Citron®
Creative Industrial
Electronic Systems

CITRON 3rd Generation Infrared Touchscreens

IRT65-V3.2
IRT84-V2.x
IRT104-V5.x
IRT121-V3.x
IRT151-V2.x
IRT170-V1.0
IRT181-V1.1
IRT190-V1.0

User's Manual

Document revision

Rev.	Description	Reviser	Date
001	First edition	tt	1998-08-30
002	Additional drawings for PC interconnection and IRT disable switch	tt	1998-11-04
003	Touch and bezel drawings included	pk	1999-07-12
004	Revised for 3rd IRT Generation	wh	2004-11-12
005	Release Version created	tt	2005-03-22
006	Glass specification added	tt	2007-06-27
007	Latest touch hardware revisions added	tt	2015-09-24

Exclusion of liability

The contents of this manual serve for information purposes only. Citron GmbH reserves the right to change the contents of this manual without prior notice. While reasonable efforts have been made in the preparation of this manual to assure its accuracy, errors may occur. Therefore, Citron GmbH assumes no liability resulting from errors or omissions in this manual or from the use of the information contained herein.
Citron GmbH appreciates suggestions with regard to improvements or corrections.

Tel. ++ 49 821 74945-0 FAX ++ 49 821 74945-99
e-mail: info@citron.de
http:\\www.citron.de

Document information

File name: h:\manuals\irt3g\umirt3g_e_r007.doc
Date: 21/09/2015 06:29:00
Document revision: 2
Document reference: \\ntserv1\dokument\dformat.dot\cidoku.dot

Table of contents

1 Introduction

The following brief introduction provides general information on how to use this manual and subsequently outlines the hard- and software of a Citron Infrared Touch.

1.1 Conventions

The following table lists the typographic conventions that are used in this manual and explains their respective meaning.

Convention	Meaning
bold	Data that are to be received or that are transmitted by the IRT. It is essential that the syntax of the received or transmitted data precisely corresponds to the syntax indicated that way.
Italics	Space-holder or variable. The actual value is to be entered by the user.
SEMI-CAPITAL LETTERS	Represents commands, messages, reports or constants
`monospace`	Example program or example byte-order
[ ]	Embraces optional parameters
|	Separates either / or alternatives
0x00	Just like in 'C', the notation with the preceding '0x' is used for hexadecimal numbers.
…	A variable number of additional data follows.

Within the subsequent example programs the following variable types and constants are used:

```
typedef  unsigned char   BYTE;     /* unsigned 8 bit value                  */
typedef  unsigned short  WORD;     /* unsigned 16 bit value                 */
typedef  unsigned long   DWORD;    /* unsigned 32 bit value                 */
typedef  int             BOOL;     /* Boolean variable (true or false)      */

#define  FALSE    0                /* False value for Boolean variable      */
#define  TRUE     (!FALSE)         /* True value for Boolean variable       */
```

1.2 General

Citron Touch Systems represent the most direct and natural way of communication between man and machine.
Meanwhile our Touch Systems are successfully employed in the fields of automation, telecommunication, medical and automobile engineering.
The remarkable acceptance of the Citron Touch Systems results from the high reliability of the product, the consequent adoption of customer requirements into the system's layout, the continued product development and the flexibility regarding customer-specified system designs as well as software support.
Although Citron Touch Systems were specifically developed for flat displays, they may also be employed for monitors with either spherical or cylindrical tubes.
Our strive for comprehensive system solutions required numerous experiments with filter glass as well as various constructions of mounting frames. Due to this experimental phase we are now able to support our customers with competent advice regarding the right choice of filter glass, may it be synthetic material or glass, etched, coated, laminated or with wire-mesh inlays.
The mounting frame (bezel) generally consists of PMMA material and is manufactured with a hot injection moulding process. This process guarantees best optical performance and dimensional accuracy.
Citron has developed special methods for glueing synthetic filters and glass filters in order to achieve protection levels of IP65 or better and to withstand temperature cycles from -20°C to +85°C without cracking of the glass.
Due to the integration of most different requirements from various innovative industrial applications, a Touch System could be developed that is not only highly practise-oriented but at present represents the state-of-the-art optimum with regard to compact design, installation-friendliness, programmability and value for money.

