

# LinTronic

<http://www.lintronic.dk>

## LinTronic\_IR.DLL

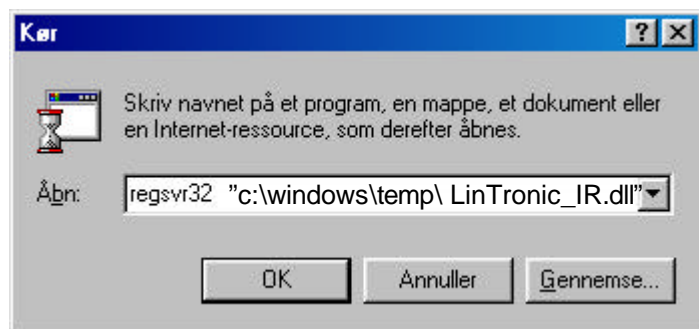
This is a description of the Visual Basic 6.0 created FUGA.DLL used to control 36-40 and 455 KHz Infrared Receivers and Transmitters in the FUGA and TableTop series (referred to as PRODUCT) from LinTronic.

The DLL is created based on the datasheet 'Protocol' available on our web-site.  
Please refer to the internet for update and details on parameters range.

The advantage of using a DLL is that you save a lot of time in code writing. Also you do not need to know anything about sending/receiving data on a serial CommPort, store codes onto the computer and how to calculate the checksum etc.

- If you are interested in using the LinTronic\_IR.DLL, please send me an email and you will receive a copy at no charge.

**Download and register** the LinTronic\_IR.DLL running Regsvr32.exe pointing to the path where you downloaded, unzipped the LinTronic\_IR.DLL (LinTronic\_IR.ZIP)



If you need help on how to use this DLL in your own Windows program, do not hesitate to contact us.

# LinTronic

<http://www.lintronic.dk>

## LinTronic\_IR.DLL EVENTS

### **Public Event Error(ByVal ErrorCode As String)**

This is the sub for critical errors in the FUGA.DLL..

Displaying the errors here should prevent the program from going down.

### **Public Event ReceivedData(ByVal ReceivedString As String)**

When the PRODUCT device receives a valid data package, the entire datastring is send to this event for your optional handling.

### **Public Event TransmittedData (ByVal TransmittedString As String)**

When a command is transmitted to the PRODUCT, the entire datastring is send to this event for your optional handling.

### **Public Event CodeStream(ByVal CodeString As String, ByVal CodeLength As Long)**

When the IT-FUGA transmitter detects an infrared code to be stored, the entire data-string is send to this event for your optional handling. We use it in the configuration-program to draw a picture of the code. If you use this software, you can mail us a picture of the code and the codes-stream data if any problems occur.

# LinTronic

<http://www.lintronic.dk>

## LinTronic\_IR.DLL FUNCTIONS

All functions return the PRODUCT commands result string.

The functions will in most cases perform a data validation. If data are out of range the result string will be "Error: " followed by an explanation.

Be aware that critical errors are raised in the Public Event Error.

**ChangeAddress(ByVal ChangeFromAddress As Byte, ByVal ChangeToAddress As Byte) As String**  
Changes the address of the PRODUCT. All devices are addressed "1" from the factory. It is recommended that you re-address your devices from 1 to the highest available number (32, 31, 30, ...).

**ChangeBaudrate(ByVal DeviceAddress As Byte, ByVal BaudrateNumber As Byte) As String**  
Changes the communication baudrate of the PRODUCT.

**CodeFlashSec(ByVal DeviceAddress As Byte, ByVal Output As Byte, ByVal Code As String, ByVal MiliSeconds As Long) As String**  
The PRODUCT can be programmed to produce a time-length programmable pulse upon detection of a specific infrared code.

**CodeOff(ByVal DeviceAddress As Byte, ByVal Output As Byte, ByVal Code As String) As String**  
The PRODUCT can be programmed to turn a specific output OFF upon detection of a specific infrared code.

**CodeOn(ByVal DeviceAddress As Byte, ByVal Output As Byte, ByVal Code As String) As String**  
The PRODUCT can be programmed to turn a specific output ON upon detection of a specific infrared code.

**CodeToggle(ByVal DeviceAddress As Byte, ByVal Output As Byte, ByVal Code As String) As String**  
The PRODUCT can be programmed to toggle a specific output upon detection of a specific infrared code.

**Conson(ByVal DeviceAddress As Byte, ByVal Master As Byte, ByVal Number As Byte, ByVal Channel As Byte, ByVal OnOff As Byte) As String**  
Controls a Conson Light System. Experience required.

**DeleteCodeFromComputer(ByVal CodeName As String) As String**  
Deletes a specific IR-code from the computers collection.

**DirectComm() As String**  
Instructs the PRODUCT to work in Direct Mode.

**FlashOutput(ByVal DeviceAddress As Byte, ByVal Output As Byte, ByVal OnTime As Long, ByVal OffTime As Long, ByVal Repeat As Long) As String**  
The PRODUCT can be programmed to produce a time-length programmable pulse.

