Polytechnic Institute of New York University (NYU-Poly)

Department of Mechanical and Aerospace Engineering

ME 5643 Mechatronics

Fall 2012

Integrated term project

SMART TREADMILL

Fouad Issa

Arti Karkar

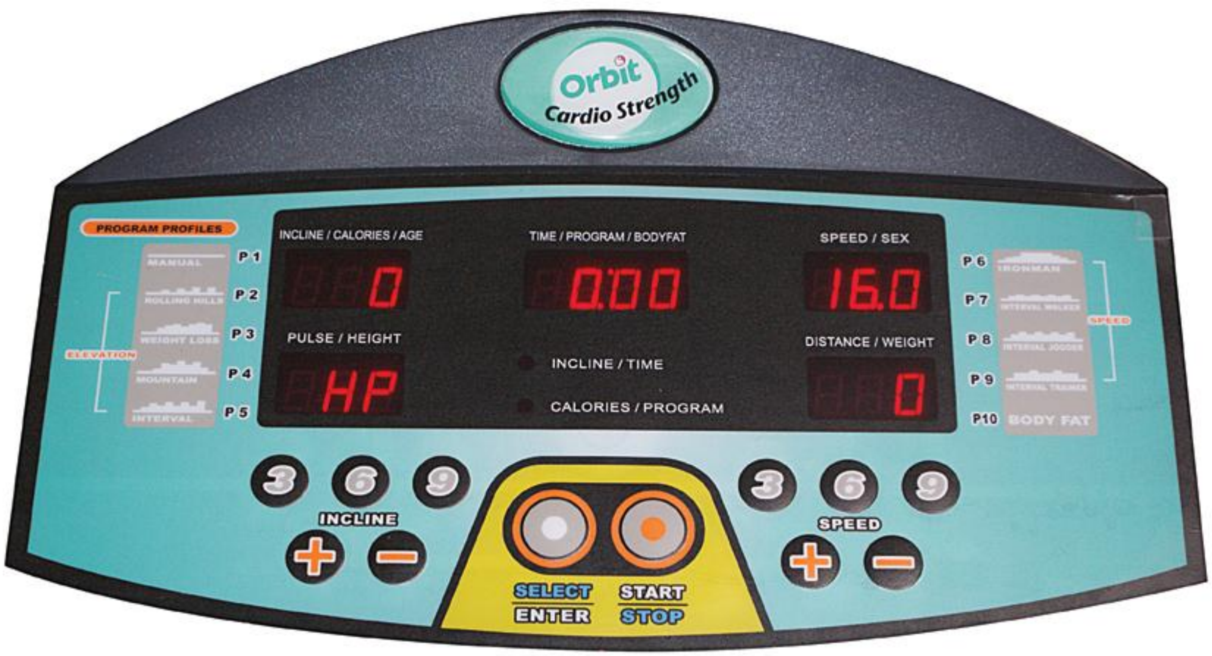
Rishi Raghuvanshi

CONTENT

1. Project motivation:

The project is motivated to make smart exercise equipment. The current date exercise treadmills are incorporated with multiple features like

- Change the speed of the treadmill
- Change elevation of the treadmill
- Choosing a workout regime based on the type of workout intended – fat burn, cardio etc.
- Displaying distance covered and calories burnt



User input is required to change these settings and enter weight, heart rate (through incorporated sensors), and workout time in order to predict the number of calories burnt and to give a workout summary. Our smart treadmill senses the user's position and adjusts the motor speed accordingly for the runner to remain in the center. Photoresistors accurately trace the position of the user and drive the motor to vary the speed through its interface with the microcontroller. This enables the user to continue the workout without any disruption in case the user does an intense workout and needs to slow down; the treadmill will automatically sense the need to slow down according to the user's speed.

2. Mathematical background

2.1. Calculations for resistors

Total number of LED's used =3

Total number of RCTime circuits used = 3

Current drawn by individual LEDs and RC circuits = 50mA/6

The voltage drop across each resistor is 3.6V

Therefore voltage resistance required for each LED = 3.6*6*1000/(50) = 432 ohm

Thus using the closest 470 ohm, keeps the circuit safe as it draws 3.6/470= 7.6 mA.

3. Project prototype

3.1. Mechanical design

3.1.1. Motor Assembly

Parallax boe-bot kit assembly is used with continuous rotation servos to run the treadmill belt between the two wheels and two rollers. The parallax standard servo with a lever assembly is used to control the inclination of the track.

