ME 5643

Integrated Term Project

Mechatronics

Guangzhou Li – gl1439 – N1778058
Apratim Bajpai – ab6462 – N1
Haareeth Parimi – vhp219 – N13733624
<table>
<thead>
<tr>
<th>Serial Number</th>
<th>Heading</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Motivation</td>
<td>1</td>
</tr>
<tr>
<td>2.</td>
<td>Goal</td>
<td>1</td>
</tr>
<tr>
<td>3.</td>
<td>Theory</td>
<td>1</td>
</tr>
<tr>
<td>4.</td>
<td>Code</td>
<td>2</td>
</tr>
<tr>
<td>5.</td>
<td>Electronic Circuits</td>
<td>13</td>
</tr>
<tr>
<td>6.</td>
<td>Bill of material</td>
<td>14</td>
</tr>
<tr>
<td>7.</td>
<td>Improvements</td>
<td>15</td>
</tr>
<tr>
<td>8.</td>
<td>Future Development</td>
<td>15</td>
</tr>
</tbody>
</table>
1. Motivation

Who doesn’t love watching kids grow up and play around? It’s a joy to watch children learn new words. With this thought in mind we have come up with a system that helps kids learn words fun way with interactive sensors and monitors the surrounding environments.

2. Goal

To develop a teaching and monitoring system that gives feedback to children in a fun way and helps them.

3. Theory

We use potentiometer and ADC together to distinguish different alphabets in English. Based on the voltage the wiper terminal creates we determine words and point to the word. As a monitoring System we have a digital thermometer that measures the temperature of the milk bottle. The thermometer would glow “RED” if the temperature is too hot, ”BLUE” if it’s too cold and “GREEN” if its optimum. We also have a night lamp that would turn on when there is not enough light in the room and turn on a light lamp. The user has freedom to change the color of light.
4. CODE:

' ' ' {$STAMP BS2}
' ' {$SPBASIC 2.5}
' '
' =================================================================================================

' -----[ Program Description ]---------------------------------------------------------------

' -----[ Revision History ]---------------------------------------------------------------

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

TX              PIN     5               ' serial output to LCD
Xaxis           PIN     6               ' data from accelerometer
CS              PIN     0               'ADC Chip Select
CLK             PIN     1               'ADC Clock
Datain1         PIN     2               '1st Alphabet
Datain2         PIN     3               '2nd Alphabet
Datain3         PIN     4               '3rd Alphabet
servoPoint      PIN     15              'Standard Servo
Correct         PIN     7

Enable555       PIN     9               'Reset of 555 timer

' -----[ Constants ]-----------------------------------------------------------------------

baud            CON     84               ' baudrate for the uart protocol
scalecon        CON     13316           ' scaling factor for accelerometer

' -----[ Variables ]-----------------------------------------------------------------------
xraw         VAR     Word
Xo           VAR     Word
ADC1         VAR     Byte
ADC2         VAR     Byte
ADC3         VAR     Byte
Alpha1       VAR     Nib
Alpha2       VAR     Nib
Alpha3       VAR     Nib
i            VAR     Byte
nightLamp    VAR     Bit

'peizo

index        VAR     Byte
offset       VAR     Nib

noteLetter   VAR     Byte
noteFreq     VAR     Word
noteDuration VAR     Word
noteOctave   VAR     Nib
noteDot      VAR     Bit

' -----[ EEPROM Data ]-----------------------------------------------------


Octaves    DATA      7,   6,   7,   6,   6,   6,   6,   6,   6,   6,   6,
            7,   6,   6,   7,   6,   6,   6,   6,   6,   6,   6,   7,   6,
            6,   7,   6,   6,   6,   6,   6,   6

Durations  DATA      2,   2,   2,   2,   2,   4,   2,   4,   2,   2,   4,   4,
            4,   4,   4,   4,   4,   4,   2,   4,   4,   4,   4,   4,   2

Dots       DATA      0,   0,   0,   0,   0,   0,   0,   0,   1,   0,   0,   0,
            0,   0,   0,   0,   0,   0,   0,   0,   1,   0,   0,   0,
            0,   0,   0,   0,   0,   0,   1

' -----[ Initialization ]--------------------------------------------------

setup:
    HIGH TX                       ' setup serial output pin
SEROUT TX, baud, [$19, $0C]  ' cursor on, clear display

