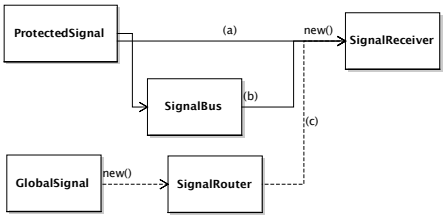
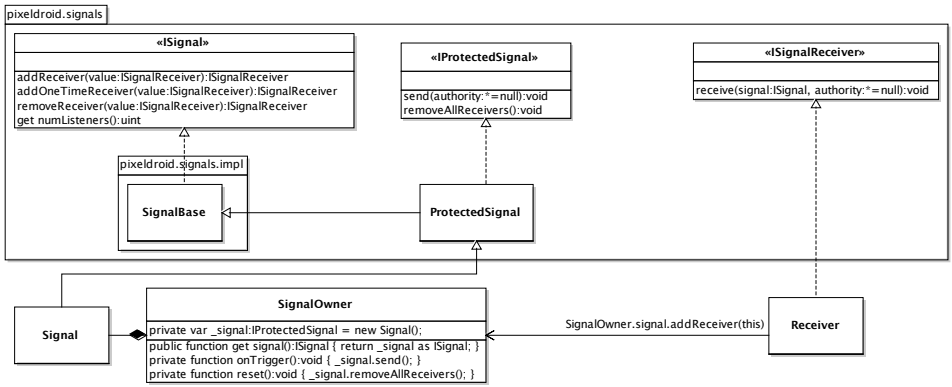
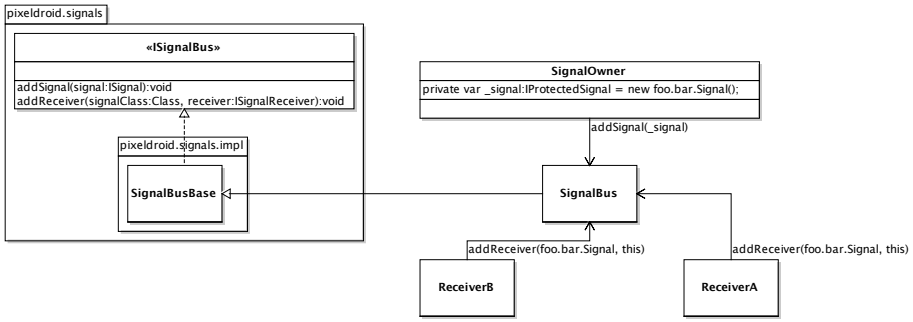
Concepts:
signal - emits information
receiver - accepts signal emissions; directly connected to signals or instantiated by routers
protected signal - connects directly to a receiver
signal bus - provides a level of indirection for signal / receiver connections
global signal - uses the signal transmitter to find suitable router for short-lived connection
signal transmitter (not shown here) - forwards signals to interested routers
router - completes indirect connections between global signals and receivers



(a)
For tight coupling, Signal and SignalReceiver provide a simple publication / subscription interface.
Classes can aggregate signals to declare the events they publish
ProtectedSignal provides elevated access for signal owners
A Signal authority is by convention whatever sent the signal, but may be anything useful, including nothing at all



(b)
For indirected coupling, SignalBus imlpements a Notifier / Observer pattern.
It functions as a third party for registration of signals and receivers



For loose coupling, the SignalRouter implements a FrontController pattern.
Receiver classes are registered for pairing with GlobalSignals via addConnection()
Users instantiate and send a GlobalSignal which the SignalTransmitter forwards to a SignalRouter who instantiates and executes the connected Receiver.
The Receiver plays the part of a Command, with its receive() method functioning as execute()