**GetBeoCodeFromBus(ByVal DeviceAddress As Byte, ByVal Mode As Byte) As String**  
Instructs whether to copy a detected B&O code on the bus-system, being send to the computer (adr. 00).

**GetCodeFromComputer(ByVal CodeName As String) As String**  
Request an infrared codeform the collection.

# LinTronic

<http://www.lintronic.dk>

**GetTemp(ByVal DeviceAddress As Byte) As String**

Reads the PRODUCT temperature from the optional temperature sensor.

**GetType(ByVal DeviceAddress As Byte) As String**

Reads the PRODUCT type ID.

**GetVersion(ByVal DeviceAddress As Byte) As String**

Reads the PRODUCT software version.

**InputMode(ByVal DeviceAddress As Byte, ByVal Input\_1 As Byte, ByVal Input\_2 As Byte) As String**

Sets the PRODUCT input functionality.

**InputRequest(ByVal DeviceAddress As Byte) As String**

Reads the status of the PRODUCT inputs.

**OpenCommport(ByVal PortNumber As Byte, ByVal Settings As String) As String**

Opens commport 1-99 using the requested Baudrate.

**OutputMode(ByVal DeviceAddress As Byte, ByVal Input\_1 As Byte, ByVal Input\_2 As Byte) As String**

Sets the PRODUCT output functionality.

**OutputRequest(ByVal DeviceAddress As Byte) As String**

Reads the status of the PRODUCT outputs.

**OutputsOff(ByVal DeviceAddress As Byte, ByVal Output\_1 As Byte, ByVal Output\_2 As Byte) As String**

Sets specific PRODUCT outputs Off.

**OutputsOn(ByVal DeviceAddress As Byte, ByVal Output\_1 As Byte, ByVal Output\_2 As Byte) As String**

Sets specific PRODUCT outputs On.

**PhoneLineBusyCode\_038(ByVal DeviceAddress As Byte, ByVal CodeStream As String) As String**

Stores the 38 KHz code to be transmitted when phonenumber is busy

**PhoneLineBusyCode\_455(ByVal DeviceAddress As Byte, ByVal Address As String, ByVal Command As String, ByVal Repeat As String) As String**

Stores the 455 KHz code to be transmitted when phonenumber is busy

**PhoneLineIdleCode\_038(ByVal DeviceAddress As Byte, ByVal CodeStream As String) As String**

Stores the 455 KHz code to be transmitted when phonenumber is idle

**PhoneLineIdleCode\_455(ByVal DeviceAddress As Byte, ByVal Address As String, ByVal Command As String, ByVal Repeat As String) As String**

Stores the 455 KHz code to be transmitted when phonenumber is idle

**PollDevice(ByVal DeviceAddress As Byte) As String**

Polls the device.

**PolledComm() As String**

Instructs the PRODUCT to work in Polled Mode.

# LinTronic

<http://www.lintronic.dk>

**ReadSwitches(ByVal DeviceAddress As Byte) As String**

Reads the internal switch settings

**RespondDelay(ByVal DeviceAddress As Byte, ByVal Delay As Byte) As String**

Instructs the PRODUCT to set the Respond delay. Usefull to slow down reply from the PRODUCT if working in relatively slow environments like for example 'Commport over Ethernet', etc.

**RespondOnOff(ByVal FugaAddress As Byte, ByVal OnOff As Byte) As String - OBSOLETED**

Instructs the PRODUCT to turn Respond On or Off.

**SaveCodeToComputer(ByVal CodeName As String, ByVal CodeStream As String) As String**

Saves a detected IR-code to computers collection.

**SendBeoCode(ByVal DeviceAddress As Byte, ByVal BeoAddress As Byte, ByVal BeoCommand As Byte, ByVal Repeat As Byte) As String**

Instructs PRODUCT to send B&O code

**SendDirect(ByVal SendString As String, ByVal Repeat As Byte, ByVal TimeOut As Integer) As String**

Allows you to send a completed package including checksum ( or --- ) without the DLL doing anything

**SendIRCode(ByVal DeviceAddress As Byte, ByVal CodeStream As String) As String**

Instructs the PRODUCT to transmit the specific code.

**SendSonyCode(ByVal DeviceAddress As Byte, ByVal SonyDevice As Byte, ByVal SonyCommand As Byte, ByVal Repeat As Byte) As String**

Instructs the PRODUCT to transmit the specific code.

**TransmitInfraRedCode(ByVal DeviceAddress As Byte, ByVal InfraRedCodeType As Integer, ByVal Data As String) As String**

Instructs the PRODUCT to transmit the specific code.

**TransmitterMode(ByVal DeviceAddress As Byte, ByVal Mode As Byte) As String**

Instructs the PRODUCT to enter a specific Mode.