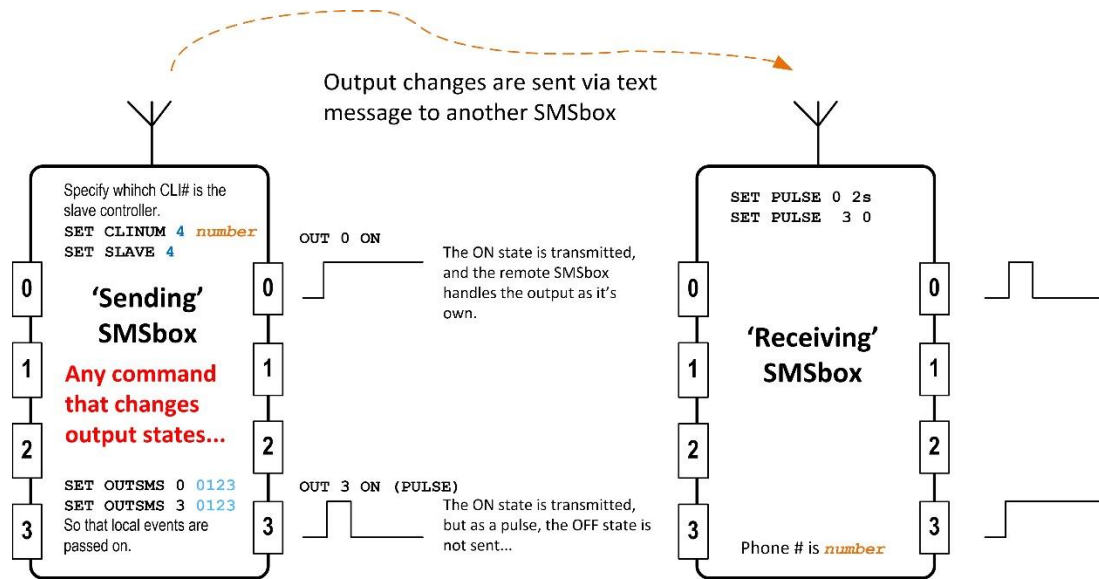


## SLAVE MODE

Slave mode uses extra outgoing messages to control output changes on *another* ControlMate. This capability allows two or more units to be located far apart, but co-operate to control external equipment. There is nothing to stop the remote unit sending back to the local unit as a separate 'slave' relationship.

If any CLI members are set as a *SLAVE* to the ControlMate, the *local* output on/off state will be mirrored to the receiving unit, but importantly, the remote units will execute the output state independently according to their own configuration.



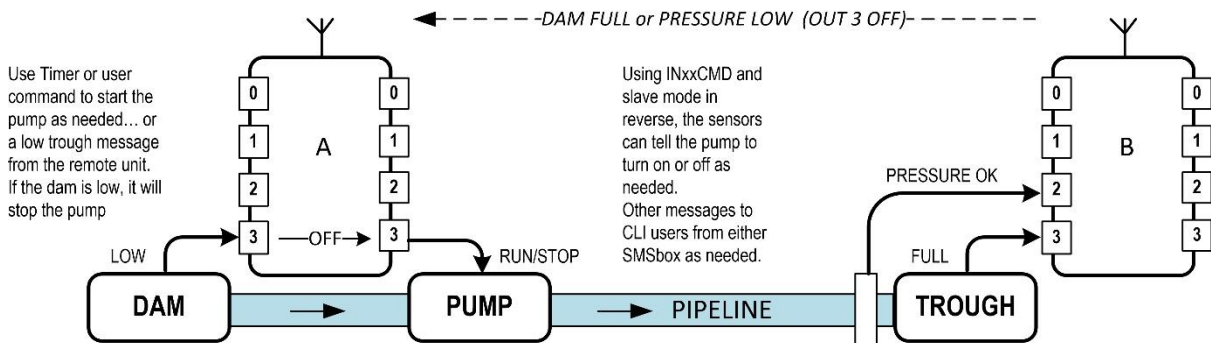
**SLAVE MODE EXAMPLE**

**Only output** messages are 'slaved' in a 1:1 manner from the 'sending' to the 'receiving' ControlMate – in either direction as configured. Only the 'sending' unit needs to have CLI SLAVE mode selected.

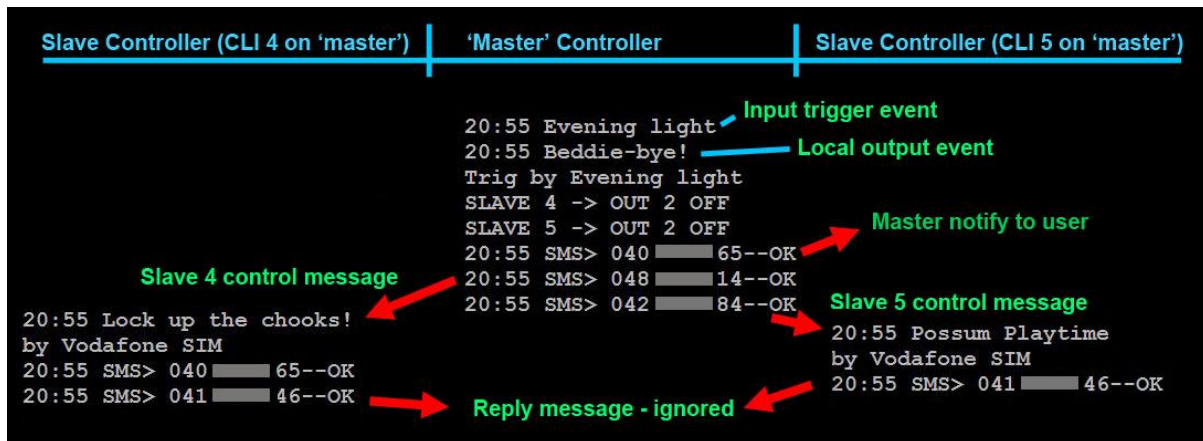
**IMPORTANT:** The sending unit must have OUTSMS enabled for the desired outputs to be 'forwarded' to the slave. The 'receiving' controller must have the 'sending' controller's number in its CLI list or the control messages will be rejected.

DON'T make both ends CLI slave to each other on the same outputs!!!

Both units can slave to each other on different outputs without a ping-pong message effect.



**A more complex SLAVE MODE example**



Typical SLAVE message transactions

The master has a trigger event – an input event in this case ‘Evening Light’ from a sensor.

Local logic says ‘turn on “Beddie-Bye” as a local output’. Then also sends a CLI user notification, and two slave commands - to CLI members 4 & 5 that have the same output number as local output “Beddie-Bye”.

These slave outputs have been named as (#4) “Lock up the chooks”, and (#5) “Possum Playtime”, along with the advice that ‘*Vodafone Test*’ (the master controller in this case) was the one that sent the command.

Due to the default SMS message logic, both slaves reply to the ‘master’ which is ignored.