Glove Controlled Wheelchair

NEW YORK UNIVERSITY

2015 Fall

Class: Mechatronics
Professor: Vikram Kapila
Date : Dec 14 2015
Name: Parth singh
Sunglyoung Kim
Ishan Darji

Abstract

Using manual wheelchair can causes repetitive strain injury on shoulder and wrist. Conventional motorized wheelchair has joysticks but people who has repetitive strain injury on his or her wrist can not use the joystick. Using flex sensor attached to gloves will give another option to control motorized wheelchair. Also a cliff detector and an obstacle detector will provide a better and safer environment to control electric wheelchair

Introduction

The electric wheelchair was invented in early 1900s but the demand of the chair has risen after World War 2 (electric wheelchair). The conventional way to control an electric wheelchair is by using a joystick. However, people who have repetitive strain injury on his or her wrist can't use the joystick for maneuver a motorized wheelchair. According to "Cooper" using a manual wheelchair can cause repetitive strain injury and roughly 3.6 million people(aged 15 or older) are using a wheelchair (Brault, Matthew W). Controlling power wheelchair with flex sensor gives a benefit to people who have repetitive strain injury on his or her wrist. We will use Flex sensors, an IR sensor, an Ultrasonic sensor to control a wheelchair. The Boe-bot will be considered as a motorized wheelchair and the two continuous servos are used as electrical motor.

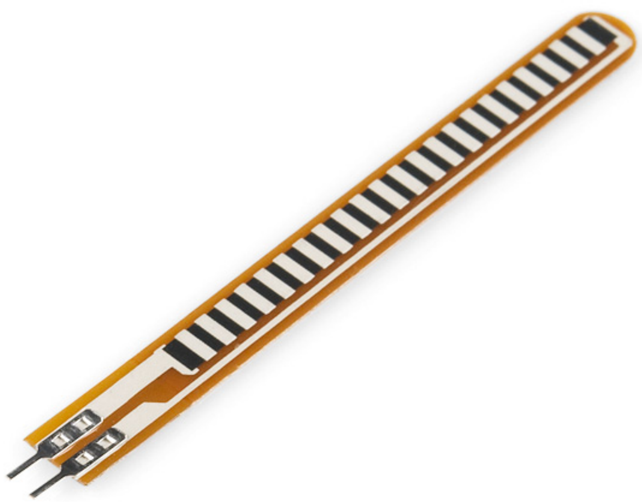
Component Selection

Module	Function
Boe-bot	Represents an electric wheelchair
Servo motors	Imparts motion to the boe-bot
Flex sensors	Control the motion of the servo motors
Ultrasonic sensor	Object Detection
IR sensor	Cliff Detection
LCD display	Interfacing with the user
BS 2 microcontroller	Interface all other components with each other

Table 1 (List of Components)

- In this prototyping project, we select the Boe-Bot from Parallax to represent an electric wheelchair which has two continuous servo motors connected to its wheels to control its motion.
- Use the flex sensors mounted on the user’s hand through wearable glove to control the servo motors of the Boe-bot(wheelchair).
- Use the Ultrasonic sensor to detect objects in front of the Bot-Bot(wheelchair) and stop it from moving forward if it finds an interruption in close proximity.
- Use IR sensors at front and back to detect cliff (stairs or absence of landing in case of wheelchairs) and stop its motion if it finds a cliff.
- LCD display is used to interface with the user and display appropriate information.
- We use the BASIC STAMP 2 to control all these components.

Flex Sensors
Flex sensor is an analog variable resistance sensor with two leads. It changes its resistance output when it is bent. The change in resistance is proportional to the amount of bending.



Inside the flex sensor there are carbon resistive elements within a thin flexible substrate. When the substrate is bent, the sensor outputs a resistance relative to the bend radius. The nominal resistance value of the flex sensor when it is straight is 15KΩ to 22KΩ when it is bent to 90° angle.