1.3 Hardware outline

The optimum design of the multi-layer board results in a very low susceptibility to interferences that, for example, occur at a direct installation on plasma displays.
The various Touch types mainly differ in their corresponding display size and the way they are installed. All Citron Infrared Touch products comprise the following features:

- Touch detection by means of a matrix of infrared beams (X-axis, Y-axis)
- Detection of the pressure intensity exerted onto the front screen (Z-axis, optional)
- Evaluation electronics together with the beam matrix arranged on one PCB
- Single +5 V_{DC} power supply
- Serial interface with RS232 signal levels
- One programmable PWM output, for example for controlling the backlight inverter of TFT displays
- All input and output ports are galvanically isolated via optocoupler
- Optional audio amplifier and sound source for loudspeaker
- Optional USB interface
- Optional 3D feature making touch pressure sensitive for save operations
- Hard- and software compatible to 2nd Generation Touchscreens

1.4 Software outline

3rd Generation Touchscreens continues the philosophy giving the user the possibility to utilise the full scope of performance of the IRT hardware through an Open Software Interface, called Citron Touch Software, in short CTS. 3rd Generation CTS is fully compatible to it's predecessors, so software written for previous IRT versions will still work without change.

- Efficient and reliable transmission protocol
- Extended coordinates resolution
- Division of the Touch Zone in user-defineable polygonal shaped Areas
- Flexible pre-processing of the touch events
- Extensive error correction for a reliable detection of beam interruptions
- Tracing of two touch spots
- Extended features to influence the behaviour of the IRT
- Two Power Saving Modes to decrease the power consumtion of the IRT and to increase the life expectancy of the beams
- Complete emulation of the Carroll Touch® SmartFrame™ protocol

The items mentioned above are comprehensively explained in the following chapters.

2 Function of the IRT

The following chapters contain comprehensive descriptions about both the functional components of the IRT and their use.

2.1 Initialization

Prior to using the IRT, it needs to be initialized by the host computer beforehand. When initialized, the IRT automatically recognizes the baud rate, the parity as well as the transmission protocol.

2.1.1 Automatic baud rate recognition

During the initialization, the IRT is capable of automatically recognizing the following baud rates with either even, odd or without parity:

1200, 1800, 2400, 3600, 4800, 7200, 9600, 14400, 19200, 38400 and 57600 bps (bits per second)

As long as the IRT is not yet initialized, it emits BREAK signals[1] with a mark-to-space ratio of 90 ms to 10 ms. In case an already initialized IRT is to be reset, a BREAK signal of at least 50 ms time span needs to be applied to the RxD input of the IRT. As soon as the BREAK signal clears at the RxD input, the IRT resumes emitting BREAK signals.
In order for the baud rate to be recognized, the host computer needs to transmit at least one Carriage Return signal (<CR>, 0x0D). As soon as the baud rate is recognized, the IRT stops emitting BREAK signals. In order for the parity and the desired transmission protocol to be recognized, one of the following data bytes has now to be transmitted to the IRT:

Data byte	Used protocol
0x3C	Carroll Touch emulation
0x81	Citron Touch Software

A delay interval of at least 50 ms is required after each character. The acknowledgement of the initialization depends on the selected transmission protocol:

Used protocol	Acknowledgement
Carroll Touch emulation	None
Citron Touch Software	XON (0x11)

[1] A BREAK signal is created by applying Low level (0 V) to the serial input of a SIO for longer than 1 ½ character lengths. The character length is determined by the set transmission rate of the SIO. At the inverted RS232 outputs, however, a BREAK signal equals +12 V!

The 'C' program fragments listed below initializes the IRT in the CTS transmission protocol using the following external functions:

```
extern BYTE ReadComm();              /* reads a character from the serial interface     */
extern void WriteComm(BYTE x);       /* writes a character to the serial interface      */
extern void Delay(WORD n);           /* waits n milliseconds                            */
extern DWORD GetTick();              /* System time in milliseconds                     */
```

Initializing the CTS protocol:

```
#define  CR           0x0D
#define  XON          0x11
#define  Reset_CTS    0x81

BOOL InitCTS(void)
{
    int      i;
    DWORD    time_out;

    /* assume the IRT is ready for connection */
    WriteComm(CR);     /* emit first CR  */
    Delay(50);         /* wait 50 milliseconds                                        */
    WriteComm(Reset_CTS); /* select protocol                                          */
    time_out = GetTick() + 1000;  /* wait max. 1 second for an answer                 */
    while (GetTick() < time_out)
    {
         if (ReadComm() == XON)
         return TRUE; /* report success */
    }
    return FALSE;      /* report error   */
}
```

