Lock and Unlock Commands
OVERVIEW

The CUE controllers possess a multi-task control system, which makes it possible to run several sequences at once. In programming it is however necessary to count with states of collision that may for example occur when more sequences work with the same ports or share the same data.

LOCK AND UNLOCK COMMANDS

For example, imagine you are working with a PRESET.DAT file, in which you are loading data related to the settings of rooms, saving it via SAVE_PRESET sequence and reading it via RECALL_PRESET sequence. If each of the sequences starts running at a different moment, everything is all right. However, if one of the sequences starts at a time when the other is running, unexpected troubles may occur. For example, the RECALL_PRESET sequence may initiate a situation when part of the equipment is put into a state based on the old data stored in the PRESET.DAT file (before the saving of new data), while the other part is set according to the new data (after saving). This situation is illustrated in the left part of Fig. 1.

![Diagram](https://www.cuesystem.com/fig1.png)
The program must therefore be written in a way to prevent such collisions. To make the programming of a protection as simple as possible, new commands have been implemented in the XPL language (Lock and Unlock). These commands are supported by ipCUE firmware versions 2.05 and higher.

If sequence1 calls the Lock command and then sequence2 calls the Lock command too, sequence2 will wait until sequence1 calls the Unlock command or until sequence1 finishes. There is no influence on other sequences running at the moment.

From the systemic point of view, those commands have the following functions:

When the Lock command is executed in a sequence, the system checks whether some other sequence has already entered the Lock. If not – which means the system is „unlocked“ - the system status changes to Lock and the program continues carrying out the next commands in the given sequence. However, if the system is already „locked“, the processing of commands in the given sequence stops until the system is unlocked again (usually by the sequence which has locked the system). The unlocking is done through the Unlock command, or it occurs when the sequence that has locked the system finishes. An example of using the two commands is shown in the right part of fig. 1. Using Lock and Unlock does not affect the running of other sequences which do not use those commands.

MULTI-LEVEL LOCK

In ipCUE with firmware version up to 2.06b, the Lock command has only one level. It means if more Lock commands are placed one after another in one sequence (without Unlock between them), the second and other commands have no function (the system will remain locked) and the first Unlock command will unlock the system. You can usually avoid this situation by writing the program code carefully. But it can for example cause problems in a situation when you use the Lock and Unlock commands in Sub or Function called from a sequence. Remember that if you call subroutines (Sub and Function) from a sequence, these subroutines are still running as part of this sequence (program thread). Therefore these subroutines will not be paused by the Lock command.

If you call these subroutines from the locked state, they will unlock the main sequence, which as a result may cause unexpected behaviour of the program. This situation is illustrated in the left part of Fig. 2. You can expect the commands in the red frame to be executed in the locked state, but the system is unlocked at this moment.

In ipCUE with firmware version 2.07 and higher, this situation is solved by a multi-level lock. This may be compared to the common multi-turn door lock. If you use more Lock commands one after another, every new Lock command will lock the system to the next level (the second Lock command will lock the system to the second level, the third Lock command will lock system to the third level, etc.). If you use the Unlock command, it will unlock the system to the previous level (until the whole system is unlocked) - see the right part of Fig. 2. In this case the red-framed commands are executed in the locked state, as expected.

Note: If you are using the Scheduler program module with ipCUE firmware version 2.07 or higher, use Scheduler version 1.8 or higher.
EVERY LOCK MUST BE UNLOCKED

When using the Lock and Unlock commands, one has to remember preventing locking the system without the possibility of unlocking it. If you use Lock, do not forget to put the Unlock command in the program, otherwise the system will remain locked until the sequence which has locked the system finishes. To prevent the slowing down of other sequences waiting for unlocking, do not use Lock for longer periods of time than necessary.

Using the Lock and Unlock commands can also help you solve many conflict situations without having to program various kinds of flags.