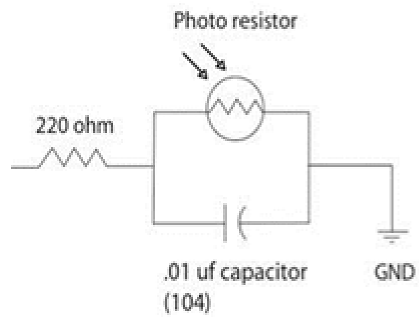
3.1.2. Tread Assembly

The tread is made using strips of paper with a clear strip of plastic running down the middle. The clear center strip allows for the user to cast a shadow on the photoresistors. This arrangement provides us the required structural strength to provide a functioning track.

3.2. Electronic circuit design

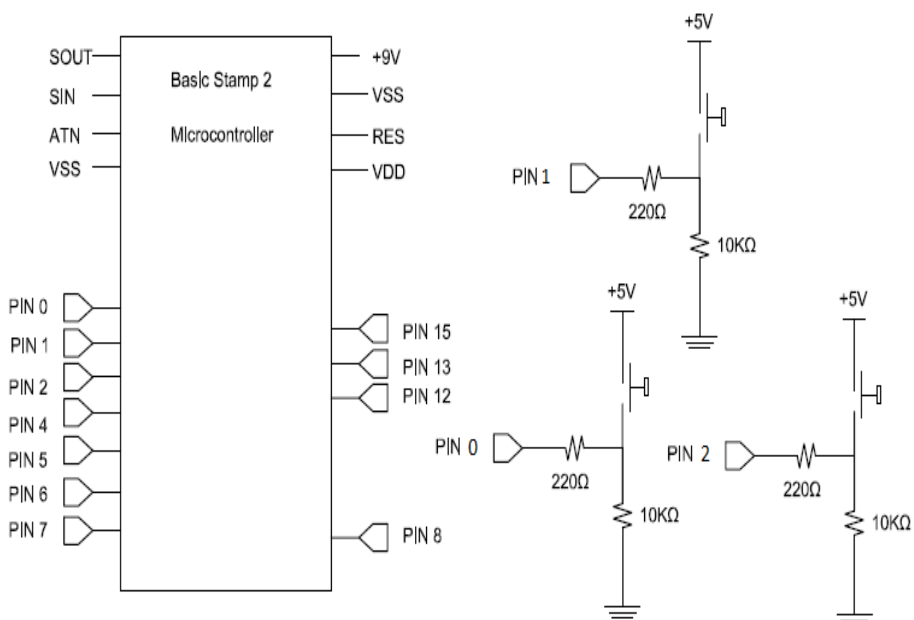
3.2.1. User position sensors

Three photoresistors are appropriately interfaced with the microcontroller using a parallel RC circuit, with a 1kΩ resistor and a 0.1µF capacitor. The RCTime value of this circuit is measured using the BasicStamp 2. This circuit gives a higher RCTime value when the photoresistor is covered, this allows us to adjust the speed of the track by comparing the RCTime values from the end resistors to the one in the center.



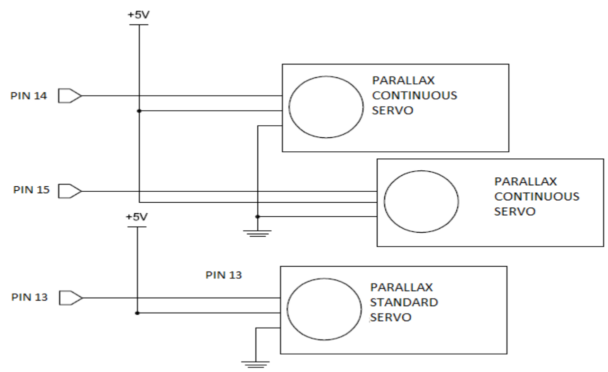
3.2.2. User Control Buttons

Three buttons are used to control the treadmill. These buttons are connected in an normally open active high arrangement. The first button allows the user to start and stop the treadmill at any time. The second and third buttons allow the user to adjust the incline of the treadmill.



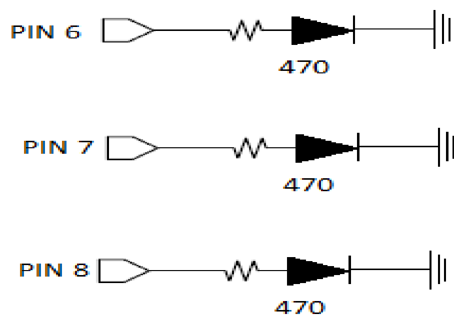
3.2.3. Motors

The parallax continuous rotation servos are interfaced appropriately with the basic stamp 2 microcontroller to run the track. Each of the continuous servo motors operate in opposite directions to keep the track moving properly. Parallax standard servo along with a lever assembly is used to raise or lower the treadmill track depending on the user input.

