Date: 12/07/10

Student Name: Mary Richeson

TA's :Mike Pridgen
Tim Martin
Ryan Stevens
Devin Hughes
Thomas Vermeer

Instructors: Dr. A. Antonio Arroyo
Dr. Eric M. Schwartz

University of Florida
Department of Electrical and Computer Engineering

EEL 4665/5666
Intelligent Machines Design Laboratory

Written Report

Rico the Recycler

Table of Contents

ABSTRACT

Rico the Recycler will locate and sort recyclables found on the ground. Rico will be able to determine the difference between aluminum cans and glass bottles. The robot will find the recyclables left on the ground. Once it locates a recyclable it will drive toward it until it rolls on to a main arm platform. The arm will then rise to a ninety degree angle where Rico will decide whether the object is a can or a bottle. Rico will then place the bottle or can in the appropriate recycling container.

EXECUTIVE SUMMARY

Rico will locate recyclables by moving in a lawn mower type fashion up and down each row of a room using infrared sensors to locate object low to the ground. Once it locates the recyclable it will drive toward it and then directly at it. The bottles will roll on to a main arm platform that is flush with the ground. An infrared sensor will detect it on the metal platform and the arm will then rise to a ninety degree angle where Rico will decide whether its heavy or light based on a weight sensor on the platform. If Rico determines it is heavy and a glass bottle it will lift a storage container lid and then continue to lift the arm platform so the bottle will roll into the first storage container. Otherwise it will not lift the lid to the first bottle storage container and as the arm is lifted the aluminum can will roll over the first container and into the second aluminum container that does not have a lid. It will continue to do this until it reaches the maximum of 4 recyclables.

