

APPENDIX A
PROTOCOL SPECIFICATION FOR COMMUNICATION
BETWEEN ARDUINOS

The aim of this protocol is the transmission of commands from the *Base Arduino* to the *Sensor Arduino*. It defines the available commands as well as their replies. The commands are sent in text and have a maximum size of 3 characters. The replies are also sent in text and have a maximum size of 4 characters. Neither the commands nor the replies are terminated with any special character.

If the *Sensor Arduino* receives a command that is not part of the specification, it replies with the 3 character sequence *NOK*.

A. REQ command

The **REQ** command is issued for requesting the LDR's value. It is sent as the 3 character sequence *REQ*, without any special character terminating it. The expected reply to this command is a character sequence with, at most, 4 characters that represents an integer value in the range from 0 to 1023. Any other value received is an error and should be discarded.

Under normal operation, the *Sensor Arduino* always executes the command.

B. STS command

The **STS** command is issued for requesting the LED's status. It is sent as the 3 character sequence *STS*, without any special character terminating it. The expected reply to this command is a character sequence with 1 character that represents an integer value. The value 0 is replied if the light is off, and 1 is replied if the light is on. Any other value received is an error and should be discarded.

Under normal operation, the *Sensor Arduino* always executes the command.

C. ON command

The **ON** command is issued to turn on the LED. It is sent as the 2 character sequence *ON*, without any special character terminating it. The expected reply to this command is the character sequence *OK*, representing the successful execution of the command. Any other value received is an error and should be discarded.

Under normal operation, the *Sensor Arduino* always executes the command.

D. OFF command

The **OFF** command is issued to turn off the LED. It is sent as the 3 character sequence *OFF*, without any special character terminating it. The expected reply to this command is the character sequence *OK*, representing the successful execution of the command. Any other value received is an error and should be discarded.

Under normal operation, the *Sensor Arduino* always executes the command.

APPENDIX B
PROTOCOL SPECIFICATION FOR COMMUNICATION
BETWEEN SERVER AND ARDUINO

The aim of this protocol is the transmission of commands from the *Server* to the *Base Arduino*. It defines the available commands as well as their replies. The commands are sent in text and have a maximum size of 3 characters. The replies are also sent in text and have a maximum size of 4 characters. Neither the commands nor the replies are terminated with any special character.

If the *Base Arduino* receives a command that is not part of the specification, it replies with the 3 character sequence *NOK*.

A. REQ command

The **REQ** command is issued for requesting the LDR's value. It is sent as the 3 character sequence *REQ*, without any special character terminating it. The expected reply to this command is a character sequence with, at most, 4 characters that represents an integer value in the range from 0 to 1023. Any other value received is an error and should be discarded.

Under normal operation, the *Base Arduino* always executes the command.

B. STS command

The **STS** command is issued for requesting the LED's status. It is sent as the 3 character sequence *STS*, without any special character terminating it. The expected reply to this command is a character sequence with 1 character that represents an integer value. The value 0 is replied if the light is off, and 1 is replied if the light is on. Any other value received is an error and should be discarded.

Under normal operation, the *Base Arduino* always executes the command.

C. ON command

The **ON** command is issued to turn on the LED. It is sent as the 2 character sequence *ON*, without any special character terminating it. The expected reply to this command is the character sequence *OK*, representing the successful execution of the command. Any other value received is an error and should be discarded.

Under normal operation, the *Base Arduino* always executes the command.

D. OFF command

The **OFF** command is issued to turn off the LED. It is sent as the 3 character sequence *OFF*, without any special character terminating it. The expected reply to this command is the character sequence *OK*, representing the successful execution of the command. Any other value received is an error and should be discarded.

Under normal operation, the *Base Arduino* always executes the command.

APPENDIX C
PROTOCOL SPECIFICATION FOR COMMUNICATION
BETWEEN SMARTPHONE AND SERVER

The aim of this protocol is the transmission of commands from the *Android Smartphone* to the *Server*. It defines the available commands as well as their replies. The commands are sent in text and have a maximum size of 4 characters. The replies are also sent in text and have a maximum size of 4 characters. The commands and the replies are terminated with the special character $\backslash n$.

If the *Server* receives a command that is not part of the specification, it replies with the 4 character sequence *NOK* $\backslash n$.

A. *ON command*

The **ON** command is issued to turn on the LED. It is sent as the 3 character sequence *ON* $\backslash n$. The expected reply to this command is the character sequence *OK* $\backslash n$, representing the successful execution of the command.

Under normal operation, the *Server* may not execute the command. This is signaled by replying with the character sequence *NOP* $\backslash n$. Any other value received is an error and should be discarded.

B. *OFF command*

The **OFF** command is issued to turn off the LED. It is sent as the 4 character sequence *OFF* $\backslash n$. The expected reply to this command is the character sequence *OK* $\backslash n$, representing the successful execution of the command. Any other value received is an error and should be discarded.

Under normal operation, the *Server* always executes the command.

C. *STS command*

The **STS** command is issued for requesting the LED's status. It is sent as the 4 character sequence *STS* $\backslash n$. The expected reply to this command is a character sequence with 2 characters that represents an integer value. The value *0* $\backslash n$ is replied if the light is off, and *1* $\backslash n$ is replied if the light is on. Any other value received is an error and should be discarded.

Under normal operation, the *Server* always executes the command.

APPENDIX D
GAME EXTENSIONS TO THE APPENDIX C PROTOCOL

The aim of this extension to the protocol described in *Appendix C* is the transmission of commands from the *Client* to the *Server* to control the game. The commands are sent in text and have a maximum size of 8 characters. The replies are also sent in text and have a maximum size of 4 characters. The commands and replies are terminated with the special character $\backslash n$.

If the *Server* receives a command that is not part of the specification, it replies with the 4 character sequence *NOK* $\backslash n$.

A. *RES command*

The **RES** command is issued to reset the value that the user has to guess. It is sent as the 4 character sequence *RES* $\backslash n$. The expected reply to this command is the character sequence *OK* $\backslash n$, representing the successful execution of the command. Any other value received is an error and should be discarded.

Under normal operation, the *Server* always executes the command.

B. *G command*

The **G** command is issued to send the user's guess. It is sent as the 8 character sequence *G X Y Z* $\backslash n$, where *X* represents the most significant digit of the guess, *Y* the second most significant digit of the guess, and *Z* the least significant digit of the guess. The expected reply to this command is the character sequence *OK* $\backslash n$, representing the successful execution of the command. Any other value received is an error and should be discarded.

Under normal operation, the *Server* always executes the command.