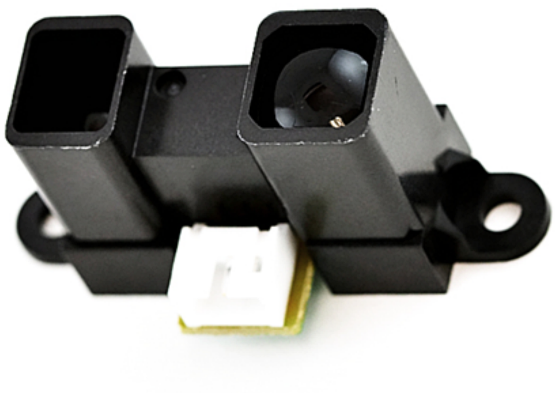
Servo Motors

There is two kind of servo motors,e.g. continuous servo and standard servo. Standard servo has only angular motion of 180°. Continuous motors, as the name suggests, can provide a continuous rotary motion. That is why continuous servos are used to impart motion to our Boe-Bot(wheelchair). The speed of the servo motor is controlled by a method called Pulse Width Modulation.

Features: Bidirectional continuous rotation 0 to 50 RPM, with a linear response to PWM for easy ramping Accepts four mounting screws Easy to interface with any Parallax microcontroller or PWM-capable device Very easy to control with the PULSOUT command in PBASIC or SX/B Weighs only 1.50 oz (42.5 g) 38 oz-in torque @ 6 V Key Specifications Power requirements: 4 to 6 VDC; Maximum current draw 140 +/- 50 mA at 6 VDC when operating in no load conditions, 15 mA when in static state Communication: pulse-width modulation Dimensions: approx 2.2 x 0.8 x 1.6 in (5.58x 1.9 x 4.06 cm)

PWM (Pulse width Modulation)
The speed of servo is controlled by duty cycle. When the pulse is high the motor will be propelled and when the pulse is low the motor is rotated by inertia. The duty cycle is pulse high time over period time. For instance, If the duty cycle is 100% the servo is operated in full speed, and if the duty cycle is 50% the servo is operated at half the speed.

IR Sensor



Absolute Maximum Ratings

parameter	symbol	rating	unit
Supply voltage	V_{DD}	-0.3 to +7	V
Output terminal voltage	V_O	-0.3 to VCC +0.3 V	V
Operating temperature	Topr	-10 to +60 °C	°C
Storage temperature	Tstg	-40 to +70 °C	°C

Detecting Distance : 10 to 80 cm

The IR sensor is used as cliff detector. One is attached in front of the wheelchair and the other one is attached in the back. When user goes to a new place he or she will experience safer and confidence with the cliff detector.