2.1.2 Emulation of other protocols

The emulation of Carroll Touch protocol is implemented in the firmware of the IRT. The emulation mainly serve the purpose to enable the further use of already existing programs with the IRT. New developments, however, should preferably operate on the CTS protocol described below. Since the coordinates origin of the original Carroll-Touch may differ from the one of the IRT, the Carroll protocol have been extended by the following commands in the firmware:

Name	Code	ASCII	Description
OriginTopLeft	0x77	w	Sets origin of coordinates to the top left-hand side corner of IRT
OriginBottomLeft	0x78	x	Sets origin of coordinates to the bottom left-hand side corner of IRT
OriginTopRight	0x79	y	Sets origin of coordinates to the top right-hand side corner of IRT
OriginBottomRight	0x7A	z	Sets origin of coordinates to the bottom right-hand side corner of IRT

In this particular mode the commands and the behaviour of the IRT correspond to the ones of a Carroll Smart-Frame™ as described in the manual "Infrared Smart-Frame Programmer's Guide" from 1987. The following particularities are to be regarded:

- The blank-out time for continuously interrupted beams is preset to 10 seconds.
- The time interval between two coordinate messages in the Continuous Mode equals 55 ms.
- The IRT responds to multiple touching with its maximum speed.
- Since the IRT is not equipped with the necessary connections, a control of the serial data flow by means of the RTS/CTS protocol is not possible. Therefore the commands HARDWAREFLOWCONTROLON (0x41) and HARDWAREFLOWCONTROLOFF (0x42) are disregarded.

2.2 Communication

The IRT communicates with a host computer via an asynchronous serial interface using a software transmission protocol as described in the following chapters. By maintaining the highest possible degree of flexibility, this protocol ensures that the beginning and the end of either a command, a message or a report can be detected. This way possible errors during data transmission can easily be detected.
In the following chapters a command, a message or a report is commonly referred to as data packet.

2.2.1 Encoding

At the CTS protocol a data packet always contains the following structure:

DC2 (0x12)	Identification (0x17..0xFF)	Parameter	DC4 (0x14)

Each data packet is framed by a DC2 code (0x12) representing the start byte and a DC4 code (0x14) representing the stop byte. Right after the start byte the identification of the data packet follows. The following range of values applies to the CTS protocol:

Type of data packet	**Range of values**
Command	[0x80 .. 0xFF]
Message	[0x17 .. 0x1F]
Report	[0x20 .. 0x7F]

Generally the identification of a report accords to the identification of the corresponding command, however with deleted bit 7. The identification of the report AREADEFINITION, for example, is 0x22. Consequently, the identification of the corresponding command GETAREADEF is 0xA2.
All values between 0x00 and 0xFF are valid parameter values. In order to prevent misinterpretations for the exceptional case of a parameter value being identical with a control character, the parameters are transmitted encoded, too. For that purpose a value range for control characters is defined. At the protocol these values range from 0x10 up to 0x16. In case of a parameter value equalling a value within this range, the parameter value is ORed bit by bit with the value 0x40 and a SYN code (0x16) is prefixed. Consequently, the byte order

```
0x00 0x05 0x12 0x80 0x14 0x44
```

would be transmitted encoded as follows:

```
0x00 0x05 0x16 0x52 0x80 0x16 0x54 0x44
```

In order to decode this parameter value, all there is to do is to bitwise AND the next value following a received SYN code bit by bit with the value 0xBF.

