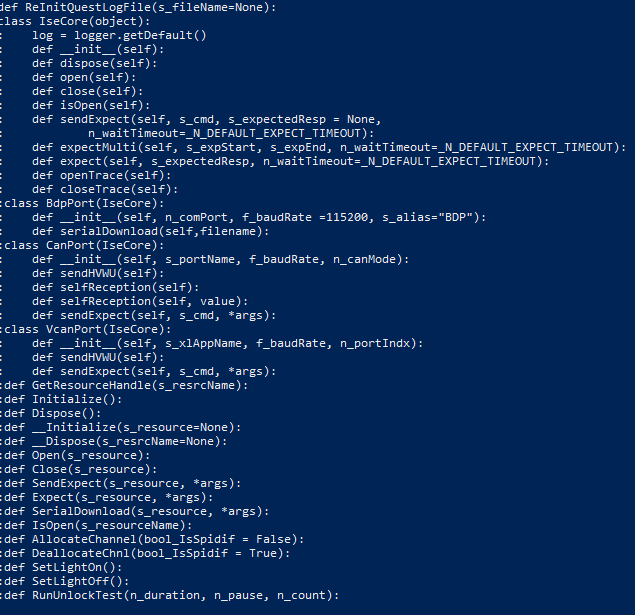
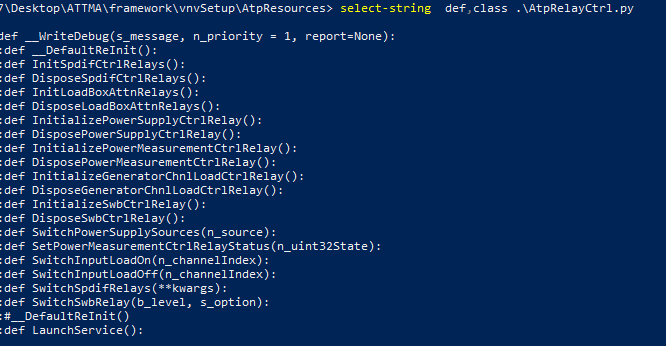
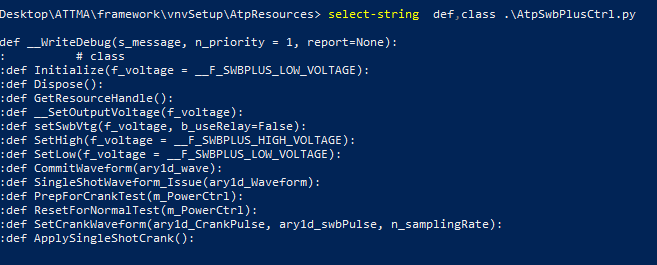
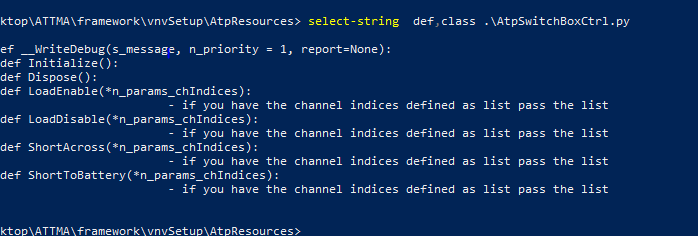
|  |  |
| --- | --- |
| **ATP** | importModules |
| **Ise (IseResources)** |
| AtpSwbPlusCtrl |
| AtpRelayCtrl |
| AtpPowerCtrl |
| **VehMaster**  **(A Car having multiple CAN nodes)** | VehMaster |
| AsmCtrlSim |
| SinkCtrlSim |
| **AudioControls** | AudioControls |
| **AudioPlayers** | AudioPlayer |
| Streoplayer |
| Monoplayer |
| MultiChnlPlayer |
| MicInPlayer |
| Verification Object | Listner |
|  |  |