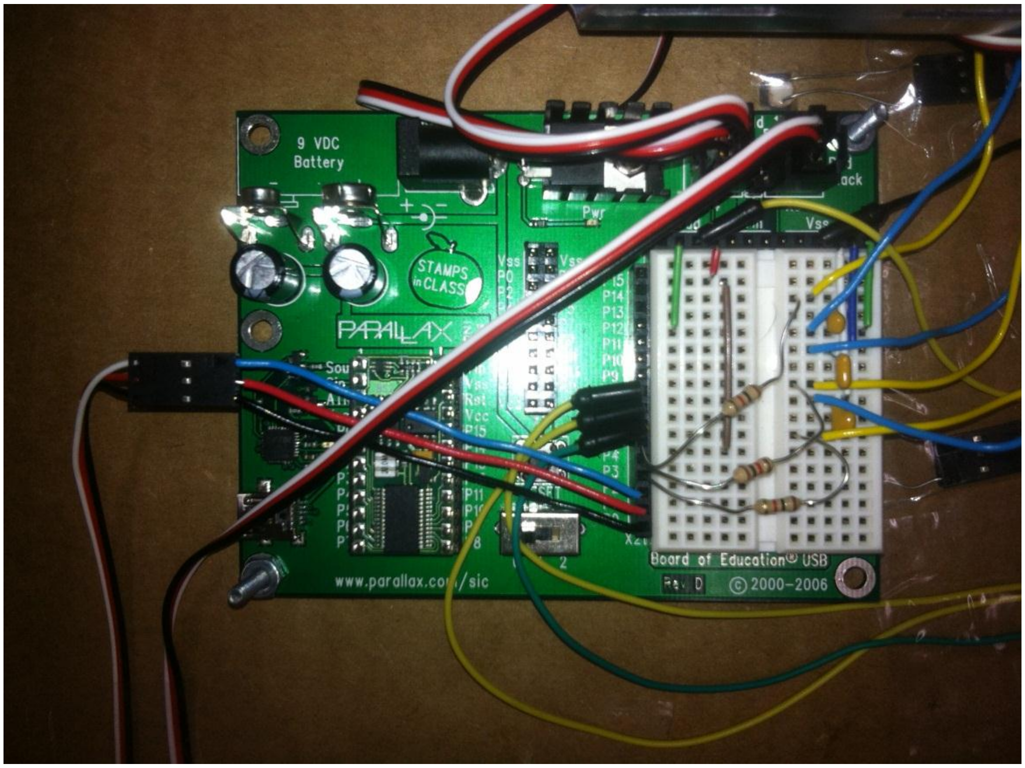


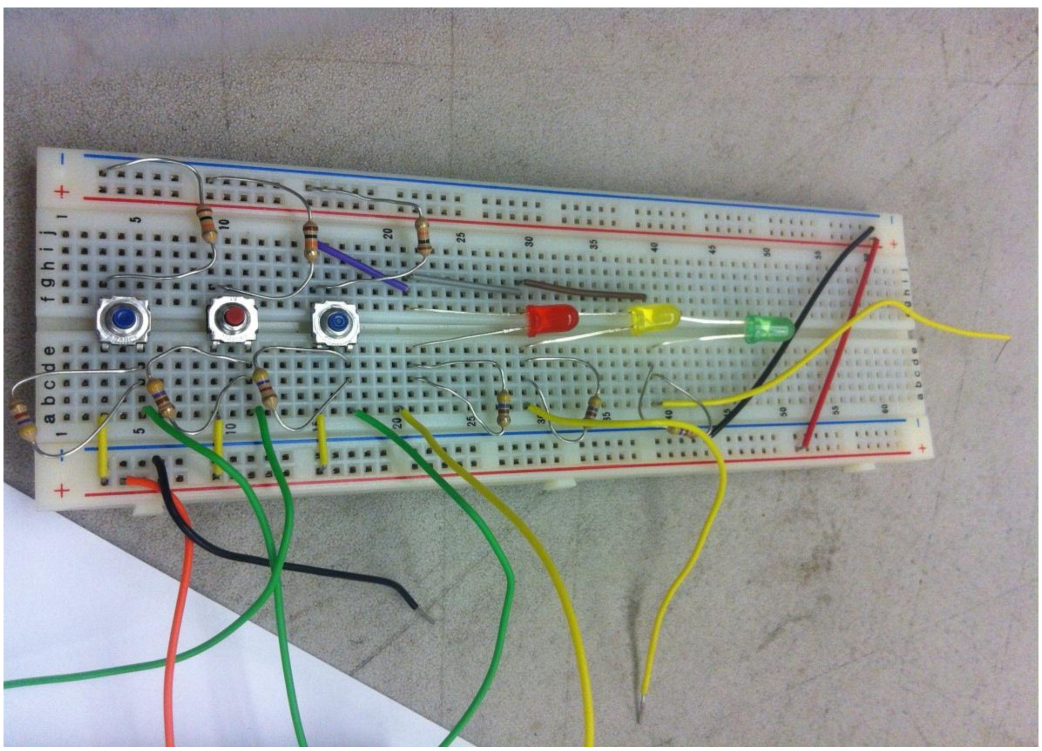
3.2.4. User position LED Indicators

Three different colored light emitting diodes (LEDs) are interfaced with the basic stamp 2 microcontroller to display the current state of user. Appropriate current limiting resistors are connected in the circuit to keep the circuit safe.



3.2.5 Real Circuits





4. **Bill of Material:**

Following table we give the list of all the materials

Sr. no	Part Name	Quantity
1	Boe-bot kit with board of education	1
2	3 pin female-female headers	3
3	Parallax standard servo	1
4	Photo resistor	3
5	Light Emitting Diode (Green)	1
6	Light Emitting Diode (Yellow)	1
7	Light Emitting Diode (Red)	1
8	Button	3
9	Battery	4
10	Clear Film (for Track)	1.0
11	Male-female Breadboard wires	40
12	Breadboard	1
13	Resistor (470 ohm)	6
14	Resistor (10 kohm)	3
15	Capacitors(0.1 micro Farad)	3

5. <u>Prototype cost:</u>

Table below we give the bill of 1 model.

Sr. no	Part Name	Quantity	Price($)	Total Price($)
1	Boe-Bot Robot kit with board of education	1	143.99	143.99
2	3 pin female-female headers	3	0.5	1.5
3	Parallax standard servo	1	24.99	24.99
4	Photo resistor	3	1.0	3.0
5	Light Emitting Diode (Green)	1	0.1	0.1
6	Light Emitting Diode (Yellow)	1	0.1	0.1
7	Light Emitting Diode (Red)	1	0.1	0.1
8	Button	3	0.3	1.0
9	Battery	4	1.0	4.0
10	Clear Film (for Track)	1.0	2.0	2.0
11	Male-female Breadboard wires	40	NA	5.95
12	Breadboard	1	8.50	8.50
13	Resistor (470 ohm)	6	0.20	1.20
14	Resistor (10 kohm)	3	0.20	0.60
15	Capacitors(0.1 micro Farad)	3	0.20	0.60
TOTAL				$ 203.43

6. Cost analysis (Mass production)

