

# eigerPanel70C

## Hardware-Manual



**eigerGraphics – The smart Touchpanel Solutions**  
Product line of S-TEC electronics AG, Zug, Switzerland.

## eigerPanel – versatile operator and control panel with color VGA display and touch screen

- I/O- digital (counter input, function keys, digital and PWM output)
- I/O- analog (power inputs, analog inputs, analog outputs)
- Audio output, stereo
- Serial interfaces RS485, RS232, I<sup>2</sup>C
- Output for TFT VGA 800x480 LCD
- CompactFlash Card for your Data (e.g. your application, pictures, CSV files)
- Battery buffered real time clock
- DC input voltage 9..30V
- Wattage: 6.0W typical
- 66 MHz clock signal
- Boot time: 4 seconds

The eigerPanel is available together with several display sizes. Our standard sizes are 5.7" VGA and 7" WVGA TFT with LED backlight. Further information at [www.eigergraphics.com](http://www.eigergraphics.com).

## Contents

Contents.....	2
Hardware Overview .....	3
Pin Assignments .....	4
Specifications.....	14
Dimensions.....	16
Dimensions of Housing and Synthetic Frame.....	17
Firm- and Software.....	17
Further Information, Contact Data, Support.....	17
Datasheet Version .....	17
Notes .....	19

**Hardware Overview**

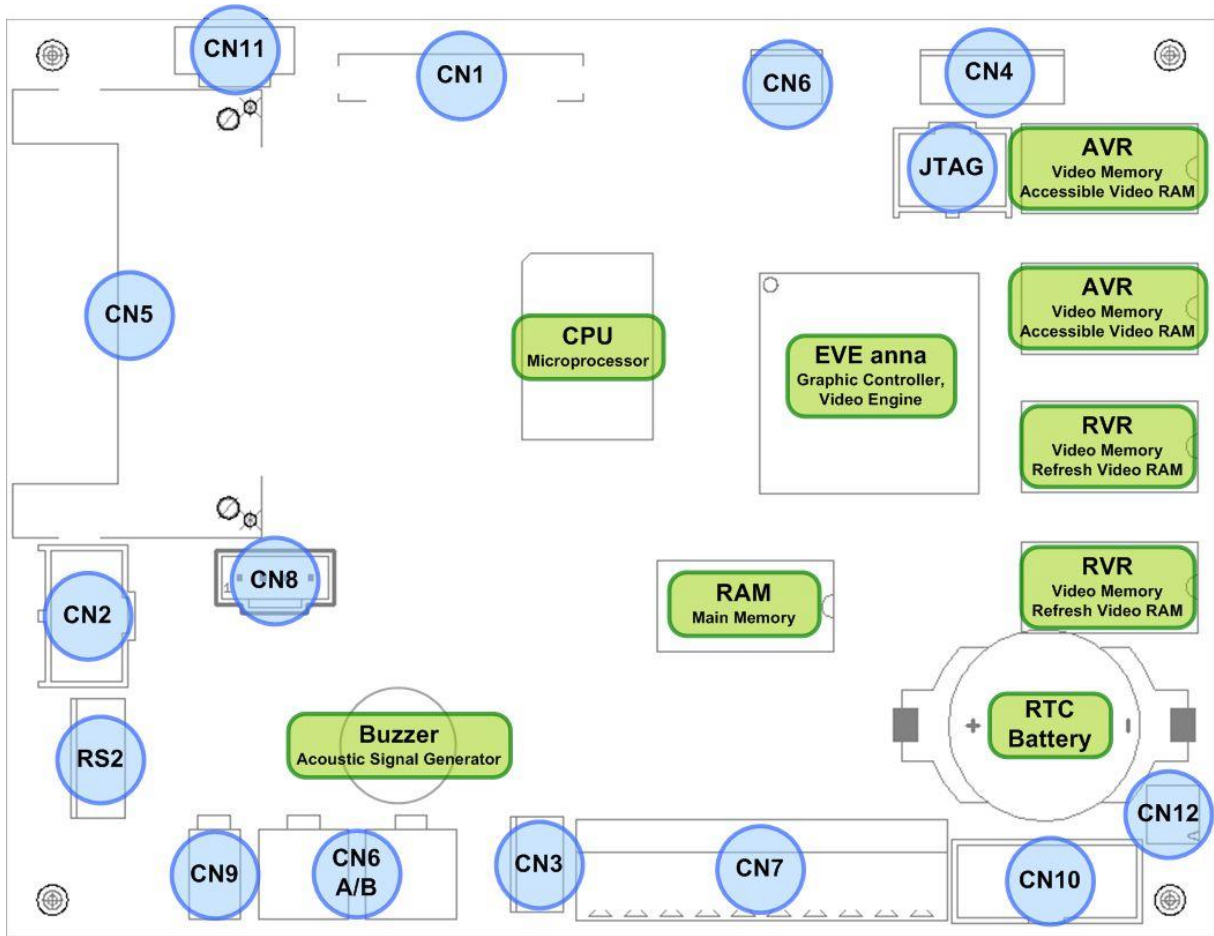


Figure 1: Ports of F70C\_v1001

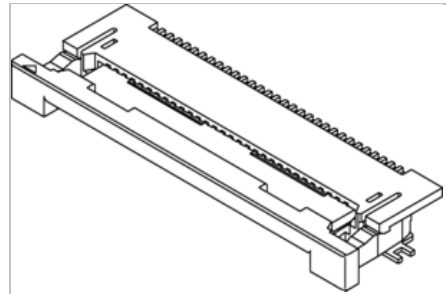
CNx	Description	Link	Page
CN1	Interface for display TFT WVGA	<a href="#">&gt;&gt;</a>	4
CN2	FOX-COM1 (UART1) programming and debug interface RS232	<a href="#">&gt;&gt;</a>	5
CN3	Analog output sound DA	<a href="#">&gt;&gt;</a>	5
CN4	Analog input NTC, potentiometer	<a href="#">&gt;&gt;</a>	6
CN5	Interface for CompactFlash Card	<a href="#">&gt;&gt;</a>	8
CN6	Analog input potentiometer	<a href="#">&gt;&gt;</a>	9
CN6A/B	BUS (serial interface RS485)	<a href="#">&gt;&gt;</a>	9
CN7	external inputs and outputs	<a href="#">&gt;&gt;</a>	10
CN8	BUS (serial interface I <sup>2</sup> C)	<a href="#">&gt;&gt;</a>	11
CN9	Power supply VDC	<a href="#">&gt;&gt;</a>	11
CN10	Digital input for 8 function keys (keyboard)	<a href="#">&gt;&gt;</a>	11
CN11	Interface for external touch screen (optional)	<a href="#">&gt;&gt;</a>	12
CN12	Interface for touch screen	<a href="#">&gt;&gt;</a>	12
JTAG	Programming interface for EVE anna (CPLD)	<a href="#">&gt;&gt;</a>	13
RS2	FOX-COM2 (UART2) serial interface RS232	<a href="#">&gt;&gt;</a>	13
-	Buzzer	<a href="#">&gt;&gt;</a>	14