The following 'C' function encodes a command and transmits it to the IRT.

```
#define  DC2          0x12
#define  DC4          0x14
#define  SYN          0x16
#define  ENCODE       0x40
#define  FIRST_CONTROL 0x10
#define  LAST_CONTROL 0x16

extern void WriteComm(BYTE x);      /* writes a character to the serial interface     */

/*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*/
/* [SendCommand]                                                                    */
/*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*/
/* Encodes and transits a command to the IRT.                                       */
/*                                                                                  */
/* Parameters:                                                                      */
/*  BYTE     cmd      Identification of the command                                 */
/*  BYTE     *pData   Pointer to the command parameters                             */
/*  WORD     len      Size of command parameters in bytes                           */
/*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*/

void SendCommand(BYTE cmd, BYTE *pData, WORD len)
{
    BYTE data;

    WriteComm(DC2);        /* transmit start byte                                   */
    WriteComm(cmd);        /* transmit command identification                       */
    while (len-- > 0)
    {    /* encode and transmit command parameters                                  */
         data = *pData++;
         if (data >= FIRST_CONTROL && data <= LAST_CONTROL)
         {    /* Parameter byte has to be encoded                                   */
              data |= ENCODE;
              WriteComm(SYN); /* mark encoding                                      */
         }
         WriteComm(data);  /* transmit parameter byte                               */
    }
    WriteComm(DC4);        /* transmit stop byte                                    */
}
```

The following 'C' function receives and decodes either a message or a report from the IRT.

```
#define  DC2          0x12
#define  DC4          0x14
#define  SYN          0x16
#define  ENCODE       0x40

extern BYTE ReadComm();      /* reads a character from the serial interface         */
extern DWORD GetTick();      /* System time in milliseconds                         */

/*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*/
/* [ReceiveMessage]                                                                 */
/*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*/
/* receives and decodes a message or a report from the IRT.                         */
/*                                                                                  */
/* Parameters:                                                                      */
/*   BYTE     *id      here the identification of the message or report is filed    */
/*   BYTE     *pData   pointer onto a buffer for the parameter bytes               */
/*   WORD     *len     number of received parameter bytes                          */
/*   DWORD    time_out max. permitted time span for the reception of the message   */
/*                     or the report.                                               */
/*                                                                                  */
/* return value:                                                                    */
/*   FALSE, in case of no reception within the max. permitted time span           */
/*   TRUE, in case of correct reception within the max. permitted time span       */
/*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*/
BOOL ReceiveMessage(BYTE *id, BYTE *pData, WORD *len, DWORD time_out)
{
    BYTE data;
    BOOL bDecode;

    *len = 0;               /* Assumption: no parameter bytes                       */
    time_out += GetTick();/* Calculate waiting time                                 */
    while (GetTick() < time_out)
    {
        if (ReadComm() == DC2)
        {   /* start byte received, now get identification                         */
            *id = ReadComm();
            bDecode = FALSE;
            while (GetTick() < time_out)
            {   /* read parameter bytes                                            */
                data = ReadComm();
                if (data == DC4)
                    return TRUE; /* report success                                 */
                if (data == SYN)
                    bDecode = TRUE;
                else
                {   /* normal parameter byte received                             */
                    if (bDecode)
                    {   /* Parameter byte has to be decoded                        */
                        data &= ~ENCODE;
                        bDecode = FALSE;
                    }
                    *pData++ = data;
                    (*len)++;
                }
            }
        }
    }
    return FALSE;           /* report error                                         */
}
```

2.2.2 Flow control

Apart from the DC2, DC4 and SYN control characters, there are also XON (0x11), XOFF (0x13) and NAK (0x15) permitted outside a command sequence. If the IRT receives a XOFF (0x13), it concludes the data transmission to the host computer. As soon as it receives a XON (0x11) again, it will resume the data transmission. In addition to that a time span can be set in which the XON signal has to be transmitted to the IRT. If within this „XOFF-Timeout“ span the IRT fails to receive the XON, it automatically resumes transmitting.

In opposite direction the IRT transmits an XOFF if either its internal buffer threatens to overflow or if it is unable to reply to commands for a prolonged period. As soon as the IRT is ready again to receive data, it transmits a XON. Prior to a reset the IRT always transmits a NAK.

Summary:

Direction of transmission	Data	Value	Meaning
Host computer ⇒ IRT	XOFF	0x13	Stops transmission of messages and reports.
	XON	0x11	Resumes transmission of messages and reports.
IRT ⇒ Host computer	XOFF	0x13	Stops the transmission of commands. A command transmission in progress will first be finished.
	XON	0x11	Resumes transmission of commands.