Cost of mass production for making the above device on a large scale.

Sr. no	Part Name	Quantity	Price($)	Total Price($)
1	Boe-Bot Robot kit with board of education	1	143.99	115
2	3 pin female-female headers	3	0.5	1.5
3	Parallax standard servo	1	24.99	20.0
4	Photo resistor	3	1.0	3.0
5	Light Emitting Diode (Green)	1	0.1	0.1
6	Light Emitting Diode (Yellow)	1	0.1	0.1
7	Light Emitting Diode (Red)	1	0.1	0.1
8	Button	3	0.3	1.0
9	Battery	4	1.0	4.0
10	Clear Film (for Track)	1.0	2.0	2.0
11	Male-female Breadboard wires	NA	5.95	2.0
12	Breadboard	1	8.50	4.50
13	Resistor (470 ohm)	6	0.20	1.20
14	Resistor (10 kohm)	3	0.20	0.60
15	Capacitors(0.1 micro Farad)	3	0.20	0.60
TOTAL				$ 155.7

6.1. Cost Comparison

Cost of manufacturing of one device was reduced from $ 203.43 to $ 155.7 when produced in mass. This gives a reduction of cost by

(203.243-155.7)/203.43*100=23.46%

7. <u>Discussion:</u>

7.1. Advantages

- Automatic adjustment of speed keeps the user concentrated on track, this improves user safety.
- The use of LED speed display unit gives user a feedback about his speed on track.
- User doesn’t have to force himself through the workout as the reduction in running speed makes the treadmill go slow and as the user starts to run faster it increases the motor speed.

7.2. Disadvantages

- The material of the track needs to be made of out of a transparent material for the photo resistors to work.
- Photoresistors will not function properly in the dark
- Material for track and roller should have enough friction to cause movement. But, should not be too much to stop motors from movement. Hence use of paper and a thin film for clear middle surface and appropriate friction.

