Wall-Rider, The Wall Climbing Robot
Sumaiya Islam, IshraqulHuq, NibrasSajjadDipro, Mahabub Ahmed

Department of EEE, Ahsanullah University of Science and Technology, Dhaka, Bangladesh.

Abstract- This paper proposes a wall climbing robot which is able to climb on vertical surfaces using suction cup for attachment. The suction is created by a vacuum pump. It has onboard power and uses an RF (radio frequency) controlled remote to drive itself. Detail of the project is discussed below.

Introduction
Mobile robots with the ability to climb and navigate on surfaces of any orientation without leaving residue or damaging the surface have many potential applications. In general, climbing robots use one of three types of attachment mechanisms: vacuum suction, magnetic attraction, or gripping with claws or grasping mechanism. Each of these mechanisms has merits and demerits. For instance, magnetic adhesion can be very strong with the possibility of safe power failure mitigation, but is only applicable for ferromagnetic surfaces. Suction adhesion relies on a complete seal with the surface making cracked or non-smooth surfaces problematic, and power efficiency limits their untethered climbing duration. Recently, robots using microclaws have demonstrated climbing capabilities on surfaces such as brick and stone, but clawed and grasping robots cannot climb smooth surfaces such as glass or painted structures.

Equipment
Remote
1. Arduino UNO
2. Tx of the 315MHz module Rx Tx pair
3. Breadboard

Receiver
1. Arduino UNO
2. Rx of the Tx/Rx pair
3. Breadboard
4. 12V battery for motor power
5. L293D motor driver chip (2pieces)
6. 9V battery for Arduino power
7. Jumper wires

Chasis
1. DC motors (4)
2. Wheels (4)
3. Suction cap
4. Vacuum pump
5. Robot chasis

Equipment pictures:
Arduino
Arduino UNO

Arduino is a single-board microcontroller, intended to make the application of interactive objects or environments more accessible. The hardware consists of an open-source hardware board designed around an 8-bit Atmel AVR microcontroller, or a 32-bit Atmel ARM. Current models feature a USB interface, 6 analog input pins, as well as 14 digital I/O pins which allows the user to attach various extension boards.

Introduced in 2005, the Arduino platform was designed to provide an inexpensive and easy way for hobbyists, students and professionals to create devices that interact with their environment using sensors and actuators. Common examples for beginner hobbyists include simple robots, thermostats and motion detectors. It comes with a simple integrated development environment (IDE) that runs on regular personal computers and allows users to write programs for Arduino using C or C++.

Summary

<table>
<thead>
<tr>
<th>Microcontroller</th>
<th>ATmega328</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Voltage</td>
<td>5V</td>
</tr>
<tr>
<td>Input Voltage (recommended)</td>
<td>7-12V</td>
</tr>
<tr>
<td>Input Voltage (limits)</td>
<td>6-20V</td>
</tr>
<tr>
<td>Digital I/O Pins</td>
<td>14 (of which 6 provide PWM output)</td>
</tr>
<tr>
<td>Analog Input Pins</td>
<td>6</td>
</tr>
<tr>
<td>DC Current per I/O Pin</td>
<td>40 mA</td>
</tr>
<tr>
<td>DC Current for 3.3V Pin</td>
<td>50 mA</td>
</tr>
<tr>
<td>Flash Memory</td>
<td>32 KB (ATmega328) of which 0.5 KB used by bootloader</td>
</tr>
<tr>
<td>SRAM</td>
<td>2 KB (ATmega328)</td>
</tr>
<tr>
<td>EEPROM</td>
<td>1 KB (ATmega328)</td>
</tr>
<tr>
<td>Clock Speed</td>
<td>16 MHz</td>
</tr>
</tbody>
</table>

RF Module

The RF module, as the name suggests, operates at Radio Frequency. The corresponding frequency range varies between 30 kHz & 300 GHz. In this RF system, the digital data is represented as variations in the amplitude of carrier wave. This kind of modulation is known as Amplitude Shift Keying (ASK).

Transmission through RF is better than IR (infrared) because of many reasons. Firstly, signals through RF can travel through larger distances making it suitable for long range applications. Also, while IR mostly operates in line-of-sight mode, RF signals can travel even when there is an obstruction between transmitter & receiver. Next, RF transmission is more strong and reliable than IR transmission. RF communication uses a specific frequency unlike IR signals which are affected by other IR emitting sources.

This RF module comprises of an RF Transmitter and an RF Receiver. The transmitter/receiver (Tx/Rx) pair operates at a frequency of 434 MHz. An RF transmitter receives serial data and transmits it wirelessly through RF through its antenna connected at pin4. The transmission occurs at the rate of 1Kbps - 10Kbps. The transmitted data is received by an RF receiver operating at the same frequency as that of the transmitter.

The RF module is often used along with a pair of encoder/decoder. The encoder is used for encoding parallel data for transmission feed
while reception is decoded by a decoder. HT12E-HT12D, HT640-HT648, etc. are some commonly used encoder/decoder pair ICs.

**L293D Motor Driver**

L293D is a typical Motor driver or Motor Driver IC which allows DC motor to drive on either direction. L293D is a 16-pin IC which can control a set of two DC motors simultaneously in any direction.

