Controller-Module communication protocol

The controller communicates with the modules using I^2C/TWI. The controller is the master (without address) and the modules are the slaves (address space begins at 8 and ends at 127). Each module must have its own unique address.

General procedure

When the game starts, the controller requires to get the addresses of the connected modules. In order to find them, the controller begins a transmission to each of the addresses and instantly ends them. If the message is acknowledged, there is a connected module. After the required n modules the controller could stop searching.

After the game started the controller will speak with every module frequently. For every used module it will send one byte, which identifies a specific command. It might be followed by additional data bytes. Then it will request one byte of data as an answer. The slave will send either an acknowledgement or the requested information. The master will know how to interpret the answered byte.

Protocol elements

These are the command bytes, the master can send:

0 - reserved. not used (yet).

1 - game start / reset. The game starts, the module should reset and initialize. One additional byte is a random number for variance in the game. The slave answers with 0 - failed or 1 - acknowledgement.

2 - get information about game status. The module should tell if one of these events happened:

- 0 - nothing happened (but module is still active)
- 1 - the user failed once to solve the module
- 2 - the module was solved and does not need to be asked again
- 3 - internal module error (should not happen)

If the module sends 255 (0xFF), the answer were not ready yet. The controller could wait a time and request one answer byte again.

3 - game ended. The game ended, the module should not accept solutions or fails anymore. The slave answers with 0 - failed or 1 - acknowledgement.