot
}

void loop() {

}
```

Mission 1B

```
//This sketch allows the rover to recieve a number from a transmitter and convert it
//into a drive distance for the rover to go

#include <RH_ASK.h> //gets RadioHead from library
#include <ServoTimer2.h> //gets ServoTimer2 library
#include <SPI.h> // Not actualy used but needed to compile

//RH_ASK driver;   //for debug: creates RadioHead driver
RH_ASK driver(2000, 5, 12, 0); //changes default pin for reciver from RH,pin 11->5

//Payload servo
ServoTimer2 Dump;  //creates servo for payload
#define SERVO_PIN_DUMP A2    //assigns pin A2 for servo
#define DUMP 2000            //initial position for servo
int PAYLOAD=DUMP-1300;       //servo position when dropping payload

//pin layout for DC motor
int directionPin = 12; //pin 12 for DC motor direction
int pwmPin = 3;        //pin 3 for DC power
int brakePin = 9;      //pin 9 for DC brake
int drivetime;         //int for driving time

//RF reciver numbers
String start;   //sets 'start' as a string
int out;        //sets 'output' as an integer
String number;  //sets 'number' as a string

void setup() {


 //modes for the DC motor
pinMode(directionPin, OUTPUT);    //sets direction output(Forward/Back)
pinMode(pwmPin, OUTPUT);          //power settings (0-255) of input voltage
pinMode(brakePin, OUTPUT);        //Brake pin output


  //initialize set for RF Reciever
Serial.begin(9600);               //baud rate for serial monitor
if (!driver.init())               //if driver fails to load...
Serial.println("init failed");    //print failed status

}

void loop() {

  //ensures only gathers 12 character transmition
  uint8_t buf[12];
  uint8_t buflen = sizeof(buf);

 //RF Reciever conversions
   if (driver.recv(buf, &buflen)) // Non-blocking
    {
```

```
     start = String((char*)buf);     //turns data from transmitter as a string
     number = start.substring(1,3 ); //gets 2nd and 3rd digits from start string
     out = number.toInt();           //turns string number from 'number' string to integer
     drivetime = out*340;            //conversion from transmitter to drive time
    Serial.println(out);             //Debug: prints number from 'out'
    Serial.println(drivetime);       //Debug: Prints number conversion for driving time
    delay(100);
   }

if(out >14 && out<31){

//DC Motor instructions
  digitalWrite(directionPin, HIGH);    //Set direction forward
  digitalWrite(brakePin, LOW);         //Take off brakes
  analogWrite(pwmPin, 100);            //set power to DC motor
  delay(drivetime);                    //driving time
  digitalWrite(brakePin, HIGH);        //activate breaks
  analogWrite(pwmPin, 0);              //set work duty for the motor to 0 (off)

delay(2000);

//payload drop instructions
Dump.attach(SERVO_PIN_DUMP);          //Attach the servo
Dump.write(PAYLOAD);                  //Dump the payload
delay(2000);
Dump.write(DUMP);                     //return to origional position

while(1);                             //Run loop once
}
   }
```

Following modifications

Moving forward to Mission #1 we plan to add in a swivel sweep sensor system in the front center of the rover that covers 120°. We are also going to replace the current DC motor with a new brushless DC motor to improve the power and see if its capable of driving through grass.