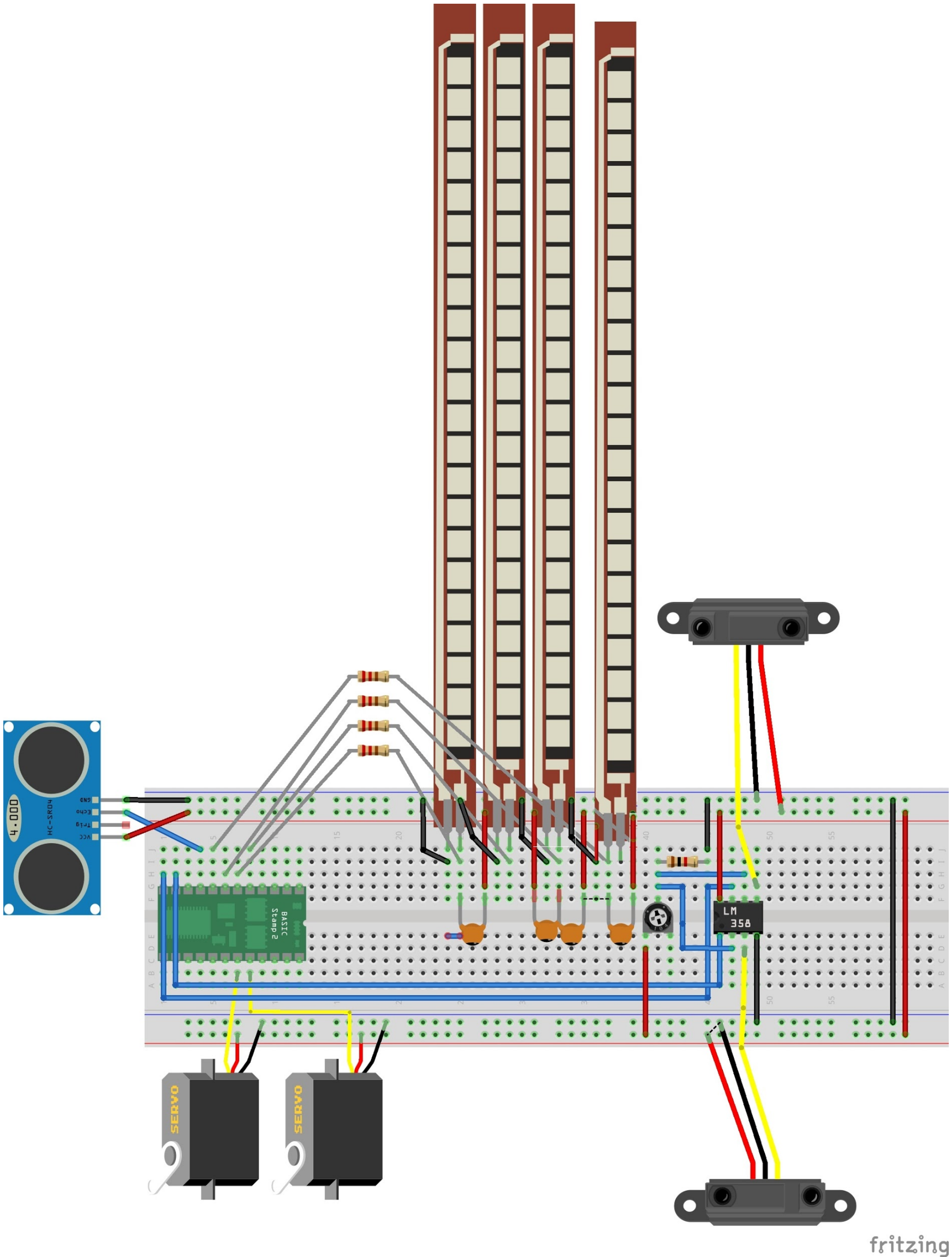
Ultrasonic Sensor

The ping sensor provided by parallax is used as a front obstacle detector
Specifications of the ping sensor:
Narrow acceptance angle
Range: approximately 1 inch to 10 feet (2 cm to 3 m)
3-pin male header with 0.1" spacing
Power requirements: +5 VDC; 35 mA active
Communication: positive TTL pulse
Dimensions: 0.81 x 1.8 x 0.6 in (22 x 46 x 16 mm)
Operating temperature range: +32 to +158 °F (0 to +70 °C)

Basic Stamp 2
Specifications:

Processor Speed: 20 MHz; ~4,000 PBASIC instructions/sec
PBASIC Commands: 42
Package: 24-pin DIP
I/O pins: 16 + 2 dedicated serial
RAM Size: 32 Bytes (6 I/O, 26 Variable)
EEPROM (Program) Size: 2 KBytes; ~500 PBASIC instructions
Voltage requirements: 5.5 to 15 VDC (Vin), or 5 VDC (Vdd)
Current requirements: 3 mA Run, 50 µA Sleep
Source/Sink Current per I/O: 20 mA / 25 mA
Source/Sink Current per unit: 40 mA / 50 mA per 8 I/O pins
Communication: Serial (9600 baud for programming)
Dimensions: 1.20 x 0.63 x 0.15 in (30.0 x 16.0 x 3.81 mm)
Operating temp range: -40 to +185 °F (-40 to +85 °C)

Circuit Design



Mathematical Background

RCtime

The resistance of the flex sensors change when we bend them, ultimately changing the RCtime of the particular RC circuit. We use the four flex sensors mounted on four fingers to control the direction of movement and the amount of bend to control the speed.