8. Program Code

8.1. Main Code

```
' {$STAMP BS2}

' {$PBASIC 2.5}

close VAR Word           'photo resistor near front of treadmill, pin 2
far VAR Word              'photo resistor near end of treadmill, pin 4
center VAR Word      'photo resistor near middle of treadmill, pin 3
speed1 VAR Word
'pulse width for motor 1 speed control, 750<speed1<1000, pin 15
speed2 VAR Word
'pulse width for motor 2 speed control, 750<speed1<1000, pin 14
elevation VAR Word
counter VAR Nib
elevation = 1050
speed1 = 750               'pulse speed for motor to be stopped
speed2 = 750
counter = 0
HIGH 3                     'Charging RCTime Capacitors on pin3
HIGH 4                     'Charging RCTime Capacitors on pin4
HIGH 5                     'Charging RCTime Capacitors on pin5
DEBUG "Press button to turn the treadmill ON."
DEBUG CR
DEBUG "Pressing the button again will turn the treadmill OFF"
DEBUG CR
DEBUG "Elevation of the track can be Increased or Decreased via Appropriate button press"
DEBUG CR
DEBUG "CAUTION! Do not stand still on the treadmill while it’s still in motion, it may cause
injuries."
DEBUG CR
DEBUG CR
DEBUG "The LED's can be monitored for the speed of treadmill."
DEBUG CR
DEBUG "........GREEN-Accelerating......"
DEBUG CR
DEBUG "........YELLOW-Constant Speed..."
DEBUG CR
DEBUG "........RED-De-accelerating......"
DEBUG CR
```

```
DEBUG "........All LED’s Blinking-Top Speed......"

start:
DO
DO:LOOP UNTIL IN0=1
DO:LOOP UNTIL IN0=0
DO
LOW 6
LOW 7
LOW 8
DEBUG CLS
GOSUB stopcheck              'Check if stop button has been pressed
GOSUB elevationcheck    ‘check the height of treadmill
GOSUB position               'Subroutine to get position of runner
GOSUB keepspeed  'Subroutine to maintain speed of the treadmill
GOSUB accelerate             'Subroutine to accelerate the treadmill
GOSUB deccelerate            'Subroutine to decelerate the treadmill
GOSUB stopcheck     'subroutine to check stop button press
GOSUB elevationcheck   'Check if elevation buttons are being pressed
IF (speed1>999)THEN            'Check if exceeding maximum speed
speed1 = 1000                  'Limit motor speeds to maximum speeds
speed2 = 500                   'Limit motor speeds to maximum speeds
HIGH 6                        'Warns user that he has reached maximum speed
HIGH 7
HIGH 8
ELSEIF (speed1<750)THEN   'Check if minimum speed has been reached
speed1 = 750                  'Prevent Treadmill from running backwards
speed2 = 750
ENDIF
GOSUB stopcheck
GOSUB elevationcheck
PULSOUT 15, speed1           'Send pulse to the motors
PULSOUT 14, speed2
PAUSE 5
GOSUB stopcheck
GOSUB elevationcheck
LOOP
PAUSE 200
LOOP
```

8.2 Subroutines

```
stoptreadmill:  'Subroutine to stop the treadmill incase button has been pressed
speed1 = 750
speed2 = 750
PULSOUT 15, speed1
PULSOUT 14, speed2
PAUSE 500
LOW 6
LOW 7
LOW 8
GOTO start
RETURN

keepspeed: ‘keep the speed constant
IF (center>close) AND (center>far) THEN
counter = 0
speed1=speed1
speed2=speed2
HIGH 7
ENDIF
RETURN

accelerate: ‘ increase the speed
IF (center<far) THEN
counter = counter + 1
speed1=speed1+counter
speed2=speed2-counter
HIGH 8
ENDIF
RETURN

deccelerate: ‘decrease the speed
IF (center<close) THEN
counter = counter + 1
speed1 = speed1 - counter
speed2 = speed2 + counter
HIGH 6
ENDIF
RETURN
```

```
position: ‘subroutine to check position of user on track
PAUSE 5
RCTIME 3, 1, far
RCTIME 4, 1, center
RCTIME 5, 1, close
HIGH 3
HIGH 4
HIGH 5
RETURN

stopcheck: ‘subroutine to check stop button press
IF IN0=1 THEN
GOSUB stoptreadmill
ENDIF
RETURN

elevationcheck: ‘subroutine to maintain height
IF IN1=1 THEN
elevation = elevation - 2
ELSEIF IN2=1 THEN
elevation = elevation + 2
ENDIF
IF (elevation > 1050) THEN
elevation = 1050
ELSEIF (elevation < 600) THEN
elevation = 600
ENDIF
PULSOUT 13, elevation
RETURN
```