It works on the concept of H-bridge. H-bridge is a circuit which allows the voltage to be flown in either direction. As you know voltage need to change its direction for being able to rotate the motor in clockwise or anticlockwise direction, Hence H-bridge IC are ideal for driving a DC motor.

In a single L293D chip there two h-Bridge circuit inside the IC which can rotate two dc motor independently. Due its size it is very much used in robotic application for controlling DC motors. Given below is the pin diagram of a L293D motor controller.

There are two Enable pins on L293d. Pin 1 and pin 9, for being able to drive the motor, the pin 1 and 9 need to be high. For driving the motor with left H-bridge you need to enable pin 1 to high. And for right H-Bridge you need to make the pin 9 to high. If anyone of the either pin1 or pin9 goes low then the motor in the corresponding section will suspend working. It’s like a switch.

There are 4 input pins for this L293d, pin 2,7 on the left and pin 15,10 on the right as shown on the pin diagram. Left input pins will regulate the rotation of motor connected across left side and right input for motor on the right hand side. The motors are rotated on the basis of the inputs provided across the input pins as LOGIC 0 or LOGIC 1.

In simple you need to provide Logic 0 or 1 across the input pins for rotating the motor.

Lets consider a Motor connected on left side output pins (pin 3,6). For rotating the motor in clockwise direction the input pins has to be provided with Logic 1 and Logic 0.

- Pin 2 = Logic 1 and Pin 7 = Logic 0 | Clockwise Direction
- Pin 2 = Logic 0 and Pin 7 = Logic 1 | Anticlockwise Direction
- Pin 2 = Logic 0 and Pin 7 = Logic 0 | Idle [No rotation] [Hi-Impedance state]
- Pin 2 = Logic 1 and Pin 7 = Logic 1 | Idle [No rotation]

In a very similar way the motor can also operated across input pin 15,10 for motor on the right hand side.
VCC is the voltage that it needs for its own internal operation 5v; L293D will not use this voltage for driving the motor. For driving the motors it has a separate provision to provide motor supply VSS (V supply). L293d will use this to drive the motor. It means if you want to operate a motor at 9V then you need to provide a Supply of 9V across VSS Motor supply.

The maximum voltage for VSS motor supply is 36V. It can supply a max current of 600mA per channel. Since it can drive motors Up to 36v hence you can drive pretty big motors with this L293d.

VCC pin 16 is the voltage for its own internal Operation. The maximum voltage ranges from 5v and upto 36v.

**Block Diagram of Project Setup**
Driver and Motor

Similar for motor 3 and motor 4.

Procedure:

At first, when the switches are open, the corresponding situation is programmed to be HIGH in the arduino. When we push any button of the remote, the logic is converted to LOW and the arduino will send a signal it to the transmitter data pin. For each switch, a corresponding value is send in order to differentiate between the different modes: forward, right, left, back and stop. The transmitter will then transmit this signal wirelessly to the receiver through the antenna pin. The receiver receives this signal and sends it to the arduino. The arduino recognizes the signal and send respective logic values to the motor driver. What combination of logics is to be sent is defined by user defined functions in the program. The driver will receive this combination of logics at the input pins and then send that to the motors through its output pins and start driving the motors correspondingly. This goes the RF controlled part for the robot. The next section is the climbing section. We used the section adhesion in order to stick the robot to the vertical surface. In order to make the suction, we used vacuum pump. The suction cap was attached to the bottom of the chassis. The voltage supplied to the pump determined how much suction will occur and it was adjusted to make sure that it was enough to stick the robot to the wall and at the same time not to much as to hamper the motors in driving the robot up the wall.

The program

// TRANSMITTER CODE
#include <VirtualWire.h>

//Assigning controller buttons to Digital Pins
int forward = 8;
int backward = 9;
intrightTurn = 10;
intleftTurn = 11;
intstopMotor = 12;
intremotePins[] = 
{8,9,10,11,12}; //array to store pin nos

void setup()
{
    Serial.begin(9600);
    Serial.println("setup");
    vw_setup(2000);
    vw_set_tx_pin(3);

    for(int i = 0; i<6 ; i++)
    {
        pinMode(remotePins[i], INPUT);
        digitalWrite(remotePins[i], HIGH);
    }
}

digitalWrite(13, true);
// DECLARING PIN NOS TO FIRST
L293D
int en1 = 5;
int en2 = 6;
int in1 = 7;
int in2 = 8;
int in3 = 9;
int in4 = 10;
// DECLARING PIN NOS TO SECOND
L293D
int en1o2 = 3;
int en2o2 = 11;
int in1o2 = 4;
int in2o2 = 12;
in3o2 = 14;
in4o2 = 15;
intmotorPin[] = {3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15}; // ARRAY FOR STORING PIN NOS

void setup()
{
    Serial.begin(9600);
    Serial.println("setup");
    vw_setup(2000);
    vw_set_rx_pin(2);
    for (int i = 0; i < 12; i++)
    {
        pinMode(motorPin[i], OUTPUT);
    }
}