## Pin Assignments

### CN1 Port for Display TFT WVGA with backlight and touch

Connector type Molex 54132, 40-Pol (0.5mm, bottom Contact)

Pin 1	LGND	Logic Ground
Pin 2	LGND	Logic Ground
Pin 3	ADJ	Adjust
Pin 4	L3V3	Logic +3.3V
Pin 5	L3V3	Logic +3.3V
Pin 6	L3V3	Logic +3.3V
Pin 7	L3V3	Logic +3.3V
Pin 8	L3V3	Logic +3.3V
Pin 9	DE	Data Enable
Pin 10	LGND	Logic Ground
Pin 11	LGND	Logic Ground
Pin 12	LGND	Logic Ground
Pin 13	B5	Blue bit 5
Pin 14	B4	Blue bit 4
Pin 15	B3	Blue bit 3
Pin 16	LGND	Logic Ground
Pin 17	B2	Blue bit 2
Pin 18	B1	Blue bit 1
Pin 19	LGND	Logic Ground
Pin 20	LGND	Logic Ground
Pin 21	G5	Green bit 5
Pin 22	G4	Green bit 4
Pin 23	G3	Green bit 3
Pin 24	LGND	Logic Ground
Pin 25	G2	Green bit 2
Pin 26	G1	Green bit 1
Pin 27	LGND	Logic Ground
Pin 28	LGND	Logic Ground
Pin 29	R5	Red bit 5
Pin 30	R4	Red bit 4
Pin 31	R3	Red bit 3
Pin 32	LGND	Logic Ground
Pin 33	R2	Red bit 2
Pin 34	R1	Red bit 1
Pin 35	LGND	Logic Ground
Pin 36	LGND	Logic Ground
Pin 37	LGND	Logic Ground
Pin 38	PCLK	Processor Clock
Pin 39	LGND	Logic Ground
Pin 40	LGND	Logic Ground



passendes Kabel: F4396 (76mm)

auf Anfrage: F4375 (85mm)

eigerScript Some EVM commands of class **Display**:

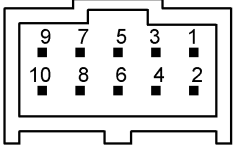
```

Display.Clear( )
Display.ClearColor( VarInt:Color )
Display.CopyLineDown( )
Display.CopyLineRight( )
Display.CopyWindow( )
Display.Direct( )
Display.Prepare( )
Display.RestoreWriteMode( )
Display.Show( )
Display.ShowLineDown( )
Display.ShowLineRight( )
Display.ShowTranslateWindow( )
Display.ShowWindow( )
Display.ShowWindowEffect( VarInt:ShowWindowEffect , VarInt:Speed )

```

< [Contents](#)

## CN2 S-PROG20 Download / Debug (FOX-COM1, UART1)

<b>Connector type</b>		Molex C-Grid III 10-Pol	Wiring Harness: Molex C-Grid III connector housing 10-Pol
Pin 1	HOLD	/Hold Line	 <p>matching cable: F4337</p>
Pin 2	LGND	Logic Ground	
Pin 3	BYTE	Byte Line	
Pin 4	L12V	Logic +12V	
Pin 5	CNV	Cnvss Line	
Pin 6	L3V3	Logic 3.3V	
Pin 7	RES	/Reset Line	
Pin 8	LGND	Logic Ground	
Pin 9	TXD_1	S-PROG20 Transmit Line	
Pin 10	RXD_1	S-PROG20 Receive Line	

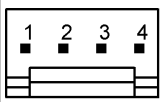
<b>eigerScript</b>	eVM commands of Class <b>Debug</b> :
<code>Debug.Mark( Str )</code>	
<code>Debug.DumpBytes( VarInt )</code>	
<code>Debug.Print_IntegerHex( Str , VarInt )</code>	
<code>Debug.Print_LongHex( Str , VarLong )</code>	
<code>Debug.Print_SingleHex( Str , VarSingle )</code>	
<code>Debug.Print_Char( VarInt )</code>	
<code>Debug.Print_CLRF( )</code>	
<code>Debug.Print_String( VarStr )</code>	
<code>Debug.Print_StringHeader( VarStr )</code>	

### Circuit examples

<p>Connection of a reset key:</p>	
<p>Connection to the debug- and programming-Interface: (F4337)</p>	

[< Contents](#)

## CN3 Analog Outputs for Sound DA

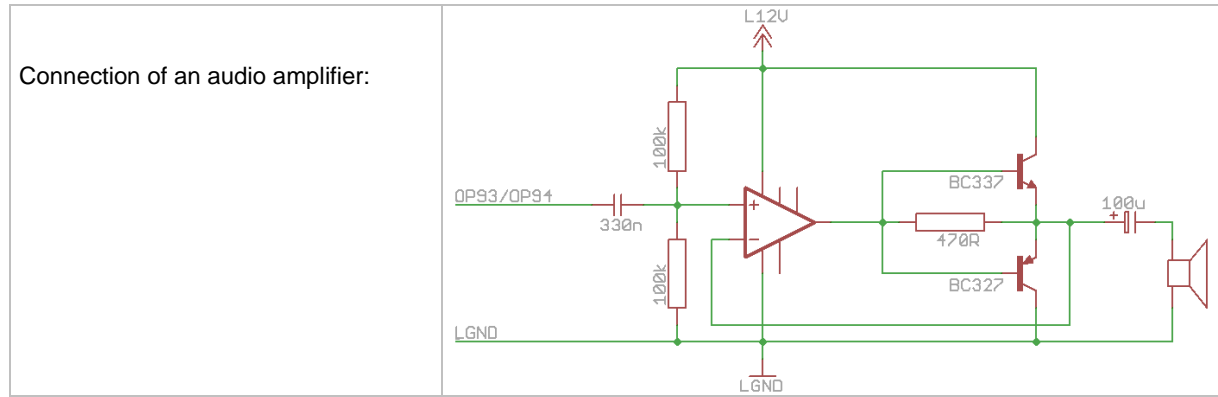
<b>Connector type</b>		TYCO CST100 4-Pol	Wiring Harness: AMP MTA100 4-Pol
Pin 1	OP93	Analog Out / Sound DA Out	 <p>matching cables: F4388 , F4389</p>
Pin 2	LGND	Logic Ground	
Pin 3	OP94	Analog Out / Sound DA Out	
Pin 4	L5V	Logic +5V	