'A
SEROUT TX, baud,[$F8,$00,$11,$11,$1F,$11,$11,$0A,$04]

'B
SEROUT TX, baud,[$F9,$00,$0F,$11,$11,$0F,$11,$11,$0F]

'C
SEROUT TX, baud,[$FA,$00,$0E,$11,$01,$01,$01,$11,$0E]

'D
SEROUT TX, baud,[$FB,$00,$0F,$11,$11,$11,$11,$11,$0F]

'E
SEROUT TX, baud,[$FC,$00,$1F,$01,$01,$1F,$01,$01,$1F]

'R
SEROUT TX, baud,[$FD,$00,$11,$09,$05,$0F,$11,$11,$0F]

'T
SEROUT TX, baud,[$FE,$00,$04,$04,$04,$04,$04,$04,$1F]

LOW Enable555

' ---- [ Program Code ]-----------------------------------------------
Main:

IF(IN8 = 1) THEN
  nightLamp = nightLamp ^ 1
ENDIF

IF(nightLamp = 1) THEN
  HIGH   Enable555
  DO UNTIL noteLetter = "Q"
    READ Notes + index, noteLetter
    LOOKUP offset, [ 4186, 4435, 4699, 4978, 5274, 5588, 5920, 6272, 6645, 7040, 7459, 7902, 0, 0 ], noteFreq
READ Octaves + index, noteOctave
noteOctave = 8 - noteOctave
noteFreq = noteFreq / (DCD noteOctave)

READ Durations + index, noteDuration
noteDuration = 1000 / noteDuration

READ Dots + index, noteDot
IF noteDot = 1 THEN noteDuration = noteDuration * 3 / 2

FREQOUT 10, noteDuration, noteFreq

index = index + 1
LOOP
ELSE
GOSUB Play
LOW Enable555
ENDIF

GOTO Main

' -----[ Subroutines ]-----------------------------------------------------
dispReverse:

HIGH CS
LOW CS
LOW CLK
PULSOUT CLK, 210
SHIFTIN Datain3,CLK,MSBPOST,[ADC3\8]
IF (ADC3<32) THEN
   SEROUT TX, baud,[\$0]  'A
   Alpha3 = 0
ELSEIF(ADC3<64) THEN
   SEROUT TX, baud,[\$1]  'B
   Alpha3 = 1
ELSEIF(ADC3<96) THEN
   SEROUT TX, baud,[\$2]  'C
   Alpha3 = 2
ELSEIF(ADC3<128) THEN
   SEROUT TX, baud,[\$3]  'D
   Alpha3 = 3
ELSEIF(ADC3<160) THEN
   SEROUT TX, baud,[\$4]  'E
   Alpha3 = 4
ELSEIF(ADC3<192) THEN
SEROUT TX, baud,[$4F] 'O
Alpha3 = 5
ELSEIF(ADC3<224) THEN
  SEROUT TX, baud,[$5]  'R
  Alpha3 = 6
ELSEIF(ADC3<256) THEN
  SEROUT TX, baud,[$6]  'T
  Alpha3 = 7
ENDIF

HIGH CS
LOW CS
LOW CLK
PULSOUT CLK, 210
SHIFTIN Datain2,CLK,MSBPOST,[ADC2]\8
IF (ADC2<32) THEN
  SEROUT TX, baud,[S0]  'A
  Alpha2 = 0
ELSEIF(ADC2<64) THEN
  SEROUT TX, baud,[S1]  'B
  Alpha2 = 1
ELSEIF(ADC2<96) THEN
  SEROUT TX, baud,[S2]  'C
  Alpha2 = 2
ELSEIF(ADC2<128) THEN
  SEROUT TX, baud,[S3]  'D
  Alpha2 = 3
ELSEIF(ADC2<160) THEN
  SEROUT TX, baud,[S4]  'E
  Alpha2 = 4
ELSEIF(ADC2<192) THEN
  SEROUT TX, baud,[S4F] 'O
  Alpha2 = 5
ELSEIF(ADC2<224) THEN
  SEROUT TX, baud,[S5]  'R
  Alpha2 = 6
ELSEIF(ADC2<256) THEN
  SEROUT TX, baud,[S6]  'T
  Alpha2 = 7
ENDIF