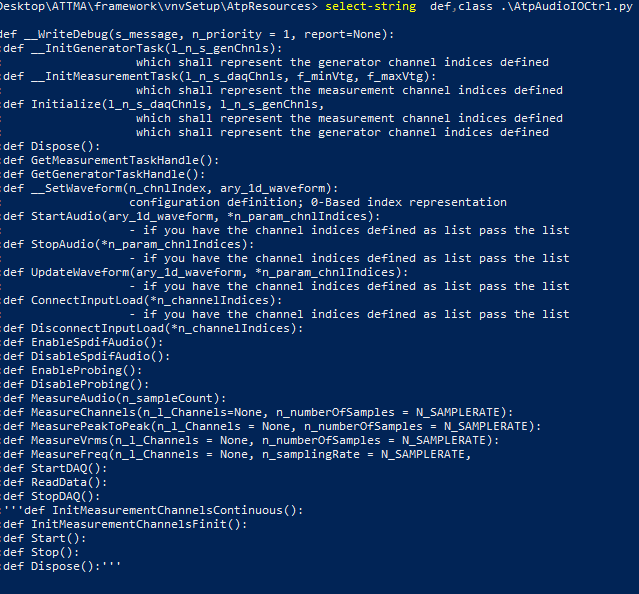
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ect | Purpose | Conf Dependency | Module dependecy | location |
| importModules | 1. Identify type of testing i,e Bench Or ATP. 2. Create and importable modules like   **IseResources**, **AtpPowerCtrl, AtpSwbPlusCtrl, AtpAudioIOCtrl**  **AtpMiscCtrl, AtpSwitchBoxCtrl**  **based on slection of hardware** | VnVConfig\setupCfg\setupSelection.json | PathManager  which is available in lib/PathManager.py | framework/vnvSetup/\_\_init\_\_.py |
| **IseResources.Initialize() Or Ise Object** | 1. Read atpConfig json, having different port config for ATP like BAUD rate , bus type. 2. Create IseCore object which has methods to perform sendExpect operation and also Traceses 3. IseCore is parent Class and will be used by other classes 4. Create BDP port class object which is inherited with IseCore having basic function. It configures and Initializes the   BDP port.   1. Create CAN port class object which is inherited with IseCore having basic function. It configures and Initializes the   CAN port.   1. Create vCanPort port class object which is inherited with IseCore having basic function. It configures and Initializes the   vCanPort port. | VnVConfig/setupCfg/atpCfg.json | ReferenceManager  DotQuest libs | framework/vnvSetup/CustomResources/IseResources.py |
| AtpSwbPlusCtrl | 1. Use Dot Quest Function generator to generate signal 2. Send signal to Sw+ (Battery) on PXI cards with specific channel as per config. 3. Setup output voltage. 4. Initialize Atp Swb+ relay using AtpRelay 5. Provides access methods to set power supply, power measurement and voltage management using Relays. | framework/vnvSetup/setupCfgs/AtpResourceConfig.py | PathManager  ReferenceManager  DotQuest libs | framework/vnvSetup/AtpResources/AtpSwbPlusCtrl.py |
| AtpRelayCtrl | Provides methods to initialize and control AtpRelays | framework/vnvSetup/setupCfgs/AtpResourceConfig.py | PathManager  ReferenceManager  DotQuest libs | framework/vnvSetup/AtpResources/AtpRelayCtrl.py |
| AtpPowerCtrl | Provides methods manage the power to various sources of Atp . | framework/vnvSetup/setupCfgs/AtpResourceConfig.py | PathManager  ReferenceManager  DotQuest libs | framework/vnvSetup/AtpResources/AtpPowerCtrl.py |
| vehMaster | 1. Use config to understand different methods of waking (by swb Or Can) up the Amp. 2. Create object which provides the access methods to invoke Amp. 3. Use config to understand different HMI CAN nodes attached. 4. Create and initialize the Can and mCan nodes by referencing json config. 5. Build can messages each Can and mCAn nodes to startup | VnVConfig/ampCfg/wakeupCfg.json  VnVConfig/CANDBC\_L21B\_GEN6.py | Tools/canNodes/\_canNode.py  Tools/CAN/CANStructs.py  Tools/CAN/PeriodicCANMessage.py  Tools/CAN/CANDBC\_DICT\_CONSTS.py | Tools/ctrlSims/\_vehMaster.py |