<b>eigerScript</b>	eVM commands of class <b>Sound</b> :
<code>Sound.SetSamplingRate( VarInt:SamplingRate_11025 )</code>	
<code>Sound.SetBitsPerSample( VarInt: BitsPerSample )</code>	
<code>Sound.PlaySamples( VarInt:Sound_MODE_STEREO , VarStr:File , VarLong:StartOffset , VarLong:NoOfSamples , VarInt:Loops )</code>	
<code>Sound.InstallLocal( labelRelative24:Event )</code>	

<b>eigerScript</b>	eVM commands of class <b>File</b> :
<code>File.Read_ToBuffer( VarInt:Handle.I , VarStr:Buffer , VarLong:Offset , VarLong:NumBytes , VarLong:BytesRead )</code>	

<b>eigerScript</b>	eVM commands of class <b>Buffer</b> :
	<code>Buffer.Get_UINT8_I( VarInt:UINT8 , VarStr:Buffer , VarLong:Offset )</code>
	<code>Buffer.Get_IN8_I( VarInt:INT8 , VarStr:Buffer , VarLong:Offset )</code>
	<code>Buffer.Get_INT16( VarInt:INT16 , VarStr:Buffer , VarLong:Offset )</code>
	<code>Buffer.Get_UINT16( VarInt:UINT16 , VarStr:Buffer , VarLong:Offset )</code>
	<code>Buffer.Get_INT32( VarInt:INT32 , VarStr:Buffer , VarLong:Offset )</code>
	<code>Buffer.Put_UINT8_I( VarInt:UINT8 , VarStr:Buffer , VarLong:Offset )</code>

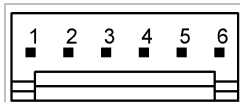
**Circuit examples**



[< Contents](#)

**CN4 Analog Inputs for NTC, Potentiometer**

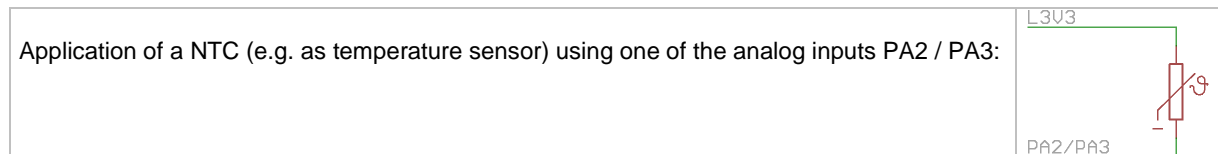
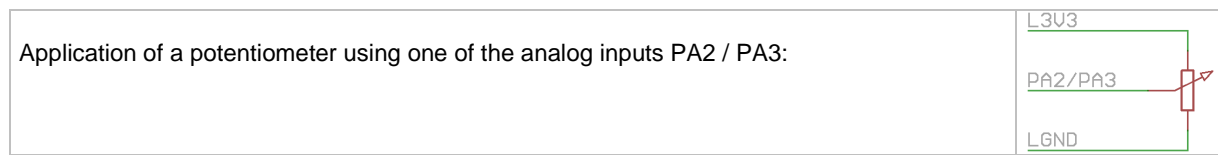
<b>Connector type</b>	TYCO CST100 6-Pol	<b>Wiring Harness:</b> AMP MTA100 6-Pol
Pin 1	L3V3	Logic +3.3V
Pin 2	PA2	Analog In
Pin 3	L3V3	Logic +3.3V
Pin 4	PA3	Analog In
Pin 5	LGND	Logic Ground
Pin 6	LGND	Logic Ground



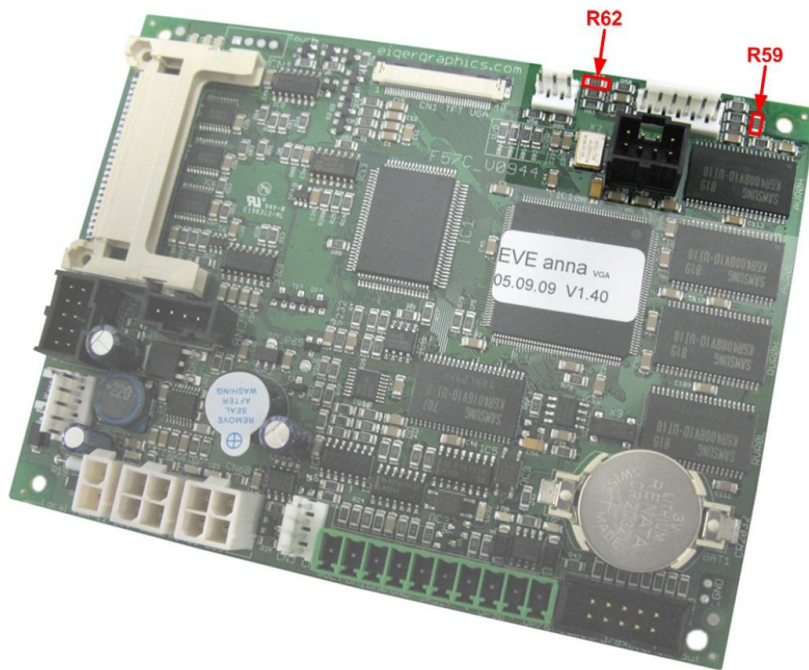
Matching cable: F4340

<b>eigerScript</b>	eVM commands of class <b>InOut</b> :
	<code>InOut.Read_ADC( VarInt:Kanal_PA2_PA3 , VarInt:Value_0-1023 )</code>

**Circuit examples**



By removal of the resistors R59 and R62 (see figure 2) the analog inputs PA2 and PA3 become voltage measurement inputs. Permissible rated voltage: 0..10VDC.