The RC time of different flex sensors is used to change the duty cycle in each case.

The R value for the flex sensor ranges from 15KΩ-22KΩ and C value for the capacitor is 0.1 μF. Therefore, the RCtime value is from 1500 to 2200, which is calculated by 15KΩ * 0.1 μF < RCtime < 22KΩ * 0.1 μF.

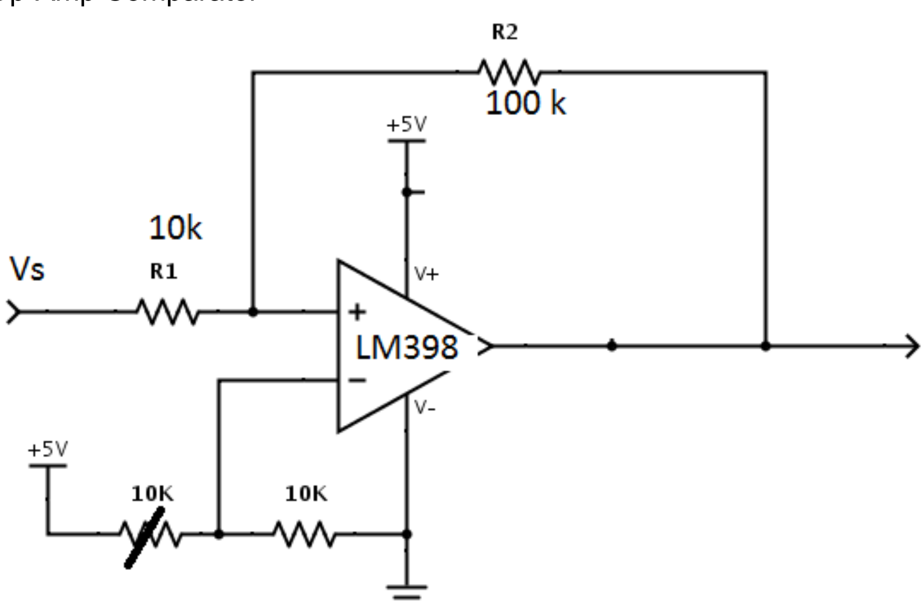
Scaling for servo control

The flex sensors provide RCtime values from 1500 to 3000 and that is scaled to a 750 to 250 range using the formula

```
'Flex Sensor1
F1:
  HIGH 0 'hold pin 0 high to charge capacitor
  PAUSE 1 'charge capacitor for 1/1000 of a second
  RCTIME 0, 1,fs1 'measure discharge time of flex resistor through capacitor
  'fs1=fs1**scalecon1+500
  temp = fs1*/439 - 2572
  fs1 = fs1 - temp
  fs1 = fs1/2

  RETURN
```

Op-Amp Comparator



The OpAmp compares the IR analog voltage to the pot voltage to check if cliff is present