void loop()
{
    uint8_tbuf[VW_MAX_MESSAGE_LEN];
    uint8_tbuflen = VW_MAX_MESSAGE_LEN;
    if (vw_get_message(buf, &buflen)) // Non-blocking
    {
        int i;
        digitalWrite(13, true);
        for (i = 0; i < buflen; i++)
        {
            Serial.print(buf[i]);
        }
    }
}
forward(); // go forward
Serial.println(" = forward");
}
if(buf[i] == '2')
   {
backward(); // go backward
Serial.println(" = backward");
}
if(buf[i] == '3')
   {
left(); // go left
Serial.println(" = left");
}
if(buf[i] == '4')
   {
right(); // go right
Serial.println(" = right");
}
if(buf[i] == '5')
stopMotor(); // stop/brake
Serial.println(" = stopped");
}
Serial.print(" ");
Serial.println(" ");
digitalWrite(13, false);

void forward()
{
digitalWrite(en1,HIGH);
digitalWrite(en2,HIGH);
digitalWrite(en1o2,HIGH);
digitalWrite(en2o2,HIGH);
digitalWrite(in1,HIGH);
digitalWrite(in1o2,HIGH);
digitalWrite(in2,LOW);
digitalWrite(in2o2,LOW);
digitalWrite(in3,LOW);
digitalWrite(in3o2,LOW);
digitalWrite(in4,HIGH);
digitalWrite(in4o2,HIGH);
}

void backward()
{
digitalWrite(en1,HIGH);
digitalWrite(en2,HIGH);
digitalWrite(en1o2,HIGH);
digitalWrite(en2o2,HIGH);
digitalWrite(in1,LOW);
digitalWrite(in1o2,LOW);
digitalWrite(in2,LOW);
digitalWrite(in2o2,LOW);
digitalWrite(in3,LOW);
digitalWrite(in3o2,LOW);
digitalWrite(in4,LOW);
digitalWrite(in4o2,LOW);
}

void left()
{
digitalWrite(en1,HIGH);
digitalWrite(en2,HIGH);
digitalWrite(en1o2,HIGH);
digitalWrite(en2o2,HIGH);
digitalWrite(in1,LOW);
digitalWrite(in1o2,LOW);
digitalWrite(in2,LOW);
digitalWrite(in2o2,LOW);
digitalWrite(in3,LOW);
digitalWrite(in3o2,LOW);
digitalWrite(in4,HIGH);
digitalWrite(in4o2,HIGH);
}

void right()
{
digitalWrite(en1,HIGH);
digitalWrite(en2,HIGH);
digitalWrite(en1o2,HIGH);
digitalWrite(en2o2,HIGH);
digitalWrite(in1,HIGH);
digitalWrite(in1o2,HIGH);
digitalWrite(in2,LOW);
digitalWrite(in2o2,LOW);
digitalWrite(in3,LOW);
digitalWrite(in3o2,LOW);
digitalWrite(in4,LOW);
digitalWrite(in4o2,LOW);
}

void stopMotor()
{
digitalWrite(en1,HIGH);
digitalWrite(en2,HIGH);
digitalWrite(en1o2,HIGH);
digitalWrite(en2o2,HIGH);
digitalWrite(in1,HIGH);
digitalWrite(in1o2,HIGH);
digitalWrite(in2,LOW);
digitalWrite(in2o2,LOW);
digitalWrite(in3,LOW);
digitalWrite(in3o2,LOW);
digitalWrite(in4,LOW);
digitalWrite(in4o2,LOW);
}
Troubleshooting and improvements

1. Too much weight of the main body causes the object to be unable from climbing up. We can improve this using a lighter body, or making the whole model with lesser components (e.g. two wheels instead of four.)

2. The wheels might not be able to push the body upwards. This is due to insufficient reaction force being applied perpendicular to the surface of the wheels. This could be improved by applying a mechanism which will allow the wheels to be pushed against the wall (e.g. Vacuum suctions can be used on left and right sides along with the wheels).

3. Leakage in the suction cup due to attachment using white tape. This could have been improved by making a plastic model of a tube with the suction cup together.

4. The Rx-Tx modules sometimes don’t pick up the signals appropriately and at a lesser range. This can be improved by using a remote control; however the process would be very expensive.

5. Higher rpm motors could have been used for better pickup.

Applications

A Wall-Climbing Robot system has wide applications include remotely monitoring hazardous environments, reconnaissance, defects inspection, and fire fighting. However, for current needs in areas such as biomedical, aerospace, environmental and military systems, walking or climbing autonomous robots are needed. Object manipulation and surveillance are crucial for many applications, and in many cases, require an ability to climb walls. One notable situation where such a robot could be useful, and perhaps life-saving, is for spacecraft hull inspection and repair. Terrestrial uses include surveillance or inspection in hazardous or difficult-to-reach areas.

References

1. http://nanolab.me.cmu.edu/projects/waalbots/tri-leg.shtml
7. http://www.researchgate.net/publication/3415252_Waalbot_An_Agile_Small-Scale_Wall-Climbing_Robot_Utilizing_Dry_Elastomer_Adhesives/file/e0b495282627b145f0.pdf