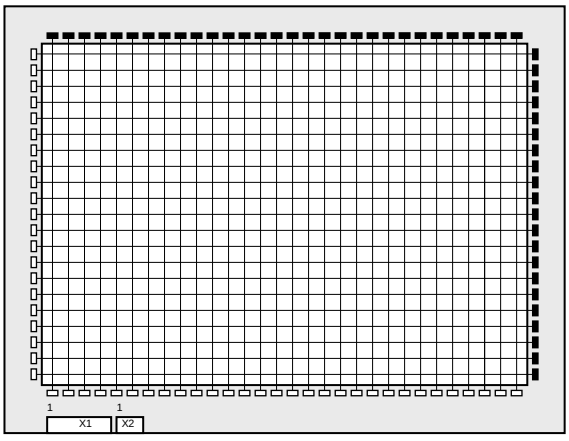
Whether or not the flow control is utilized can be set separately for transmitting and receiving by means of the command SETTRANSMISSION (0xCF). The currently used flow control can be read out from the report TRANSMISSION (0x47) by means of the command GETTRANSMISSION (0xC7).

2.3 Touch detection

Chiefly the IRT serves for the detection of the position of a touch made on a screen. The way this happens and what preconditions characterize a valid touch is described in the following chapters.

2.3.1 Basics

The IRT contains infrared beams that are arranged like an X- and Y-matrix around a frame-shaped printed-circuit board (refer to the illustration below). This beam matrix is constantly scanned by the IRT's microcontroller. By touching the screen, for example with a finger tip, a pair of beams is interrupted and that way the exact position of this touch can be detected. Beams used to detect the X-coordinates are called "X-beams", those for the Y-coordinates "Y-beams", respectively. The area within the beam matrix is called "Touch Zone"



In order to prevent malfunctions, both the minimum and the width duration of a valid interruption are limited. This way either too small subjects, for example a fly, or too big subjects, for example a necktie hanging into the Touch Zone cannot result in a coordinate message. After the initialization the minimum width of an interruption is set to the width of one beam. In that case the maximum width equals the width of 5 beams next to each other. By means of the command SETBEAMMINMAX (0xC8) these values can be adjusted for the X- and Y- direction separately. The present settings can be read out from the report BEAMMINMAX (0x40) which is called up by means of the command GETBEAMMINMAX (0xC0).

The command SETTOUCHTIME (0xD1) provides an additional safety feature: in order for the touch to be transmitted to the host computer, it has to last for the entire touch time span set by means of that command.

The currently set touch time span can be read out from the report TOUCHTIME (0x50) which is called up by means of the command GETTOUCHTIME (0xD0).

2.3.2 Multiple touching

Note: In the following this manual uses the terms „dual touching“ and „multiple touching“. These terms differ in their conceptual meaning.
The IRT allows the interruption of the Touch Zone at two different spots at the same time. It detects these two interruptions and interprets them as valid. This process is called „dual touching“.
If more than two spots of the Touch Zone are interrupted at the same time, the IRT interprets that as an invalid „multiple touching“ and issues an error message. However, this particular feature can be disabled (see SETDUALTOUCHING on page 35). In this particular case, the IRT interprets already double touching as invalid and issues an error message.

Summary: „Double touching“ represents a valid touching of two spots at the same time.
„Multiple touching“ represents an invalid touching of more than either one or two spots at the same time (depending on the Touch programming).

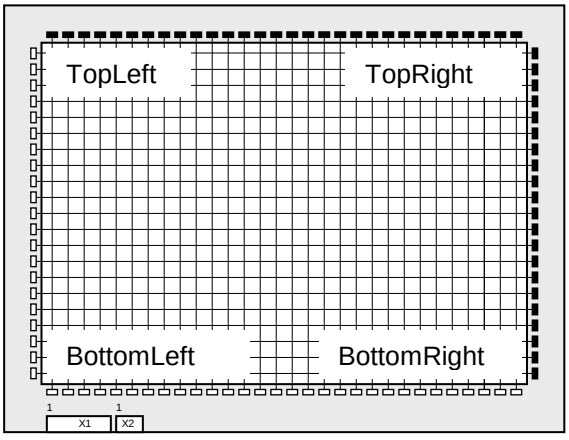
As mentioned earlier, the IRT allows the interruption of the Touch Zone at two different spots at the same time. However, for this particular case the following limitations have to be regarded:

- The two touch events have to occur successively, i.e. one after the other
- Only the second touch spot is allowed to move while the first one should remain stationary. Slow movements, however, are also allowed for the first touch spot.