CODE

```
' {$STAMP BS2}
```

```
' {$PBASIC 2.5}
' {$PORT COM3}

' -----[ I/O Definitions ]-------------------------------------------------

RightServo      PIN     15              ' Right Servo
LeftServo       PIN     14              ' Left Servo
lcd             PIN      8
' -----[ Constants ]-----------------------------------------------------

scalecon1 CON 11000    'Constant for Flex Sensor 1
scalecon2 CON 10850    'Constant for Flex Sensor 2
scalecon3 CON 10800    'Constant for Flex Sensor 3
scalecon4 CON 10800    'Constant for Flex Sensor 4

distcon  CON 2257

' -----[ Variables ]-----------------------------------------------------

fs1 VAR Word
fs2 VAR Word
fs3 VAR Word
fs4 VAR Word
ul1 VAR Word
ir1 VAR Word
ir2 VAR Word
temp VAR Word
pulsecount VAR Word
PS VAR Word
RS VAR Word
LS VAR Word

' -----[ Program Code ]---------------------------------------------------


Main:
DO
DEBUG CLS
GOSUB ld
GOSUB f1
GOSUB f2
GOSUB f3
GOSUB f4
GOSUB ping
```

```
GOSUB irs
rs = 750
ls = 750


IF (fs3 < 450) AND (fs4 <450) THEN  'Stop switch
 DO
 IF(fs2 < 450) AND (fs1 < 450) THEN  'Reset the stop switch
   SEROUT lcd, 84, [12]
   PAUSE 5
   SEROUT lcd, 84, [128,"Reset Cleared"]
   PAUSE 5
   DEBUG "reset cleared"
   GOTO main
 ELSE
   DEBUG "reset"
   SEROUT lcd, 84, [12]
   PAUSE 5
   SEROUT lcd, 84, [128,"STOP1 Switch"]
   PAUSE 5
   GOSUB f3
   GOSUB f4
 ENDIF
 LOOP
ENDIF

DO WHILE (ul1 < 20)                    'Obstacle detection
 PULSOUT LeftServo, 750                ' Left Servo Move Pulse Value
 PULSOUT RightServo, 750               ' Right Servo Move Pulse Value
 DEBUG CR,"STOP UltraSonic"
 SEROUT lcd, 84, [12]
 PAUSE 5
 SEROUT lcd, 84, [128,"STOP2 Ultrasonic"]
 PAUSE 10

 GOSUB ping
 PAUSE 20
LOOP

IF (fs1 > 650) AND (fs1 < 850) THEN     'Speed Control Using FLex Sensor
DEBUG CR,"stationary1"

ELSEIF (fs1 < 650) AND (fs1 > 550) THEN
DEBUG CR,"fw1"
```

```
SEROUT lcd, 84, [12]
PAUSE 5
SEROUT lcd, 84, [128,"Forward"]
PAUSE 5
LS = 850
RS = 650
PS = 20
GOTO mov

ELSEIF (fs1 < 550) AND (fs1 > 450) THEN
DEBUG CR,"fw2"
LS = 800
RS = 600
PS = 20
SEROUT lcd, 84, [12]
PAUSE 5
SEROUT lcd, 84, [128,"Forward"]
PAUSE 5
GOTO mov
ELSEIF (fs1 < 450)   THEN
DEBUG CR,"fw3"
LS = 780
RS = 720
PS = 20
SEROUT lcd, 84, [12]
PAUSE 5
SEROUT lcd, 84, [128,"Forward"]
PAUSE 5
GOTO mov
ENDIF

IF (fs2 > 650) AND (fs2 < 850)  THEN    'Speed Control Using FLex Sensor
DEBUG CR,"stationary2"
ELSEIF (fs2 < 650) AND (fs2 > 550)  THEN
DEBUG CR,"bck1"
SEROUT lcd, 84, [12]
PAUSE 5
SEROUT lcd, 84, [128,"BACK"]
PAUSE 5
RS = 850
LS = 650
PS = 20
GOTO mov
ELSEIF (fs2 < 550) AND (fs2 > 450)  THEN
```

```
DEBUG CR,"bck2"
SEROUT lcd, 84, [12]
PAUSE 5
SEROUT lcd, 84, [128,"BACK"]
PAUSE 5
RS = 800
LS = 700
PS = 20
GOTO mov
ELSEIF (fs2 < 450)  THEN
DEBUG CR,"bck3"
SEROUT lcd, 84, [12]
PAUSE 5
SEROUT lcd, 84, [128,"BACK"]
PAUSE 5
RS = 780
LS = 720
PS = 20
GOTO mov
ENDIF

IF (fs3 > 650) AND (fs3 < 850)   THEN    'Speed Control Using FLex Sensor
DEBUG CR,"stationary3"
ELSEIF (fs3 < 650) AND (fs3 > 550)   THEN
DEBUG CR,"rw1"
SEROUT lcd, 84, [12]
PAUSE 5
SEROUT lcd, 84, [128,"RIGHT"]
PAUSE 5
RS = 650
LS = 650
PS = 20
GOTO mov
ELSEIF (fs3 < 550) AND (fs3 > 450)   THEN
DEBUG CR,"rw2"
SEROUT lcd, 84, [12]
PAUSE 5
SEROUT lcd, 84, [128,"RIGHT"]
PAUSE 5
RS = 700
LS = 700
PS = 20
GOTO mov
ELSEIF (fs3 < 450)   THEN
```

```
DEBUG CR,"rw3"
SEROUT lcd, 84, [12]
PAUSE 5
SEROUT lcd, 84, [128,"RIGHT"]
PAUSE 5
RS = 720
LS = 720
PS = 20
GOTO mov
ENDIF

IF (fs4 > 650) AND (fs4 < 850)   THEN    'Speed Control Using FLex Sensor
DEBUG CR,"stationary4"
ELSEIF (fs4 < 650) AND (fs4 > 550)   THEN
DEBUG CR,"lf1"
SEROUT lcd, 84, [12]
PAUSE 5
SEROUT lcd, 84, [128,"LEFT"]
PAUSE 5
RS = 850
LS = 850
PS = 20
GOTO mov
ELSEIF (fs4 < 550) AND (fs4 > 450)   THEN
