

How to write a SocExplorer plugin

Once you have installed SocExplorer from sources on your system, it should have added an new template to QtCreator. You can create a new plugin skeleton filling the wizard forms. Your plugin description is written in its qmake project file. A SocExplorer plugin inherits from the socexplorerplugin base class which inherits from QDockWidget.

SocExplorerEngine services

Through SocExplorerEngine object you get some shared and centralized services.

Message logging

Instead of using printf inside a plugin to print some information, you can use this function:

```
void SocExplorerEngine::message(socexplorerplugin *sender, const QString &message, int debugLevel)

//From your plugin:

SocExplorerEngine::message(this, "Here is a message with a debug level of 2",2);
```

With this function you will centralize all the plugins messages and they will share the same formalism. Note that you can set the debug level of your message, if the message debugLevel is lower than SocEplorer loglevel then the message will be printed else it will be dropped.

To set the SocExplorer debug level you can use the -d option like this:

```
socexplorer -d 4 #all the message with a debug level or equal lower than 4 will be printed
```

Managing peripherals base address (Plug'n'Play)

You don't necessary know the address of the peripherals present in your soc when you write a plugin for SocExplorer. So SocEplorer allows you to enumerate the peripheral inside your soc at run-time and share their base address with other plugins.

To enumerate a device/peripheral:

```
int addEnumDevice(const QString& rootPlugin,int VID,int PID,qint32 baseAddress,const QString& name
);

//From your plugin:
QString devname = SocExplorerEngine::getDevName(VID,PID);
// If the device is known you will get it name
SocExplorerEngine::addEnumDevice(this,VID,PID,BaseAddress,devname);
```

Now to get a peripheral base address:

```
qint32 getEnumDeviceBaseAddress(const QString& rootPlugin,int VID,int PID,int count=0);
//or
qint32 getEnumDeviceBaseAddress(socexplorerplugin* plugin,int VID,int PID,int count=0);

//From your plugin:
unsigned int DSUBASEADDRESS = SocExplorerEngine::self()->getEnumDeviceBaseAddress(this,0x01 ,
0x004,0); //you will get the Grlib's DSU3 base address
```

Loading a plugin from an other one

For example once you get successfully connected to the target through a root plugin you may want to automatically trigger an AMBA bus scan with the AMB plugin.

```
void loadChildSysDriver(socexplorerplugin *parent, const QString child);
```

```
//From your plugin:
```

```
socexplorerproxy::loadChildSysDriver(this, "AMBA_PLUGIN");
```