HIGH CS
LOW CS
LOW CLK
PULSOUT CLK, 210
SHIFTIN Datain1,CLK,MSBPOST,[ADC1]\8
IF (ADC1<32) THEN
SEROUT TX, baud,[$0] 'A
Alpha1 = 0
ELSEIF(ADC1<64) THEN
    SEROUT TX, baud,[$1] 'B
    Alpha1 = 1
ELSEIF(ADC1<96) THEN
    SEROUT TX, baud,[$2] 'C
    Alpha1 = 2
ELSEIF(ADC1<128) THEN
    SEROUT TX, baud,[$3] 'D
    Alpha1 = 3
ELSEIF(ADC1<160) THEN
    SEROUT TX, baud,[$4] 'E
    Alpha1 = 4
ELSEIF(ADC1<192) THEN
    SEROUT TX, baud,[$4F] 'O
    Alpha1 = 5
ELSEIF(ADC1<224) THEN
    SEROUT TX, baud,[$5] 'R
    Alpha1 = 6
ELSEIF(ADC1<256) THEN
    SEROUT TX, baud,[$6] 'T
    Alpha1 = 7
ENDIF
RETURN

dispStraight:

HIGH CS
LOW CS
LOW CLK
PULSOUT CLK, 210
SHIFTIN Datain1,CLK,MSBPOST,[ADC1\8]
IF (ADC1<32) THEN
    SEROUT TX, baud,[$41] 'A
    Alpha1 = 0
ELSEIF(ADC1<64) THEN
    SEROUT TX, baud,[$42] 'B
    Alpha1 = 1
ELSEIF(ADC1<96) THEN
    SEROUT TX, baud,[$43] 'C
    Alpha1 = 2
ELSEIF(ADC1<128) THEN
    SEROUT TX, baud,[$44] 'D
    Alpha1 = 3
ELSEIF(ADC1<160) THEN  
    SEROUT TX, baud,[\$45] 'E  
    Alpha1 = 4
ELSEIF(ADC1<192) THEN  
    SEROUT TX, baud,[\$4F] 'O  
    Alpha1 = 5
ELSEIF(ADC1<224) THEN  
    SEROUT TX, baud,[\$52] 'R  
    Alpha1 = 6
ELSEIF(ADC1<256) THEN  
    SEROUT TX, baud,[\$54] 'T  
    Alpha1 = 7
ENDIF

HIGH CS
LOW CS
LOW CLK
PULSOUT CLK, 210
SHIFTIN Datain2,CLK,MSBPOST,[ADC2\8]
IF (ADC2<32) THEN  
    SEROUT TX, baud,[\$41] 'A  
    Alpha2 = 0
ELSEIF(ADC2<64) THEN  
    SEROUT TX, baud,[\$42] 'B  
    Alpha2 = 1
ELSEIF(ADC2<96) THEN  
    SEROUT TX, baud,[\$43] 'C  
    Alpha2 = 2
ELSEIF(ADC2<128) THEN  
    SEROUT TX, baud,[\$44] 'D  
    Alpha2 = 3
ELSEIF(ADC2<160) THEN  
    SEROUT TX, baud,[\$45] 'E  
    Alpha2 = 4
ELSEIF(ADC2<192) THEN  
    SEROUT TX, baud,[\$4F] 'O  
    Alpha2 = 5
ELSEIF(ADC2<224) THEN  
    SEROUT TX, baud,[\$52] 'R  
    Alpha2 = 6
ELSEIF(ADC2<256) THEN  
    SEROUT TX, baud,[\$54] 'T  
    Alpha2 = 7
ENDIF

HIGH CS
LOW CS
LOW CLK
PULSOUT CLK, 210
SHIFTIN Datain3,CLK,MSBPOST,[ADC3\8]
IF (ADC3<32) THEN
    SEROUT TX, baud,[${41}]  'A
    Alpha3 = 0
ELSEIF(ADC3<64) THEN
    SEROUT TX, baud,[${42}]  'B
    Alpha3 = 1
ELSEIF(ADC3<96) THEN
    SEROUT TX, baud,[${43}]  'C
    Alpha3 = 2
ELSEIF(ADC3<128) THEN
    SEROUT TX, baud,[${44}]  'D
    Alpha3 = 3
ELSEIF(ADC3<160) THEN
    SEROUT TX, baud,[${45}]  'E
    Alpha3 = 4
ELSEIF(ADC3<192) THEN
    SEROUT TX, baud,[${4F}]  'O
    Alpha3 = 5
ELSEIF(ADC3<224) THEN
    SEROUT TX, baud,[${52}]  'R
    Alpha3 = 6
ELSEIF(ADC3<256) THEN
    SEROUT TX, baud,[${54}]  'T
    Alpha3 = 7
ENDIF
RETURN