DEBUG CR,"lf2"
RS = 800
LS = 800
PS = 20
SEROUT lcd, 84, [12]
PAUSE 5
SEROUT lcd, 84, [128,"LEFT"]
PAUSE 5
GOTO mov
ELSEIF (fs4 < 450)   THEN
DEBUG CR,"lf3"
SEROUT lcd, 84, [12]
PAUSE 5
SEROUT lcd, 84, [128,"LEFT"]
PAUSE 5
RS = 780
LS = 780
ps = 20
GOTO mov
ENDIF
```

```
'DEBUG ? fs1,? fs2,? fs3,? fs4
PAUSE 500
LOOP


'Initialize LCD
LD:
  SEROUT lcd, 84, [22, 12] 'Initialize LCD
  PAUSE 5
  SEROUT lcd, 84, [128,"Glove Controlled", 148, "      BoeBot "]
  PAUSE 2000



' ***********************************************************************
' * USE THE INFRARED SENSORS TO DETECT CLIFFS                        *
' * WHILE THE PING))) IS FACING FORWARD.                          *
' ***********************************************************************

'IR sensor
IRS:                                'Check for IR Sensor 1 and 2 HIGH
  DO WHILE(IN6 = 0) OR (IN7 = 0)
  PULSOUT LeftServo, 750              ' Left Servo Move Pulse Value
  PULSOUT RightServo, 750              ' Right Servo Move Pulse Value
  DEBUG CR,"STOP IR Sensor"
  SEROUT lcd, 84, [12]
  PAUSE 5
  SEROUT lcd, 84, [128,"CLIFF DETECTED"]
  PAUSE 5
  PAUSE 20
  LOOP

' ***********************************************************************
' * USE THE ULTRASONIC SENSOR TO GET DISTANCE VALUES                   *
' * WHILE THE PING))) IS FACING FORWARD.                          *
' ***********************************************************************

'Ping sensor
PING:                               'Ultrasonic Distance Measuring
  PULSOUT 4, 5
  PULSIN 4, 1, ul1
  ul1 = distcon ** ul1
  DEBUG CR,DEC3 ul1, " cm"
```

```
  RETURN

' ************************************************************************
' * USE THE RCTIME VALUES TO GET THE FLEX SENSOR VALUES                 *
' * WHILE THE PING))) IS FACING FORWARD.                                *
' ************************************************************************


'Flex Sensor1
F1:
  HIGH 0 'hold pin 0 high to charge capacitor
  PAUSE 1 'charge capacitor for 1/1000 of a second
  RCTIME 0, 1,fs1 'measure discharge time of flex resistor through capacitor
  'fs1=fs1**scalecon1+500
  temp = fs1*/439 - 2572
  fs1 = fs1 - temp
  fs1 = fs1/2

  RETURN

'Flex Sensor2
F2:
  HIGH 1 'hold pin 0 high to charge capacitor
  PAUSE 1 'charge capacitor for 1/1000 of a second
  RCTIME 1, 1,fs2 'measure discharge time of flex resistor through capacitor
'  fs2=fs2**scalecon2+500
  temp = fs2*/439 - 2572
  fs2 = fs2 - temp
  fs2 = fs2/2

  RETURN

'Flex Sensor3
F3:
  HIGH 2 'hold pin 0 high to charge capacitor
  PAUSE 1 'charge capacitor for 1/1000 of a second
  RCTIME 2, 1,fs3 'measure discharge time of flex resistor through capacitor
  'fs3=fs3**scalecon3+500
  temp = fs3*/439 - 2572
  fs3 = fs3 - temp
  fs3 = fs3/2

  RETURN
```

```
'Flex Sensor4
F4:
  HIGH 3 'hold pin 0 high to charge capacitor
  PAUSE 1 'charge capacitor for 1/1000 of a second
  RCTIME 3, 1,fs4 'measure discharge time of flex resistor through capacitor
  'fs4=fs4**scalecon4+500
  temp = fs4*/439 - 2572
  fs4 = fs4 - temp
  fs4 = fs4/2

  RETURN

' *************************************************************************
' * USE THE APPROPRIATE PULSOUT VALUES TO MAKE YOUR BOE-BOT MOVE          *
' * WHILE THE PING))) IS FACING FORWARD.                                  *
' *************************************************************************

MOV:                                  ' Send Moving Pulse
  FOR pulseCount = 0 TO 200
   PULSOUT LeftServo, LS              ' Left Servo Move Pulse Value
   PULSOUT RightServo, RS             ' Right Servo Move Pulse Value
   PAUSE PS                          'Refresh Delay using values passed from speed control
  NEXT
  RETURN
```

