Step 1 – Required Utilities

```
sudo apt-get install git
```

pip – install this with: `sudo apt-get install python-pip`

Setting Python Wrapper

```
cd ~
git clone git://github.com/doceme/py-spidev
cd py-spidev/
sudo python setup.py install
```

Reading a value from the sensor

```
Python 2.7.3 (default, Jan 13 2013, 11:20:46)
[GCC 4.6.3] on linux2
>>> import mcp3008
>>> mcp3008.readadc(5)
444
```

Monitoring the Console

```
from time import sleep
import mcp3008

while True:
  m = mcp3008.readadc(5)
  print "Moisture level: {:>5} ".format(m)
  sleep(.5)
```

The output should be

```
Moisture level:   452
Moisture level:   486
Moisture level:   485
Moisture level:   483
Moisture level:   489
Moisture level:   491
Moisture level:   490
```

For console Monitoring the Land

```
from time import sleep
import mcp3008

# ANSI escape codes
PREVIOUS_LINE="\x1b[1F"
RED_BACK="\x1b[41;37m"
GREEN_BACK="\x1b[42;30m"
YELLOW_BACK="\x1b[43;30m"
RESET="\x1b[0m"

# Clear the screen and put the cursor at the top
print '\x1b[2J\x1b[H'
print 'Moisture sensor'
print '===============\n'

while True:
    m = mcp3008.readadc(5)
    if m < 150:
        background = RED_BACK
    elif m < 500:
        background = YELLOW_BACK
    else:
        background = GREEN_BACK
    print PREVIOUS_LINE + background + "Moisture level: {:>5} ".format(m) + RESET
    sleep(.5)
```

Program

```
#!/usr/bin/python

# Monitor two soil sensors on MCP3008, ch 2 and 3
# (pin 3 and 4)

import spidev
import time
import os

# Open SPI bus
spi = spidev.SpiDev()
spi.open(0,0)

# Function to read SPI data from MCP3008 chip
def ReadChannel(channel):
        adc = spi.xfer2([1,(8+channel)<<4,0])
        data = ((adc[1]&3) << 8) + adc[2]
        return data


# Main loop - read raw data and display
while True:
        soilOne = ReadChannel(2)
        soilTwo = ReadChannel(3)

        # Output
        print "Soil1=",soilOne," : Soil2=",soilTwo

        time.sleep(0.5)
```

LCD Monitoring

```
import sys
sys.path.append('/home/pi/py/Adafruit-Raspberry-Pi-Python-
Code/Adafruit_CharLCDPlate')

from time import sleep
from Adafruit_CharLCDPlate import Adafruit_CharLCDPlate
import mcp3008

lcd = Adafruit_CharLCDPlate()

while True:
    m = mcp3008.readadc(5)
    try:
        lcd.home()
        lcd.message("Moisture level:\n%d     " % m)
        if m < 150:
            lcd.backlight(lcd.RED)
        elif m < 500:
            lcd.backlight(lcd.YELLOW)
        else:
            lcd.backlight(lcd.GREEN)
        except IOError as e:
            print e
        sleep(.5)
```

Run program like this

```
pi@raspberrypi ~/py/tutorials/moisture $ sudo python moist_lcd.py
```

To add a line chart for the moisture control

```
from time import sleep
import mcp3008
from controlmypi import ControlMyPi
import logging
import datetime

def on_msg(conn, key, value):
    pass

def append_chart_point(chart, point):
    if len(chart) >= 10:
        del chart[0]
    chart.append(point)
    return chart

logging.basicConfig(level=logging.INFO)

p = [
    [ ['G','moist','% level',0,0,100], ['LC','chart1','Time','Value',0,100]
],
    ]

c1 = []

conn = ControlMyPi('you@gmail.com', 'password', 'moistcmp2', 'Moisture
monitor 2', p, on_msg)
if conn.start_control():
    try:
        while True:
            dt = datetime.datetime.now().strftime('%H:%M:%S')
            m = mcp3008.read_pct(5)
            c1 = append_chart_point(c1, [dt, m])
            conn.update_status({'moist':m,'chart1':c1})
            sleep(30)
    finally:
        conn.stop_control()
```

Final Program

```
import sys
sys.path.append('/home/pi/py/Adafruit-Raspberry-Pi-Python-
Code/Adafruit_CharLCDPlate')

from time import sleep
from Adafruit_CharLCDPlate import Adafruit_CharLCDPlate
import mcp3008
from controlmypi import ControlMyPi
import logging
import datetime
import pickle
from genericpath import exists
import smtplib

lcd = Adafruit_CharLCDPlate()

PICKLE_FILE = '/home/pi/py/moisture/moist.pkl'

def on_msg(conn, key, value):
    pass

def append_chart_point(chart, point):
    if len(chart) >= 48:
        del chart[0]
    chart.append(point)
    return chart

def save(data):
    output = open(PICKLE_FILE, 'wb')
    pickle.dump(data, output)
    output.close()

def load(default):
    if not exists(PICKLE_FILE):
        return default
    pkl_file = open(PICKLE_FILE, 'rb')
    data = pickle.load(pkl_file)
    pkl_file.close()
    return data

def update_lcd(m):
    try:
        lcd.home()
        lcd.message("Moisture level:\n%d%%    " % m)
        if m < 15:
            lcd.backlight(lcd.RED)
        elif m < 50:
            lcd.backlight(lcd.YELLOW)
        else:
            lcd.backlight(lcd.GREEN)
    except IOError as e:
        print e

def send_gmail(from_name, sender, password, recipient, subject, body):
        senddate=datetime.datetime.strftime(datetime.datetime.now(), '%Y-%m-
%d')
    msg="Date: %s\r\nFrom: %s <%s>\r\nTo: %s\r\nSubject: %s\r\nX-Mailer: My-
Mail\r\n\r\n" % (senddate, from_name, sender, recipient, subject)
    server = smtplib.SMTP('smtp.gmail.com:587')
    server.starttls()
    server.login(sender, password)
    server.sendmail(sender, recipient, msg+body)
    server.quit()

logging.basicConfig(level=logging.INFO)

p = [
    [ ['G','moist','level',0,0,100], ['LC','chart1','Time','Value',0,100] ],
    ]

c1 = load([])

readings = []

conn = ControlMyPi('you@gmail.com', 'password', 'moisture', 'Moisture
monitor', p, on_msg)

delta = datetime.timedelta(minutes=30)
next_time = datetime.datetime.now()

delta_email = datetime.timedelta(days=1)
next_email_time = datetime.datetime.now()

if conn.start_control():
    try:
        while True:
            dt = datetime.datetime.now()
            m = mcp3008.read_pct(5)
            readings.append(m)
            update_lcd(m)
            to_update = {'moist':m}

            # Update the chart?
            if dt > next_time:
                # Take the average from the readings list to smooth the graph
a little
                avg = int(round(sum(readings)/len(readings)))
                readings = []
                c1 = append_chart_point(c1, [dt.strftime('%H:%M'), avg])
                save(c1)
                next_time = dt + delta
                to_update['chart1'] = c1
            conn.update_status(to_update)

            #Send an email?
            if dt > next_email_time:
                next_email_time = dt + delta_email
                if m < 40:
                    send_gmail('Your Name', 'you@gmail.com', 'password',
'recipient@email.com', 'Moisture sensor level', 'The level is now: %s' % m)
```

sleep(30)