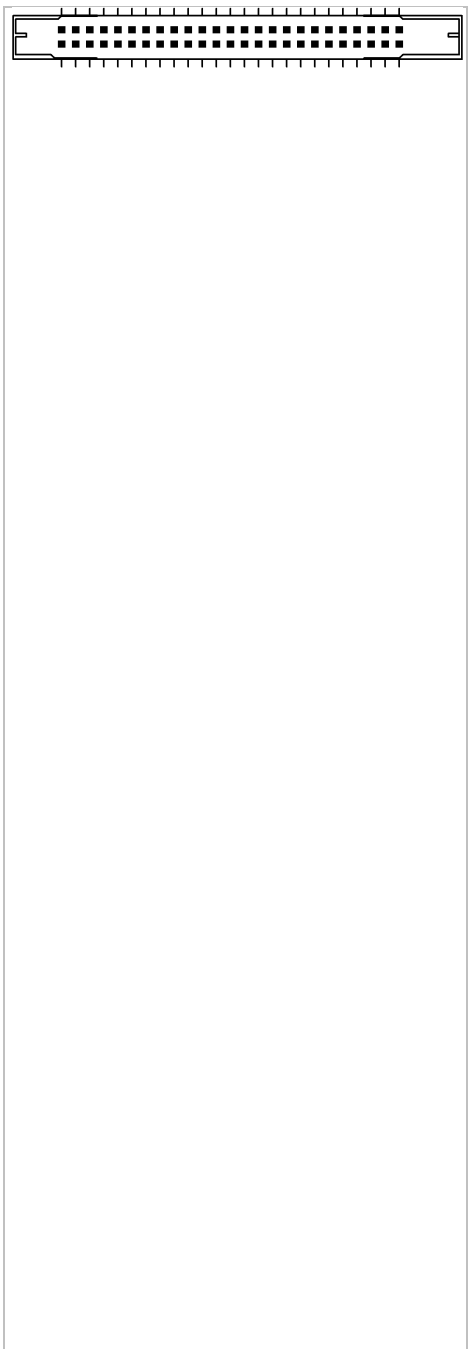
**Image 2:** Resistors PA2 / PA3 on F70H

[< Contents](#)

**CN5 CompactFlash Card Port/ Mass Storage**

Connector type TCO 752DF50NC00001 , CFA Type 1 , 50-Pol

Pin 1	LGND	Logic Ground
Pin 2	CD3	
Pin 3	CD4	
Pin 4	CD5	
Pin 5	CD6	
Pin 6	CD7	
Pin 7	C/CE2.2	
Pin 8	LGND	Logic Ground
Pin 9	LGND	Logic Ground
Pin 10	LGND	Logic Ground
Pin 11	LGND	Logic Ground
Pin 12	LGND	Logic Ground
Pin 13	CF+3.3V	Logic +3.3V
Pin 14	LGND	Logic Ground
Pin 15	LGND	Logic Ground
Pin 16	LGND	Logic Ground
Pin 17	LGND	Logic Ground
Pin 18	CA3	
Pin 19	CA2	
Pin 20	CA1	
Pin 21	CD0	
Pin 22	CD1	
Pin 23	CD2	
Pin 24	IOCS16	
Pin 25	/CD2	
Pin 26	/CD1	
Pin 27	CD11	
Pin 28	CD12	
Pin 29	CD13	
Pin 30	CD14	
Pin 31	CD15	
Pin 32	C/CE2.3	
Pin 33	VS1	
Pin 34	C/RD	
Pin 35	C/WR	
Pin 36	CF+3.3V	Logic +3.3V
Pin 37	INTRQ	
Pin 38	CF+3.3V	Logic +3.3V
Pin 39	/CSEL	
Pin 40	VS2	
Pin 41	CFRES	
Pin 42	/IORDY	
Pin 43	INPACK	
Pin 44	CF+3.3V	Logic +3.3V
Pin 45	/DASP	
Pin 46	PDIAG	
Pin 47	CD8	
Pin 48	CD9	
Pin 49	CD10	
Pin 50	LGND	Logic Ground
Pin 51	LGND	Logic Ground
Pin 52	LGND	Logic Ground



eigerScript Some eVM commands of class **File**:

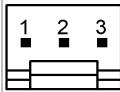
```
File.AppendString( VarStr:FileName , VarStr:String )
File.DeleteFile( VarStr:FileName )
File.Read_CSV( VarStr:FileName , VarStr:Buffer )
File.Read_EGI( VarStr:FileName )
File.Read_TextFile( VarStr:FileName , VarStr:Buffer )
File.ScreenDump( VarStr:FileName )
```

[< Contents](#)



## CN6 Analog Input for Potentiometer

Connector type	TYCO CST100 3-Pol	Wiring Harness: AMP MTA100 3-Pol
Pin 1	L3V3	Logic +3.3V
Pin 2	PA0	Analog IN
Pin 3	LGND	Logic Ground

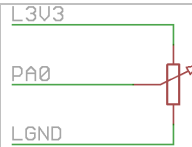


Matching cable: F4336

eigerScript	eVM Command-Class: <b>InOut</b>
<code>InOut.Read_ADC( VarInt:Kanal_PA0 , VarInt:Value_0-1023 )</code>	


### Circuit examples

Use of a potentiometer by the use of analog input PA0:



---

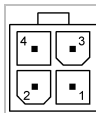
Use of an NTC (e.g. as temperature sensor) by the use of analog input PA0:



[< Contents](#)

## CN6A / CN6B BUS, Serial Interface RS485

Connector type	Molex MiniFit jr. 5566 4-Pol	Wiring Harness: Molex MiniFit jr. 5566 housing 4-Pol
Pin 1	L12V	Logic +12V
Pin 2	RXTX-	RS485 Line minus
Pin 3	LGND	Logic Ground
Pin 4	RXTX+	RS485 Line plus



