

```
#!/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);

        read_fkey(loc_id);

        #activate_fkey(loc_id);

```

```

def read_fkey(loc_id):

    i=0

    while i <= 32:

        i_str = str(i)

        print ("i_str is: "+i_str)

```

```

message = (bytes('get('+loc_id+', funcicon['+i_str+']), 'utf-8'))
#message = (bytes('get('+loc_id+', funcexists['+i_str+']), 'utf-8'))
s.sendall(message)
reply = s.recv(1024)
#print ("Reply from funcdesc["+i_str+"] is: ", reply)
replystr = reply.decode('utf-8')
#print (replystr)
func = replystr.split('\n')[1].split(',')[1].split(' ')[0]
print ('fkey['+i_str+] is: ', func)
i = i+1

```

```

def activate_fkey(loc_id):
    message = (bytes('set('+loc_id+', func[17,1]), 'utf-8'))
    s.sendall(message)
    reply = s.recv(1024)
    print ("Reply from set func[17,1] is: ", reply)

```

```

if __name__ == "__main__":
    monitor_modules('200', '0');
    dbconnect.close();
sys.exit();

```