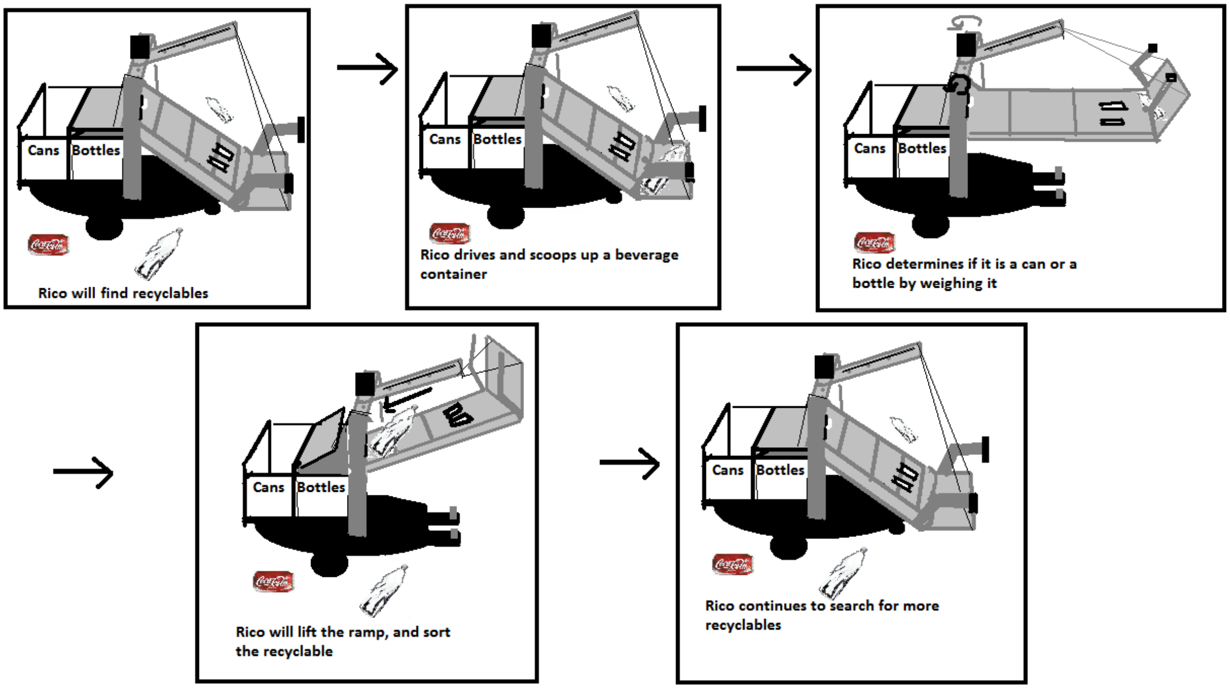
INTRODUCTION

Many times after sporting events, such as swimming tons of trash is left on the ground including items that can be recycled. Often many of these recyclables are just thrown into the trash and taken to a landfill. A robot that could help pick up and sort recyclables would be ideal in this situation.

Rico is just the robot to help with recycling. Rico works indoors to cleaning up arena hallways or swimming patios littered with recyclables. Rico will have an integrated system to help it navigate pathways in a lawn mowing type fashing, by driving up a pathway one row at a time. Rico's mobile platform is customized to picking up, sorting and storing aluminum and glass recyclables. The different actuation systems will allow the robot to easily drive the platform and lift recyclables. Four infrared sensors, two for obstacle avoidance, and two for recyclable recognition will be in place. A pressure sensor will also be implemented to help with sorting.

INTEGRATED SYSTEM

Rico will survey an area by moving in a lawn mower type fashion, moving up one row at a time. It will use infrared sensors for obstacle avoidance. Once it locates a recyclable it will drive toward it until it rolls on to a main arm platform. On the platform a weight sensor will determine whether the object is an aluminum can or a glass bottle. Depending which object is detected a specific flap will be lifted and the recyclable will be placed into the appropriate container. Rico will stop looking for cans once a maximum number of cans or bottles have been found. In this manner Rico will successfully meet the objectives by locating, sorting and storing the recyclables.


Cans
Bottles
Rico will find recyclables
Rico drives and scoops up a beverage container
Rico determines if it is a can or a bottle by weighing it
Rico will lift the ramp, and sort the recyclable
Rico continues to search for more recyclables