Matching cable: F4368

eigerScript	eVM commands of class <b>DataServer</b> :
<code>DataServer.ReceiveToFile( VarInt:NodeAdress , VarStr:FileName )</code>	
<code>DataServer.ReceiveToString( VarInt:NodeAdress , VarStr:ZielString )</code>	
<code>DataServer.Rx_ReadInteger( VarInt:NodeAdress , VarInt:Register , VarInt:Value )</code>	
<code>DataServer.Rx_ReadLong( VarInt:NodeAdress , VarInt:Register , VarLong:Value )</code>	
<code>DataServer.Rx_String( VarInt:NodeAdress , VarInt:Register , VarLong:Value )</code>	
<code>DataServer.SendFromFile( VarInt:NodeAdress , VarStr:FileName )</code>	
<code>DataServer.SendFromString( VarInt:NodeAdress , VarStr:QuellString )</code>	
<code>DataServer.Tx_Command( VarInt:NodeAdress , VarInt:Register , VarInt:Command )</code>	
<code>DataServer.Tx_String( VarInt:NodeAdress , VarInt:Register , VarStr:String )</code>	
<code>DataServer.Tx_WriteInteger( VarInt:NodeAdress , VarInt:Register , VarInt:Value )</code>	
<code>DataServer.Tx_WriteLong( VarInt:NodeAdress , VarInt:Register , VarLong:Value )</code>	

eigerScript	eVM commands of class <b>Debug</b> :
<code>Debug.Mark( Str )</code>	
<code>Debug.DumpBytes( VarInt )</code>	
<code>Debug.Print_IntegerHex( Str , VarInt )</code>	
<code>Debug.Print_LongHex( Str , VarLong )</code>	
<code>Debug.Print_SingleHex( Str , VarSingle )</code>	
<code>Debug.Print_Char( VarInt )</code>	
<code>Debug.Print_CLRF( )</code>	
<code>Debug.Print_String( VarStr )</code>	
<code>Debug.Print_StringHeader( VarStr )</code>	

[< Contents](#)

## CN7 External Inputs and Outputs

<b>Connector type</b>		Phoenix MCS 3.81mm 10-Pol	Wiring Harness: PTR female connector strips 3.81mm
Pin 1	OP76	Open collector (PWM) pulse width 0..1000	<p>Matching cable: F4405</p>
Pin 2	UVL	Supply plus	
Pin 3	OP90	Open collector	
Pin 4	UVL	Supply plus	
Pin 5	IP91+	Input plus / 12-30 VAC / opto isolated	
Pin 6	IP91-	Input minus	
Pin 7	LGND	Logic Ground	
Pin 8	DA0	0 .. 255 → 0 .. 10VDC	
Pin 9	DA1	0 .. 255 → 0 .. 10VDC	
Pin 10	VI	0 .. 50V → 0 .. 1023	

<b>eigerScript</b>	eVM commands of class <b>InOut</b> :
<code>InOut.Read_ADC( VarInt:Kanal , VarInt:ADC-Value0-1023 )</code>	
<code>InOut.DigitalOutputDriver( VarInt.KanalOutput_OP90 , VarInt:Output_Function )</code>	
<code>InOut.PWM_Out( VarInt:Output_OP72_OP76 , VarInt:Value_0-1000 ) <sup>1)</sup></code>	
<code>InOut.DA_Out_8( VarInt:Output_OP93_OP94 , VarInt:Value_0-255 )</code>	
<code>InOut.Read_Input( VarInt:Input_IP91 , VarInt:Value )</code>	
<code>InOut.UpCounter_Get( VarInt:Input_IP91 , VarLong:Counter )</code>	
<code>InOut.UpCounter_Reset( VarInt:Input_IP91 )</code>	
<code>InOut.DownCounter_Get( VarInt:Input_IP91 , VarLong:Counter )</code>	
<code>InOut.DownCounter_Load( VarInt:Input_IP91 , VarLong:Counter )</code>	
<code>InOut.DownCounter_Zero_InstallLocal( labelRelative24:Event )</code>	

### Circuit examples

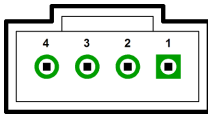
<p>Example for the use of PWM output OP76: LED lighting.</p>	
<p>Controlling a relay by the use of digital output OP90: <i>(Don't use PWM output for controlling a relay.)</i></p>	
<p>Measuring a voltage by the use of voltage input: <i>(e.g. for voltage monitoring)</i></p>	
<p>Controlling and retrieving speed of an AC-ventilator by the use of analog outputs DA0 / DA1 and counter input IP91:</p>	

[< Contents](#)

1) Alternatively, you can directly put a value into the port-register (el.DA\_P76); Range: 0 .. 3000.

## CN8 I<sup>2</sup>C-Bus Port (Ext.)

Connector type	Molex C-Grid III 4-Pol (single row)	Wiring Harness: Molex SL housing 4-Pol
Pin 1	SDA	Serial data line
Pin 2	L5V	Logic +5V
Pin 3	LGND	Logic Ground
Pin 4	SCL	Serial clock line




passendes Kabel: F4404

eigerScript	eVM commands of class <b>I2C</b> :
<code>I2C.Init( )</code>	
<code>I2C.Start( )</code>	
<code>I2C.RepeatedStart( )</code>	
<code>I2C.Stop( )</code>	
<code>I2C.WriteConfigByte( VarInt:SlaveAddress , VarInt:I2C_DATADIRECTION )</code>	
<code>I2C.WriteByte( VarInt:Value )</code>	
<code>I2C.ReadStreamByte( VarInt:Value )</code>	
<code>I2C.ReadLastByte( VarInt:Value )</code>	
<code>I2C.SetMode( VarInt:I2C_MODE )</code>	

[< Contents](#)

## CN9 Power Supply VDC

Connector type	Molex MiniFit jr. 5566 2-Pol	Wiring Harness: Molex MiniFit jr. 5566 housing 2-Pol
Pin 1	LVDC	Logic +8 .. 30VDC
Pin 2	LGND	Logic Ground

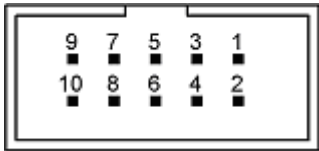


appropriate power supply:  
F4325 (12V) or F4394 (24V)

