

```

# SET CYCLE PROGRAM
#
import serial
import datetime
print ""
LLAP UTILITY TO SEND CISECO XRF MODULES INTO SLEEP CYCLE,-
ONLY TESTED FOR THERMISTORS!
See:
http://openmicros.org/index.php/articles/87-llap-lightweight-local-automationprotocol/llap-devices-commands-and-instructions/122-xrf-firmware-temp
for details (above address should cut and paste).
NOTE! This program will simply hang if used to set an XRF module that
is switched off.
Use 'ctrl+z' to exit if this happens.
This python file is written to set the XRF labelled "--" to cycle.
This is the
factory default. If the selected device has been given a new ID (by
sending
"a--CHDEVIDXX", where "XX" is the new name and then sending "a--
REBOOT---"
to save it), the variable "devid" in this file will need to be edited.
The sleep interval is set by a 3 digit code (001 to 999), followed by
D (days),
H (hours), M (minutes) or S (guess what?). The default is 15 minutes =
015M
Once the settings commands are sent, this program sits and listens,
displaying
the time of each reading received. This allows you to verify that the
new
cycle is being followed by the XRF.
""
#ID of devid to listen to
devid = '--'
baud = 9600
port = '/dev/ttyAMA0'
interval = "001M"
# reset the original 030M to 001M above...
# 2 readings an hour is enough to allow one to drop out and still get
an hourly update
now = datetime.datetime.now()
print "Computer reports time as " + now.strftime("%H:%M %m-%d-%Y")
print "Set calibration to " + interval
print "Use use ctrl+z to exit and then edit this file if you want to
change this."
go = raw_input('Press return key to commit to this setting.')
print "Opening port "+ port + " at ", baud, " baud."
ser = serial.Serial(port, baud)
# Clear out anything in the read buffer
ser.flushInput()

```

```
# if this XRF has never been put into cycle before, this is all you
need :
ser.write('a'+devid+'WAKE-----')
ser.write('a'+devid+'INTVL'+interval)
ser.write('a'+devid+'CYCLE----')
# See if it worked...
print "Monitoring cycle timing and waiting for battery reading at " +
now.strftime("%H:%M %m-%d-%Y")
print "If interval is correct, use ctrl+z to exit"
while 1 :
# wait for message, if it's a battery reading we've go just 100 ms to
send the WAKE call,
# so put all the checking the time and printing of details AFTER the
reading/writing
# and make the if statement just one character long!
if ser.inWaiting() > 0 :
llapMsg = ser.read(ser.inWaiting())
if "B" in llapMsg :
ser.write('a'+devid+'WAKE-----')
ser.write('a'+devid+'INTVL'+interval)
ser.write('a'+devid+'CYCLE----')
print ' '
print '!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!'
print '!Calibration signal sent!'
print '!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!'
print ' '
print 'Interval should now change to ' + interval
print ' '
# Get time from system
now = datetime.datetime.now()
print 'Content = |' + llapMsg + "| detected at "+ now.strftime("%H:%M
%m-%d-
%Y")
```