```
// *** fslbot demo ***
// touch mouth to blink leds
// adjust potentiometer for walk speed
// touch left/right cheek to take 6 steps leading from left/right
// touch forehead to stop march
//
dim mpr121
mpr121 = 0x5a  // i2c address
dim ioex
ioex = 0x38  // i2c address
//
dim blink, lit  // state of the mouth LEDs
dim right, left  // counts of steps to take
//
// *** initialize our modules ***
gosub mpr121_init
gosub mouth_init
//
// *** configure the mpr121_isr ***
dim isr as pin irq7* for digital input inverted
on isr do gosub mpr121_isr
//
// *** configure the mouth_timer ***
configure timer 1 for 300 ms
on timer 1 do gosub mouth_timer
//
// *** configure the servo pins ***
dim rfoot as pin dtin0 for servo output
dim rhip as pin dtin1 for servo output
dim lhip as pin dtin2 for servo output
dim lfoot as pin dtin3 for servo output
//
// *** configure the walk speed potentiometer
dim pot as pin an6 for analog input
//
// *** set the walk stride variables ***
dim delay
dim flower, fstart, fraise, ftip, hshift
delay = 15
flower = 600, fstart = 150, fraise = 230, ftip = 50, hshift = 250
//
// *** stand square ***
gosub stand
//
// *** main loop -- just walk ***
//
while 1 do
  if left>right&&left>0 then
    // lead with the left foot
    gosub left_step
    left = left-2
  elseif right>left&&right>0 then
    // lead with the right foot
    gosub right_step
    right = right-2
  else
    gosub stand
  endif
  delay = pot*15/1650
endwhile
end
```

```
//
// *** walking subroutines ***
//
sub right_step
  // step the right foot
  lfoot = 0  // relax left foot
  for rfoot = rfoot to 1500+flower step 10  // lean left
    sleep delay ms
  next
  for lfoot = 1500+fstart to 1500+fraise step 10  // stand left
    sleep delay ms
  next
  do   // take right step
    if rfoot>1500+ftip then
      rfoot = rfoot-10
    endif
    if rhip>1500-hshift then
      rhip = rhip-10
      lhip = rhip
    endif
    sleep delay ms
  until rfoot<=1500+fstart&&rhip<=1500-hshift
  for lfoot = lfoot to 1500 step -10  // fall back right
    sleep delay ms
  next
endsub
//
sub left_step
  // step the left foot
  rfoot = 0  // relax right foot
  for lfoot = lfoot to 1500-flower step -10  // lean right
    sleep delay ms
  next
  for rfoot = 1500-fstart to 1500-fraise step -10  // stand right
    sleep delay ms
  next
  do   // take left step
    if lfoot<1500-ftip then
      lfoot = lfoot+10
    endif
    if lhip<1500+hshift then
      lhip = lhip+10
      rhip = lhip
    endif
    sleep delay ms
  until lfoot>=1500-fstart&&lhip>=1500+hshift
  for rfoot = rfoot to 1500 step 10  // fall back left
    sleep delay ms
  next
endsub
//
sub stand
  // stand square
  rfoot = 1500, rhip = 1500, lhip = 1500, lfoot = 1500
endsub
//
// *** mouth subroutines ***
//
sub mouth_timer
  dim r as byte, d as byte
  // blink all bits if we're supposed to
```

```
  if blink then
    lit = ~lit
    r = 1, d = lit
    i2c start ioex
    i2c write r, d
    i2c stop
  endif
endsub
//
sub mouth_init
  dim r as byte, d as byte
  // configure bits for output
  r = 3, d = 0
  i2c start ioex
  i2c write r, d
  i2c stop
endsub
//
// *** touch subroutines ***
//
sub mpr121_isr
  dim bits
  // get the touch value and respond appropriately
  gosub mpr121_poll bits
  if bits&128 then
    // print "mouth"
    blink = !blink
  endif
  if bits&64 then
    // print "forehead"
    left = 0, right = 0
  endif
  if bits&32 then
    // print "left cheek"
    left = 6, right = 5
  endif
  if bits&16 then
    // print "right cheek"
    right = 6, left = 5
  endif
endsub
//
sub mpr121_poll bits
  dim r as byte, r0 as byte, r1 as byte
  // read and return both bytes of the touch register
  i2c start mpr121
  r = 0
  i2c write r
  i2c read r0
  r = 1
  i2c write r
  i2c read r1
  i2c stop
  bits = r1<<8|r0
endsub
//
sub mpr121_init
  dim i
  dim r as byte, d as byte
  // just follow AN3944: MPR121 Quick Start Guide
  i2c start mpr121
```

```
  for i = 1 to 0x17 step 2
    r = 0x40+i, d = 0xf
    i2c write r, d
    r = 0x41+i, d = 0xa
    i2c write r, d
  next
  restore mpr121
  do
    read r, d
    i2c write r, d
  until r==0x5e
  i2c stop
  label mpr121
  data 0x2b, 0x1, 0x2c, 0x1, 0x2d, 0x0, 0x2e, 0x0
  data 0x2f, 0x1, 0x30, 0x1, 0x31, 0xff, 0x32, 0x2
  data 0x5d, 0x4, 0x7b, 0xb, 0x7d, 0x9c, 0x7e, 0x65
  data 0x7f, 0x8c, 0x5e, 0x8c
endsub
```