Despite these limitations, dual touching can be utilized meaningful. That way, for instance, it is possible to realize the SHIFT, CTRL or ALT keys of a virtual keyboard.
In order to enable dual touching, the according behaviour has to be set by means of the command SETDUALTOUCHING (0xCB) beforehand. Furthermore this command determines whether an error message DUALTOUCHERROR (0x18) is issued at multiple touching. The present settings can be read out from the report DUALTOUCHING (0x43) which is called up by means of the command GETDUALTOUCHING (0xC3).

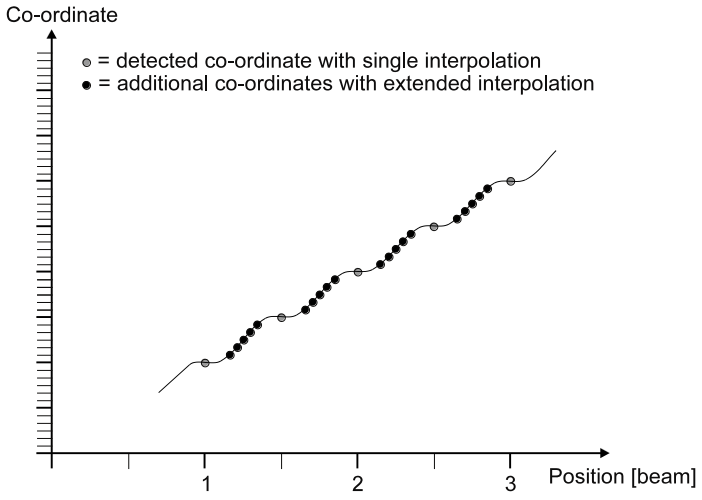
2.4 Coordinates system

To determine the touch spot position the IRT uses a Cartesian coordinates system. The minimum coordinates equal (0,0), the maximum coordinates can be set within the range from 1 up to 65535 by means of the command SETRESOLUTION (0xCD).
By means of the command SETORIGIN-Befehl (0xCC) the coordinates origin can be placed to any of the four Touch corners. The currently set coordinates origin can be read out from the report ORIGIN (0x44) by means of the command GETORIGIN (0xC4). After the initialization the coordinates origin is preset to the top left-hand side corner of the IRT (when viewed onto the components' side of the IRT). The following illustration one more time points out the four Touch corners:



2.5 Extended coordinates resolution

Primarily the resolution of the IRT is determined by the number of beams. These beams are arranged next to each other with a distance of 5 mm. For several beams being interrupted at the same time, the position's resolution is increased to 2.5 mm by the means of single interpolation. In addition to that, at the IRT the signal levels of the adjoining beams are also implemented into the interpolation. That way even within the absolute resolution of 2.5 mm additional positions can be determined. At a 12" TFT display with a resolution of 640 × 480 pixels, for instance, each pixel can be individually driven. This, however, results in a non-linear relation between the position of the interruption and the detected coordinate.
The following diagram illustrates the typical dependence of the detected coordinate on the actual position of the interruption.
However, in order to maintain it clearly arranged, the diagram shows considerably less additional coordinates than the number that can actually be achieved.



2.6 3D-Feature

The IRT can optionally be equipped with force sensors. These sensors sense the force of pressure onto the screen. That way, in addition to the X/Y-position the force of pressure is gained as the third dimension or the Z-axis, respectively.
Whether or not force sensors are available can be read out from the report HARDWARE (0x31) by means of the command GETHARDWARE (0xB1).
Compared to the evaluation of only the X/Y-position, adding the pressure information grants many advantages. For example, by raising the pressure sensitivity of safety-relevant operating components, these components can be protected from being wrongly or unintentionally operated. Furthermore, the use of the IRT as a mouse-substitute, for instance in conjunction with the Citron Mouse Emulation Drivers, becomes far more comfortable since mouse clicks can now be simulated just by increasing the exerted pressure onto the screen.
There are several ways the IRT subsequently carries on working with the pressure information. Firstly, a PRESSURE message (0x1B) can be created whenever a pressure value either exceeds or falls below a predetermined value. Secondly, both the COORD message (0x19) when interrupting an Area and the EXIT message (0x1A) when exiting an Area can be linked with the case of the pressure either exceeding or falling below a predetermined limit value. The way the IRT acts can be individually set for each Area. The Areas are described in the following chapter.