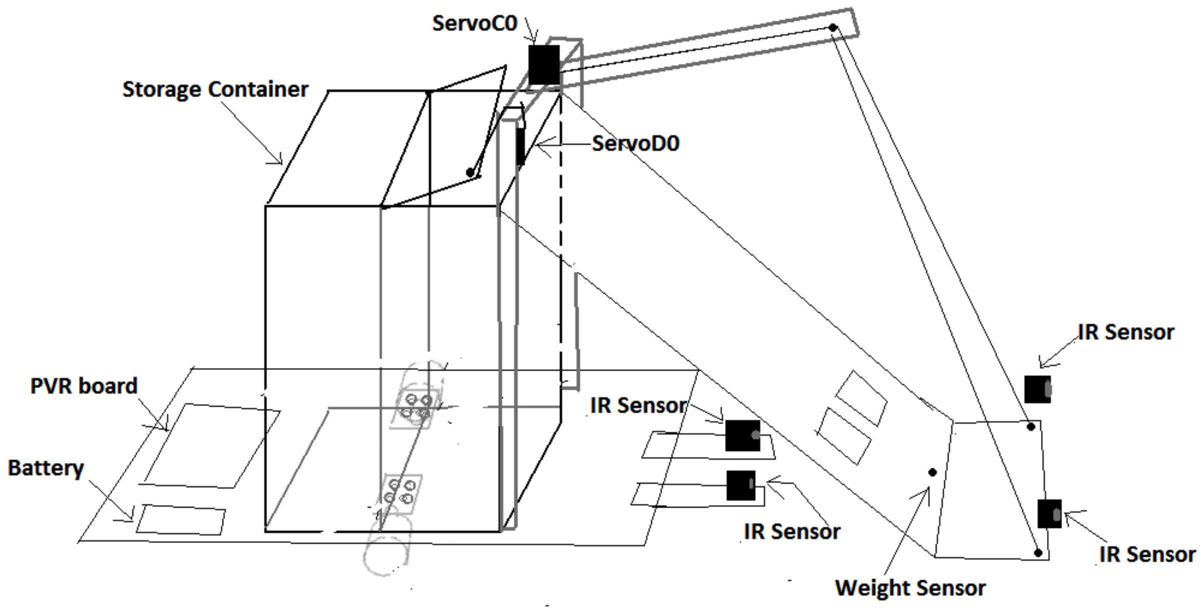
MOBILE PLATFORM

The platform is a two feet in length by one foot in width. It is made of wood a quarter inch thick. There are three wheels attached in a triangular position. The two side wheels will be motorized with the third front wheel just being a castor. On the middle of the platform rests the storage container. The storage area is divided with one side for cans and one side for bottles. An arm ramp is mounted with one side attached to the storage containers and the other side resting on the ground. The plastic arm ramp is 12.5 inches wide, 24 inches long and 1/12 of an inch thick. The micro processor board and batteries will be located at the back of the platform.

Lessons

While trying to find a light weight platform plastic was used for the arm ramp. While this was a good intention, it had unintended negative consequences as the ramp was unstable and had half and inch of unanticipated side to side movement. To further complicate this the storage container it was mounted to was also plastic and contributed to more movement.

Platform Specifications		
Main Platform	Wood	2' x 1' x ¼"
Wheels	Black foam	2.5" x 2.5"
Storage Container	Plastic	8" x 12" x 6"
Arm Ramp	Plastic	12.5" x 24" x 1/12"



ACTUATION

The types of actuation are the wheels, arm platform, and storage container lid flap.

Motor and Wheels

There is a motor for two foam tires. The foam wheels are 2.5 inches by 1.5 inches. This will allow Rico to a medium speed and carry the weight of the robot adequately. Each wheel motor has 3.6 kg-cm torque to drive around the arena and pick up recyclables. The motors are 12 V with 200 RPM. The torque will be well above the calculated into the max torque of the motor which includes the weight of the platform, wheels, battery, motors, servos, platform arm ramp, and filled storage containers. The wheels are mounted with an aluminum mount. A duel motor driver with a built in opto-isolator will allow the wheels to interact and easily allow it to move forward, reverse, and turn either left or right. The opto-isolator will prevent the motors from accidentally dropping too much current onto the ground line and destroying the microprocessor.

*I used Mark Milks code for the motor and wheels because we had the same wheel kit and dual motor driver

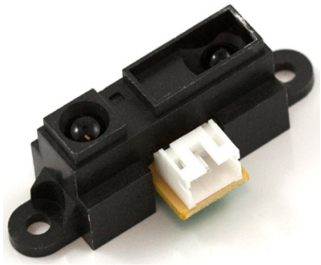
Arm Ramp Platform

The first servo will be connected to the arm ramp with a pulley system. A hacked servo will
will be attached to a stainless steel frame mounted over the storage containers. Fishing wire will be strung through an extended metal structure which exits and ties at each end of the metal platform that is flush with the ground. This servo moves in forward 360 ° motion for 12 seconds forward to lift a glass bottle and 10 seconds for a can. This 4.8 V servo has a stall torque of 42 oz/in and a max operating speed of 1.14 seconds per 360° of rotation.

Storage Container Lid

The storage container lid will operate on the second servo. This hacked servo can rotate 360°. It is also mounted on the metal frame located over the storage containers. It is attached to fishing wire that runs over the metal frame and down to one courner of the storage lid. To lift the lid the servo will move in a counter clockwise motion to wind up the string for 1500 ms, and then in reverse to close the lid again. This 4.8 V servo has a stall torque of 42 oz/in and a max operating speed of 0.29 seconds per 90 ° of rotation.