gotoCOD:
    FOR i = 1 TO 100
        PULSOUT servoPoint, 180
        PAUSE 20
    NEXT
RETURN

gotoCAT:
    FOR i = 1 TO 100
        PULSOUT servoPoint, 360
        PAUSE 20
    NEXT
RETURN

gotoBAT:
    FOR i = 1 TO 100
PULSOUT servoPoint, 540
PAUSE 20
NEXT
RETURN
gotoDOE:
FOR i = 1 TO 100
    PULSOUT servoPoint, 720
    PAUSE 20
NEXT
RETURN
gotoRAT:
FOR i = 1 TO 100
    PULSOUT servoPoint, 900
    PAUSE 20
NEXT
RETURN
gotoBEE:
FOR i = 1 TO 100
    PULSOUT servoPoint, 1080
    PAUSE 20
NEXT
RETURN
compareWord:
'COD
IF((Alpha1=2) AND (Alpha2=5) AND (Alpha3=3) )THEN
    GOSUB gotoCOD
    HIGH Correct

'CAT
ELSEIF((Alpha1=2) AND (Alpha2=0) AND (Alpha3=7)) THEN
    GOSUB gotoCAT
    HIGH Correct

'BAT
ELSEIF((Alpha1=1) AND (Alpha2=0) AND (Alpha3=7)) THEN
    GOSUB gotoBAT
    HIGH Correct

'DOE
ELSEIF((Alpha1=3) AND (Alpha2=5) AND (Alpha3=4)) THEN
    GOSUB gotoDOE
'RAT
ELSEIF((Alpha1=6) AND (Alpha2=0) AND (Alpha3=7)) THEN
  GOSUB gotoRAT
  HIGH Correct

'BEE
ELSEIF((Alpha1=1) AND (Alpha2=4) AND (Alpha3=4)) THEN
  GOSUB gotoBEE
  HIGH Correct

'NONE
ELSE
  LOW Correct
ENDIF
ENDIF

RETURN

Play:

' your code here
PULSIN Xaxis, 1, xraw
Xo=xraw**scalecon-508
DEBUG HOME, SDEC Xo , CR, SDEC xraw

IF (Xo.HIGHBIT = 1) THEN
  SEROUT TX, baud,[0C,9A]
  'SEROUT TX, baud,[6,0,2,1,3,4,5]
  GOSUB dispReverse
ELSE
  SEROUT TX, baud,[0C,87]
  GOSUB dispStraight
  'SEROUT TX, baud,[43,41,54,52]
ENDIF

GOSUB compareWord

RETURN
5. Electronic Circuits

ADC:

![ADC Circuit Diagram]

Servo Motor:

![Servo Motor Diagram]
6. Bill-of-material

<table>
<thead>
<tr>
<th>Component</th>
<th>Cost/Piece in USD</th>
<th>Quantity</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Stamp 2</td>
<td>99.95</td>
<td>1</td>
<td>99.95</td>
</tr>
<tr>
<td>Potentiometer</td>
<td>0.3</td>
<td>3</td>
<td>0.9</td>
</tr>
<tr>
<td>ADC</td>
<td>1.3</td>
<td>3</td>
<td>3.9</td>
</tr>
<tr>
<td>Standard servo</td>
<td>15</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>Bicolor LED</td>
<td>1.5</td>
<td>1</td>
<td>1.5</td>
</tr>
<tr>
<td>555timer</td>
<td>0.5</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>LCD display</td>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Accelerometer</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Resistor pack</td>
<td>10</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>Wire spool</td>
<td>5</td>
<td>3</td>
<td>15</td>
</tr>
<tr>
<td>Battery</td>
<td>1.2</td>
<td>1</td>
<td>1.2</td>
</tr>
<tr>
<td><strong>Total cost</strong></td>
<td><strong>$142.75</strong></td>
<td></td>
<td><strong>$155.95</strong></td>
</tr>
</tbody>
</table>
7. Improvements

Using all the alphabets, in the English Language.
Increasing the number of words that can be learnt.
Making the system more robust.
Coding can be optimized for better efficiency.
Use of EEPROM to save memory.
Displaying pictures on the LCD.

8. Future Development

ABC: Automatic Barrier Constructor