

```
#!/usr/bin/python

import socket
import sys
import time
import sqlite3

# Connect to ECoS
try:
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
except socket.error as err:
    print ("socket creation failed with error %s" %(err))
    sys.exit()

try:
    host_ip = socket.gethostbyname('ecos-0008fc50')
except socket.gaierror:
    print ("there was an error resolving the host")
    sys.exit()

port = 15471

s.connect((host_ip, port))

print ("The socket has successfully connected to ECoS")
print ("IP address:", host_ip)
print ("Port:", port)

# Detect Feedback modules (id=26)
# 100-199 = S88
```

```
# 200 - 300 = Railcom
```

```
message = bytes('queryObjects(26, ports)', 'utf-8')
```

```
try:
```

```
    s.sendall(message)
```

```
except socket.error:
```

```
    print ('send failed')
```

```
    sys.exit()
```

```
reply = s.recv(1024)
```

```
print ('Following detector modules found:')
```

```
print (reply.decode('utf-8'))
```

```
# Connect to ECoSdata.db
```

```
dbconnect = sqlite3.connect("ECoSdata.db")
```

```
dbconnect.row_factory = sqlite3.Row;
```

```
dbconnect.text_factory = str;
```

```
cursor = dbconnect.cursor();
```

```
#message = bytes('get(200, state)', 'utf-8')
```

```
#s.sendall(message)
```

```
#reply = s.recv(1024)
```

```
#print (reply.decode('utf-8'))
```

```
global loc_id
```

```
loc_id=""
```

```
def monitor_modules(mod_no, mod_ports):
```

```
    strmsg = 'get('+mod_no+', railcom['+mod_ports+'])'
```

```

message = bytes(strmsg, 'utf-8')
s.sendall(message)
reply = s.recv(1024)
replystr = reply.decode('utf-8')
rcport = replystr.split(' ')[2].split(',')[0]
print ("Port: ", rcport)
locaddr = replystr.split(',')[2].split(',')[0]
print ("Loc addr: ", locaddr)
direction = replystr.split(' ')[2].split(',')[2].split(' ')[0]
print ("Direction: ", direction)
locdb_lookup(locaddr)

```

```

def locdb_lookup(ecos_id):
    print ("locdb_lookup called - ecosid = ", ecos_id)
    cursor.execute('SELECT * from Locotable where addr = '+ecos_id)
    for row in cursor:
        print(row['locid'],row['name'],row['addr'], row['protocol'] );
        loc_id = (row['locid']);
        print ("LocID = ", loc_id);
        #set_loc_speed(loc_id);
        set_fkey(loc_id);

```

```

def set_loc_speed(loc_id):
    speedmsg = 'set('+loc_id+', speed[20])'
    print ("Speedmsg = ", speedmsg)
    message = bytes(speedmsg, 'utf-8')
    s.sendall(message)
    reply = s.recv(1024)
    print ("Reply from Speedmsg: ", reply)

```

```

def set_fkey(loc_id):
    message = bytes('get('+loc_id+', speed)', 'utf-8')
    s.sendall(message)
    reply = s.recv(1024)
    print ("Reply from get speed: ", reply)

    message = bytes('get('+loc_id+', speedstep)', 'utf-8')
    s.sendall(message)
    reply = s.recv(1024)
    print ("Reply from get speedstep: ", reply)

    message = bytes('get('+loc_id+', dir)', 'utf-8')
    s.sendall(message)
    reply = s.recv(1024)
    print ("Reply from get direction: ", reply)

    message = bytes('get('+loc_id+', func[1])', 'utf-8')
    s.sendall(message)
    reply = s.recv(1024)
    print ("Reply from get func[1]: ", reply)

    message = bytes('get('+loc_id+', funcdesc[1])', 'utf-8')
    s.sendall(message)
    reply = s.recv(1024)
    print ("Reply from get funcdesc[1]: ", reply)

    message = bytes('get('+loc_id+', funcexists[1])', 'utf-8')
    s.sendall(message)
    reply = s.recv(1024)
    print ("Reply from get funcexists[1]: ", reply)

```

```
message = bytes('get('+loc_id+', locodesc)', 'utf-8')
```

```
s.sendall(message)
```

```
reply = s.recv(1024)
```

```
print ("reply from get locodesc: ", reply)
```

```
if __name__ == "__main__":
```

```
    monitor_modules('200', '1');
```

```
    dbconnect.close();
```

```
sys.exit();
```