SENSORS



Infrared Proximity Sensor

An infrared sensor was used for obstacle avoidance so Rico stays in the boundaries of the 10 x 10' arena. The 3.1 V analog output infrared sensors can detect distances from 4 inches to 30 inches. Once Rico gets within 5 inches of a wall it will turn at a 180 degrees. In this driving pattern Rico will run into the different pieces of recyclables placed in the arena.

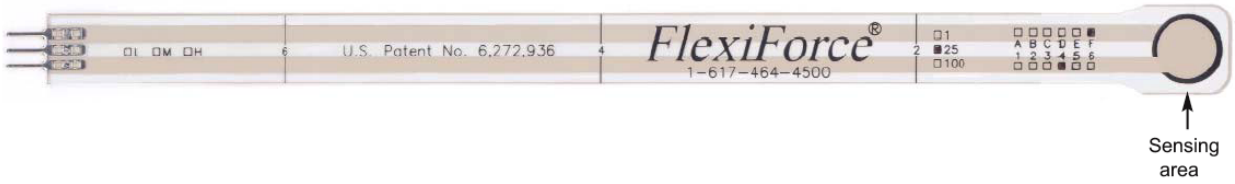
Infrared Proximity Sensor	
Current Range	0 - 3.1 V
Resistance Range	4 – 30 inches
Output range	2200-500

Distance (inches)	**IR 1 Output Value**	**IR 2 Output Value**	**IR 3 Output Value**	**IR 4 Output Value**
20	900	1960	2200	1940
15	890	1900	1900	1950
12	938	3000	2200	2100
10	938	3200	3000	3000
8	1700	3500	3100	3100
6	2179	3188	3500	3400
4	3500	3100	3100	3500
2	3100	3100	3500	3600

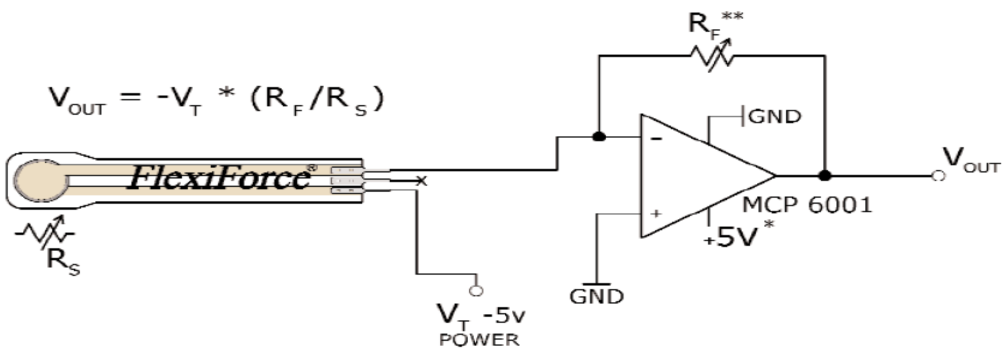
Reference:SEN-00242 Infrared Proximity Sensor - Sharp GP2Y0A21YK, SparkFun Electronics, 1-303-284-0979

Weight Sensor

A weight sensor will be used to determine the type of recyclable on the scale part of the platform. Once a recyclable is on the metal platform a servo will move the arm ramp to a horizontal position where sufficient weight of the recyclable will be on the weight sensor. The 8.5 inch weight sensor is a small flexi-force pressure sensor. Adding weight to the sensor reduces the resistance of the sensor which it translates into a weight. The resistance changes from infinity to 300k ohms. It is sensitive from 0 to 1 pound. When it detects an object weighing 0.031 pound it will determine it is an empty 12 oz aluminum can (below the 3000 output value threshold). If it detects an object weighing 0.41 pounds it will decide it is an empty 12 oz glass bottle (above the 3000 output value threshold). After determining the type of recyclable Rico will lift up the arm ramp further until the recyclable rolls into the correct container. An 100k ohm resistor was used in this circuitry. At a value of above 3000 the sensor will consistently detect a bottle and not detect a can