[< Inhaltsverzeichnis](#)

## CN10 Digital Inputs for 8 Functional Keys (Keyboard)

Connector type	Box Header 10-Pol	Wiring Harness: connector 10pol for ribbon cable
Pin 1	L3V3	Logic +3.3V
Pin 2	SW_A	Switch A
Pin 3	SW_B	Switch B
Pin 4	SW_C	Switch C
Pin 5	SW_D	Switch D
Pin 6	SW_E	Switch E
Pin 7	SW_F	Switch F
Pin 8	SW_G	Switch G
Pin 9	SW_H	Switch H
Pin 10	LGND	Logic Ground



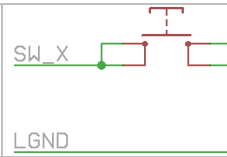
matching ribbon cable: F4406  
(to Keyboard K2050)

eigerScript	eVM commands of class <b>HotKey</b> :
<code>HotKey.DeInstallKey( VarInt:Key )</code>	
<code>HotKey.DisableLocalKeys( )</code>	
<code>HotKey.EnableLocalKeys( )</code>	
<code>HotKey.InputFlush( )</code>	
<code>HotKey.InputSelect( )</code>	
<code>HotKey.Install( b , procName )</code>	
<code>HotKey.InstallLocalKey( VarInt:Key , labelRelative24:Event , VarInt:Tag )</code>	
<code>HotKey.TableDisable( )</code>	
<code>HotKey.TableEnable( )</code>	
<code>HotKey.TableInit( )</code>	

Circuit examples

Using a push-button by the use of a keyboard input:

(SW\_X means: SW\_A .. SW\_H)



Using a switch by the use of a keyboard input

(SW\_X means: SW\_A .. SW\_H)

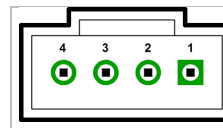


**CN11 Touch Screen Port (Ext.)**

(not populated)

Connector type	AVX 8370 4-Pol	Wiring Harness: Molex SL housing 4-Pol
----------------	----------------	--

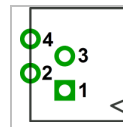
Pin 1	TP_X-	Touchpanel X-direction -
Pin 2	TP_Y+	Touchpanel Y-direction +
Pin 3	TP_X+	Touchpanel X-direction +
Pin 4	TP_Y-	Touchpanel Y-direction -



**CN12 Touch Screen Port**

Connector type	AVX 8370 4-Pol	Wiring Harness: Molex SL housing 4-Pol
----------------	----------------	--

Pin 1	TP_X-	Touchpanel X-direction -
Pin 2	TP_Y-	Touchpanel Y-direction -
Pin 3	TP_X+	Touchpanel X-direction +
Pin 4	TP_Y+	Touchpanel Y-direction +



eigerScript eVM commands of class **HotSpot**:

```
HotSpot.Install( labelRelative24:Event , labelRelative24:Event ,
                labelRelative24:Event , labelRelative24:Event )
HotSpot.DeInstall( )
HotSpot.DeInstallGroup( )
HotSpot.GetNextGroupNumber( )
HotSpot.GetCurrGroupNumber( )
HotSpot.TableDestroy( )
HotSpot.TableDisable( )
HotSpot.TableEnable( )
HotSpot.DisableRegion( )
HotSpot.EnableRegion( )
HotSpot.Disable_BaseGroup( )
HotSpot.Enable_BaseGroup( )
HotSpot.Disable_By_ID( VarInt:HotSpot_ID )
HotSpot.Enable_By_ID( VarInt:HotSpot_ID )
HotSpot.DisableGroup( VarInt:Group )
HotSpot.EnableGroup( VarInt:Group )
HotSpot.GetInfo_By_ID( VarInt:HotSpot_ID )
```

< [Contents](#)

**JTAG Programming Interface for EVE anna (CPLD)**

<b>Connector type</b>		Molex C-Grid III 8-Pol	<b>Wiring Harness:</b> Molex C-Grid III housing 8-Pol	
Pin 1	L3V3	Logic +3.3V		
Pin 2	TDO	Test Data Output		
Pin 3	TDI	Test Data Input		
Pin 4	ISPEN	VSS		
Pin 5	TRST	Test Reset		
Pin 6	TMS	Test Mode Select		
Pin 7	LGND	Logic Ground		
Pin 8	TCK	Test Clock		

**RS2 FOX-COM2 (UART2), Serial Interface RS232**

<b>Connector type</b>		TYCO CST100 5-Pol	<b>Wiring Harness:</b> AMP MTA100 5-Pol	
Pin 1	L3V3	Logic +3.3V		
Pin 2	TXD_2	RS232 Transmit Line		
Pin 3	LGND	Logic Ground		
Pin 4	RX_2	RS232 Receive Line		
Pin 5	L12V	Logic +12V		

matching cables: F4259 , F4339

<b>eigerScript</b>	eVM commands of class <b>Serial</b> :
<code>Serial.Rx_Get_EventChar( VarInt:COM2 , VarInt:Char )</code>	
<code>Serial.Rx_InBuf_Clear( VarInt:COM2 )</code>	
<code>Serial.Rx_InBuf_LastChar( VarInt:COM2 , VarInt&gt;LastChar )</code>	
<code>Serial.Rx_InBuf_Length( VarInt:COM2 , VarInt:Length )</code>	
<code>Serial.Rx_InBuf_to_String_Append( VarInt:COM2 , VarStr:InputString )</code>	
<code>Serial.Rx_Monitor_ClearLocal( VarInt:COM2 )</code>	
<code>Serial.Rx_Monitor_DisableLocal( VarInt:COM2 )</code>	
<code>Serial.Rx_Monitor_EnableLocal( VarInt:COM2 )</code>	
<code>Serial.Rx_Monitor_ScanContinue( VarInt:COM2 )</code>	
<code>Serial.Rx_MonitorSlot_Disable( VarInt:COM2 , VarInt:Slot )</code>	
<code>Serial.Rx_MonitorSlot_Enable( VarInt:COM2 , VarInt:Slot )</code>	
<code>Serial.Rx_MonitorSlot_InstallLocal( VarInt:COM2 , VarInt:Slot , labelRelative24:Event , VarInt:CharLO , VarInt:CharHI )</code>	
<code>Serial.SetBaudrate( VarInt:COM2 , VarInt:Baud_9600 )</code>	
<code>Serial.Tx_Char( VarInt:COM2 , VarInt:Char )</code>	
<code>Serial.Tx_CRLF( VarInt:COM2 )</code>	
<code>Serial.Tx_NUL( VarInt:COM2 )</code>	
<code>Serial.Tx_String( VarInt:COM2 , VarStr:String )</code>	