| AtpAudioIOCtrl | 1. Initializes the NI channels which generate signals and represent   AMP i/p channels   1. Initializes the NI channels which measure the signals and represent   AMP i/p channels   1. Provides different access methods to perform audio (signal) operation management like,   Start audio, stop audio, measure audio,  Measure channels | framework/vnvSetup/setupCfgs/AtpResourceConfig.py | PathManager  ReferenceManager  DotQuest libs | framework/vnvSetup/AtpResources/AtpAudioIOCtrl.py |
| AudioPlayer | 1. It is abstract class used as parent class creating other sound players for testing. 2. Configures sound wave to particular input channels. | VnVConfig/ampCfg/ioCfg.json | vnvSetup.AudioIoCtrl  PathManager  ReferenceManager  DotQuest libs | Tools/audioIo/\_audioPlayer.py |
| StereoPlayer | 1. It is derived class from AudioPlayer 2. Configures feature specific input channels to get frequencies from AtpAudioIOCtrl by referring Amp specific configuration. 3. Provides methods to control player like start/stop | VnVConfig/ampCfg/ioCfg.json | vnvSetup.AudioIoCtrl  PathManager  ReferenceManager  DotQuest libs | Tools/audioIo/\_audioPlayer.py |
| MonoPlayer | 1. It is derived class from AudioPlayer 2. Configures feature specific input channels to get frequencies from AtpAudioIOCtrl by referring Amp specific configuration. 3. Provides methods to control player like start/stop | VnVConfig/ampCfg/ioCfg.json | vnvSetup.AudioIoCtrl  PathManager  ReferenceManager  DotQuest libs | Tools/audioIo/\_audioPlayer.py |
| MultiChnlPlayer | 1. It is derived class from AudioPlayer 2. Configures feature specific input channels to get frequencies from AtpAudioIOCtrl by referring Amp specific configuration. 3. Provides methods to control player like start/stop | VnVConfig/ampCfg/ioCfg.json | vnvSetup.AudioIoCtrl  PathManager  ReferenceManager  DotQuest libs | Tools/audioIo/\_audioPlayer.py |
| MicInPlayer | 1. It is derived class from AudioPlayer 2. Configures feature specific input channels to get frequencies from AtpAudioIOCtrl by referring Amp specific configuration. 3. Provides methods to control player like start/stop/play | VnVConfig/ampCfg/ioCfg.json | vnvSetup.AudioIoCtrl  PathManager  ReferenceManager  DotQuest libs | Tools/audioIo/\_audioPlayer.py |
| SinkCtrlSim |  |  |  |  |
| AsmCtrlSim |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