Flexiforce Pressure Sensor Piezoresistive force sensor- as you increase the pressure the lower the sensor's resistance	
Pressure range	0 -1 lbs
Resistance range	Infinite – 300k
Dimensions	8.5 x 0.55 x 0.008 inches
Sensing diameter	0.375"
Response Time	Less than 5 microseconds
Can	< 3000 (LCD output value)
Bottle	>3000 (LCD output value)



For the reference resistance

```
                    ServoC0(-100);
///descends arm all way down

                    delay_ms(20000);

                    ServoC0(0);

                    delay_ms(5000);


                    M0_ForwardSlow(70);

                    M1_ForwardSlow(70);

                    }




                    else{                    ///if light then is can

                    ServoC0(100);

                    delay_ms(13000);

                    ServoC0(0);

                    delay_ms(3000);
                    ServoC0(-100);

                    delay_ms(14000);


                    M0_ForwardSlow(70);

                    M1_ForwardSlow(70);

                    }


            M0_ForwardSlow(70);

            M1_ForwardSlow(70);


            }


    }


}


void M0_ReverseSlow(int a){


    sendData[0] = 170;

    sendData[1] = 9;

    sendData[2] = 8;

    sendData[3] = a;


    for(int i = 0; i<4;i++){
        do{


        }while(!USART_IsTXDataRegisterEmpty(&USART));

        USART_PutChar(&USART, sendData[i]);


    }



}


void M0_ReverseFast(int a){


    sendData[0] = 170;

    sendData[1] = 9;

    sendData[2] = 9;

    sendData[3] = a;


    for(int i = 0; i<4;i++){


        do{


        }while(!USART_IsTXDataRegisterEmpty(&USART));

        USART_PutChar(&USART, sendData[i]);


    }
```

```
}

void M0_ForwardSlow(int a){

        sendData[0] = 170;
        sendData[1] = 9;
        sendData[2] = 10;
        sendData[3] = a;

        for(int i = 0; i<4;i++){

                do{

                }while(!USART_IsTXDataRegisterEmpty(&USART));
                USART_PutChar(&USART, sendData[i]);

        }


}

void M0_ForwardFast(int a){

        sendData[0] = 170;
        sendData[1] = 9;
        sendData[2] = 11;
        sendData[3] = a;
        for(int i = 0; i<4;i++){

                do{

                }while(!USART_IsTXDataRegisterEmpty(&USART));
                USART_PutChar(&USART, sendData[i]);

        }


}

void M1_ForwardSlow(int a){

        sendData[0] = 170;
        sendData[1] = 9;
        sendData[2] = 12;
        sendData[3] = a;

        for(int i = 0; i<4;i++){

                do{

                }while(!USART_IsTXDataRegisterEmpty(&USART));
                USART_PutChar(&USART, sendData[i]);

        }
```

```
}

void M1_ForwardFast(int a){

        sendData[0] = 170;
        sendData[1] = 9;
        sendData[2] = 13;
        sendData[3] = a;

        for(int i = 0; i<4;i++){

                do{

                }while(!USART_IsTXDataRegisterEmpty(&USART));
                USART_PutChar(&USART, sendData[i]);

        }

}

void M1_ReverseSlow(int a){

        sendData[0] = 170;
        sendData[1] = 9;
        sendData[2] = 14;
        sendData[3] = a;
        for(int i = 0; i<4;i++){

                do{

                }while(!USART_IsTXDataRegisterEmpty(&USART));
                USART_PutChar(&USART, sendData[i]);

        }

}

void M1_ReverseFast(int a){

        sendData[0] = 170;
        sendData[1] = 9;
        sendData[2] = 15;
        sendData[3] = a;

        for(int i = 0; i<4;i++){

                do{

                }while(!USART_IsTXDataRegisterEmpty(&USART));
                USART_PutChar(&USART, sendData[i]);

        }
```

```
}
```