**Circuit examples**

Using FOX-COM2 (UART2) to connect the eigerPanel to your computer:  
(F4339)

Using FOX-COM2 (UART2) to connect the eigerPanel (as DTE) to another electronic device:  
(F4259)

< [Contents](#)

**Buzzer**

```
eigerScript eVM commands of class Buzzer:
Buzzer.Off( )
Buzzer.On( )
Buzzer.Timer( VarInt:Time )
Buzzer.Alarm( VarInt:ON_Time , VarInt:OFF_Time , VarInt:Pulse ,
              VarInt:Pause , VarInt:Loops )
Buzzer.Click( )
Buzzer.Touch( VarInt:Time )
```

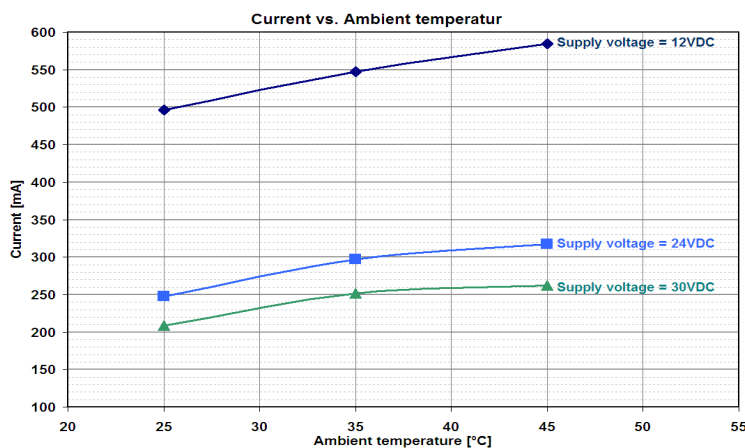
Alternativ kann der Buzzer gleich angesteuert werden wie der digitale Ausgang **OP90** (siehe Seite 10). Der Portname für den Buzzer lautet **OP92**.

**Specifications**

**Energy**

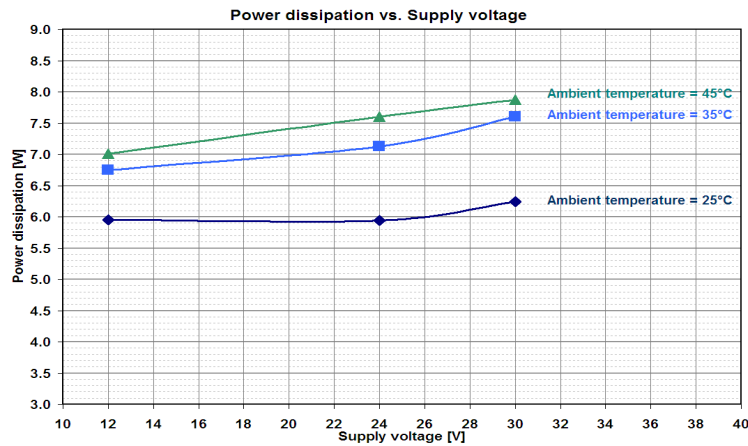
		25°C	35°C	45°C	25°C	35°C	45°C	
		<i>with Display</i>			<i>without Display</i>			
Power consumption	Supply: 12VDC	495.9	547.6	495.9	166.9	215.3	231.9	mADC
	Supply: 24VDC	247.6	297.2	247.6	98.6	146.8	159.4	
	Supply: 30VDC	208.2	251.8	208.2	84.4	133.7	147.1	
Power loss	Supply: 12VDC	5.95	6.57	5.95	2.00	2.58	2.78	W
	Supply: 24VDC	5.94	7.13	5.94	2.37	2.52	3.83	
	Supply: 30VDC	6.24	7.55	6.24	2.53	4.01	4.41	

		MIN	TYP	MAX	
Supply voltage		9	12 / 24	9	VDC
Temperature range	Operation	-10	-	+70	°C
	Storage	-30	-	+80	



**Graph 1:** Power requirements in dependence on the ambient temperature. Measurements were carried out with three different supply voltages. The eigerPanels power consumption increases with increase of ambient consumption.

[< Contents](#)



**Graph 2:** Power loss in dependence on the supply voltage. Measurements were carried out with three different ambient temperatures. The eigerPanels power requirement increases with the increase of the supply voltage.

**Memory**

Type	CompactFlash™ (CFC) / RAM
Useable memory for programming data	According to the capacity of the CFC
Useable RAM (just needed for the byte code of your view; pictures are handled by the two video memories AVR & RVR)	280kB (fix)

**Interfaces**

BUS	serial, RS485 serial, I <sup>2</sup> C	Power supply via BUS possible
UART1	serial, RS232	Reserved for Debugging / Download
UART2	serial, RS232	For your Application
Inputs		
analog	1x potentiometer	voltage range 0 .. 3.3VDC
	2x NTC / potentiometer	voltage range 0 .. 3.3VDC
	1x measurement input	voltage range 0 .. 50VDC
digital	8x function keys	voltage range 0V / 3.3VDC e.g. keyboard or sensors etc.
	1x counter input up to 3kHz	voltage range 0 .. 10VDC
Output		
analog	2x programmable voltage outputs	voltage range 5 .. 30VDC 3 .. 20VAC
digital (PWM)	1x digital power output up to 1A (OP76)	0 .. 1000 → Duty Cycle 0 .. 100% voltage range 9 .. 30VDC (complies with the supply voltage)
digital	1x digital power output up to 1A (OP90)	voltage range 9 .. 30VDC (complies with the supply voltage)

[< Contents](#)