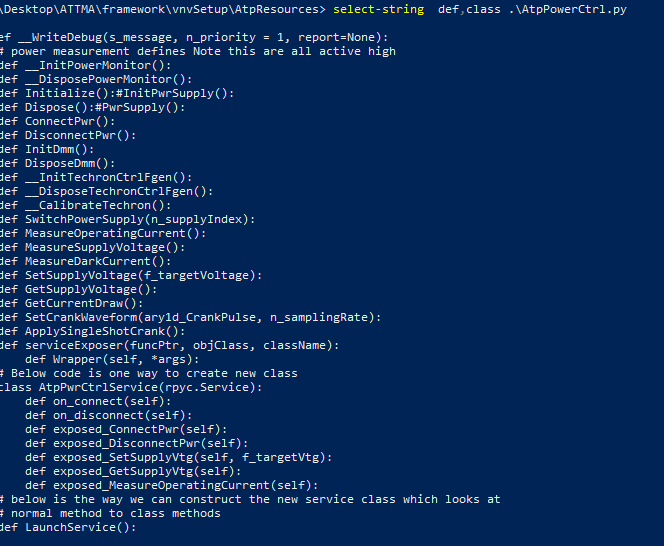


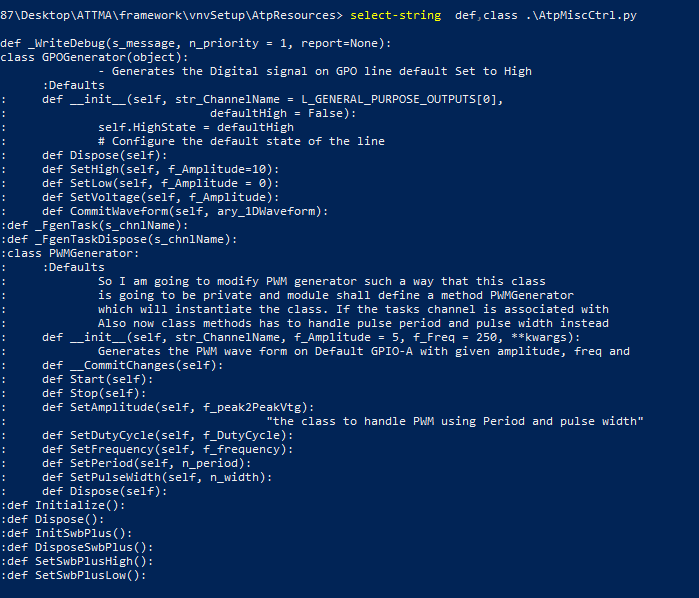




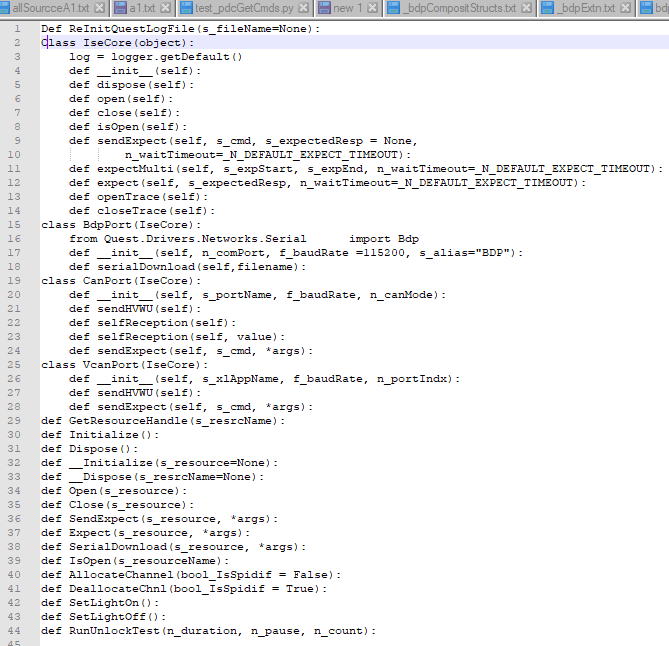




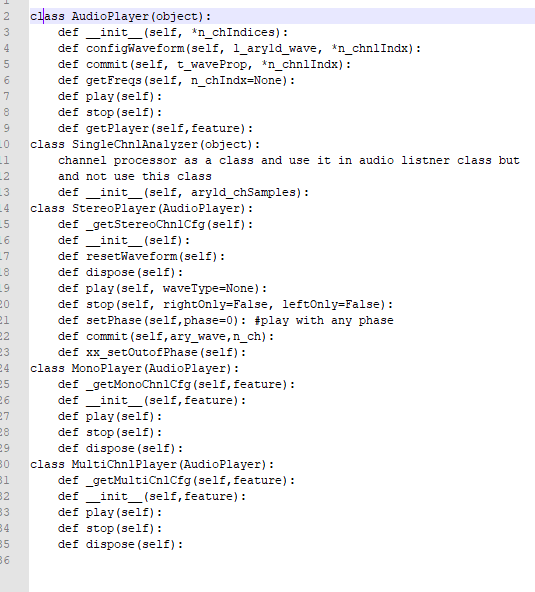




**ISECore/BDP Port/Vcan Port**



**AudioIOPlayer**



**AudioIOListner**