Bill of materials and Prototype cost

Bill of Material for Mechatronics Term Project				
Item	Cost per Item	No of Item	Total	Mass Production
Flex Sensor	7.49	4	29.96	20
IR Sensor	12	2	24	16
Wires	3	3	9	3
Ultrasonic	15	1	15	12
Boebot Chasis	59	1	59	40
Total			136.96	91

Cost for mass production

For mass production the Boe bot kit would be replaced by an actual electric wheelchair or a custom built wheelchair which would be the primary cost of the appliance. Mass supply of flex sensors, IR sensors and Ultrasonic sensors would further reduce the cost of the glove to not be a significant portion of the total cost. Estimated cost for the mass production units would range from 350 - 500 USD and based on future customer surveys focusing on willingness to pay vs features the price estimate for the market would be established.

Pros and cons

The current prototype uses PARALLAX BS2 as the heart of the system which being an fundamentally good microcontroller lacks certain features like internal A2D converter and interrupts. This sets the limitations of the prototype as the use of separate A2D converters and comparators means more external wiring increasing the chance of malfunction and the absence of interrupts and polling means a longer code pertaining to the activation and deactivation of the system for safety and security purposes.

On the positive side BS2 being a microcontroller with a small form factor can be easily integrated into the future products with use of modules that are absolutely necessary and discarding those which aren't required but are present by default on competitor boards.

Future module additions

- The glove can be made wireless using RF/Xbee or Bluetooth modules which will give the user better flexibility and freedom.
- There are 16 different combinations of user inputs available to be programmed for the user ease of access and can be used to switch on lights using relays, activate doors inside home and the implications are limitless.

- With the increasing use of Internet of things (IOT) the board can be connected to the Internet services like Microsoft Azure or IBM Bluemix and the reach of the gloves can be increased manifolds.

Conclusion

The basic stamp 2 based glove controlled wheelchair prototype is a product that will usher the world into the age of joystick less wheelchair control and freedom to use the appendage for other tasks.

References

Brault, Matthew W., “Americans With Disabilities: 2010,” Current Population Reports, P70-131, U.S. Census Bureau, Washington, DC, 2012.

Cooper, R and **Boninger, Michael Lee** and **Robertson, R** (1998) *Heavy Handed: Repetitive strain injury among manual wheelchair users.* Team Rehab Report, 9 (2). pp. 35-38.