**Further Characteristics**

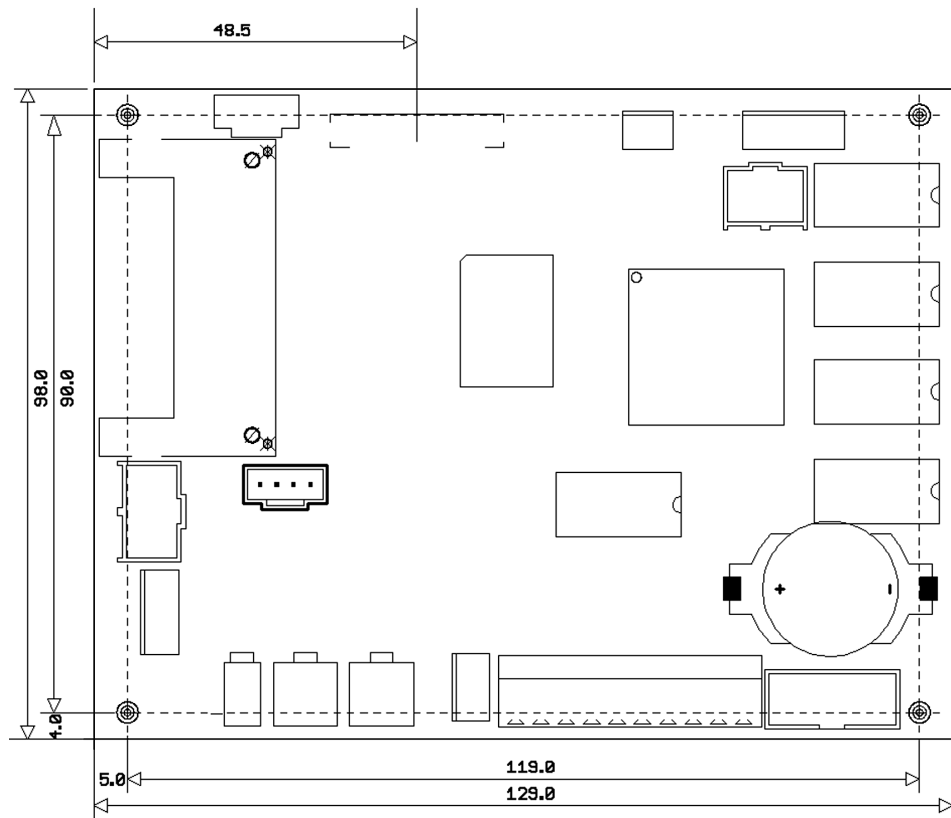
Graphic performance	The FOX embedded computer F57H/F57C can display up to 160 labels per second (buttons, titles, letterings, etc.) and change the view up to 70 times per second.
Videospeicher	Two Video memory (Accessible Video Ram AVR und Refresh Video Ram RVR)
Clock/Date	Battery buffered real time clock RTC
Battery for RTC	Li-battery CR2032, 3V, 235mAh, 10 years power reserve
maximum warming	Temperature heat sink 60°C (at 30°C room temperatur e)
acoustic signal generator	Buzzer (OP92)
Boot time	4 seconds

**LCD Connections**

LCD type and resolution	TFT LCD, WVGA 800 x 480 pixels
Power supply for the LCD display	No separate power supply necessary; the power supply is provided by the LCD connection (CN1).
Backlight voltage	5VDC
Touch Screen	4 wire, resistive

[< Contents](#)

**Dimensions**



maximale Printheöhe: 16.5mm (MiniFit jr. 4pol)

Max height of the board: 16.5 mm

Hole diameter: 3.2 mm

Image 2: Layout and dimensions of the circuit board

[< Contents](#)



## Dimensions of Housing and Synthetic Frame

Links where you find the exact dimensions:

- Housing of the eigerPanel70C:  
[http://www.eigergraphics.com/Download/docs/K2006\\_EP70C\\_mit\\_Gehaeuse\\_und\\_Kunststoffrahmen.pdf](http://www.eigergraphics.com/Download/docs/K2006_EP70C_mit_Gehaeuse_und_Kunststoffrahmen.pdf)
- STEP-Model (CAD) of the housing of the eigerPanel70C:  
[http://www.eigergraphics.com/Download/eP\\_CAD/EP70C-STEP.zip](http://www.eigergraphics.com/Download/eP_CAD/EP70C-STEP.zip)
- Opening for the eigerPanel (e.g. in the wall or in the housing of your machine):  
[http://www.eigergraphics.com/Download/docs/K2006\\_eP70\\_Oeffnung\\_fuer\\_Gehaeuse.pdf](http://www.eigergraphics.com/Download/docs/K2006_eP70_Oeffnung_fuer_Gehaeuse.pdf)
- Display:  
[http://www.eigergraphics.com/touchscreens.htm#DataImage\\_Display70\\_WVGA](http://www.eigergraphics.com/touchscreens.htm#DataImage_Display70_WVGA)

## Firm- and Software

Graphic Controller, eigerVideo Engine	EVE anna V1.30
Firmware	
Firmware Update	<a href="http://www.eigergraphics.com/download.htm">http://www.eigergraphics.com/download.htm</a>
Software, Application Notes programmer Infos for the ports	<a href="http://www.eigergraphics.com/download.htm">http://www.eigergraphics.com/download.htm</a>

## Further Information, Contact Data, Support

Phone	+ 41 41 754 50 10
Fax	+ 41 41 754 50 19
Web	<a href="http://www.eigergraphics.com/">http://www.eigergraphics.com/</a>
Email	<a href="mailto:eiger@s-tec.ch">eiger@s-tec.ch</a>

## Datasheet Version

Version	11 <sup>th</sup> December 2009
Update	30 June 2011
Author	Samuel Nussbaumer, S-TEC electronics AG, Switzerland

< [Contents](#)

---

## Revision-History

---

24<sup>th</sup> November 2010 SN - Name of CN2.6 corrected (S-PROG20-Interface)  
old: LGND "Logic Ground"  
new: L3V3 "Logic +3.3V"

[Page 5](#)





**FOX** embedded computers 

*the canny swiss solution*

© Copyright 2010 S-TEC electronics AG, Inc. Alle Rechte vorbehalten



**S-TEC electronics AG**

Gewerbestrasse 6 • CH-6314 Unterägeri • Telefon (+41) 41 / 754 50 10 • Telefax (+41) 41 / 754 50 19  
info@s-tec.ch • www.s-tec.ch