2.7 Areas

In the CTS protocol the Touch Zone is not considered one coherent unit but it can be divided in several user-defined rectangles or polygones. Each one of these shapes is called an "Area". The following chapters explain all the possibilities resulting from this concept.

2.7.1 Basics

There are two different kinds of Areas: Simple Areas with a rectangular shape and polygonal Areas with an arbitrary shape of up to 64 polygone edges. Areas can be defined, changed, cleared, enabled and disabled. In order to simplify and speed up the change between the various configurations, the Areas can be placed on up to 71 so-called Area Pages.

2.7.1.1 Area definition

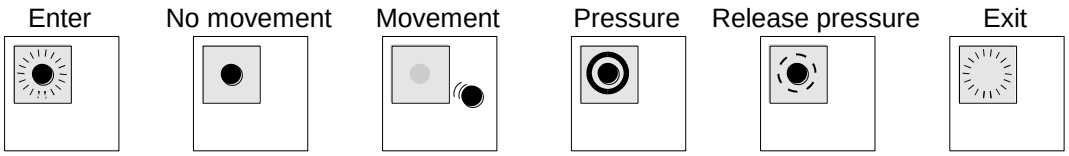
A rectangular Area is newly defined by means of the command DEFINEAREA (0xA1). A polygonal Area is newly defined by means of the command DEFPOLYAREA (0xAA). Each Area is assigned a unique number representing the identification.
If by means of the command DEFINEAREA or DEFPOLYAREA the number of an already existing Area is to be defined, the definition of this already existing Area will be changed. The parameters of an already existing Area can be read out from the report AREADEFINITION (0x22) or POLYAREADEF (0x2A) by means of the command GETAREADEF (0xA2). The command CLEARAREA (0xA0) clears either all Areas, whole Area Pages or just single Areas. The so released memory can then be used again for the definition of new Areas.
The rectangle coordinates that were passed over at the definition set the position and size of the Area within the Touch zone. The reported coordinates values always range within the limits determined by the command SETRESOLUTION (0xCD). Each single Area can be assigned an own Operating Mode with an individual set of Modifiers. The several Operating Modes and the Modifiers are subsequently described in this chapter. Furthermore for each single Area an individual pressure sensitivity can be set.
If just the Operating Mode, the Modifiers or the pressure sensitivity of an Area is to be changed, this can easily be carried out by means of the command SETAREASTATE (0xA6).

2.7.1.2 Area Pages

The Areas can be freely placed on various "Pages". This way it is easy to quickly switch between various Area configurations. Areas on different Pages may be assigned the same number. There is a total of 16 Pages available. The active Area Page is determined by means of the command SELECTAREAPAGE (0xA5). Which Page is currently active can be read out from the report AREAPAGE (0x23) by means of the command GETAREAPAGE (0xA3).
The number of Areas per Page is not fixed. For example, if there is a total of 100 Areas available, it is possible to define 10 Areas on 10 Pages or 99 Areas on 1 Page plus 1 Area on the other Page. How many Areas can be defined at a given point of time can be read out from the report FREEAREASPACE (0x24) by means of the command GETFREEAREASPACE (0xA4).

2.7.2 Area behaviour

The behaviour of an Area at an interruption of the Touch Zone within the Area rectangle is determined by the subsequently described Area Operating Mode and the corresponding Modifiers.
The following symbols are used for the graphical illustration of the Area behaviour:



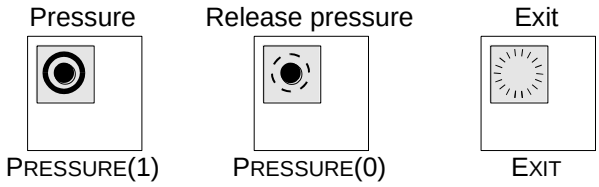
The white rectangle represents the entire Touch Zone. The grey rectangle symbolises an active Area. The touch spot is represented by the black dot.

2.7.2.1 Operating Mode

The Operating Mode of an Area determines the conditions for the touch of an Area to be reported. Different Areas that are on the same Area Page may work in different Operating Modes at the same time.

2.7.2.1.1 AOM_OFF

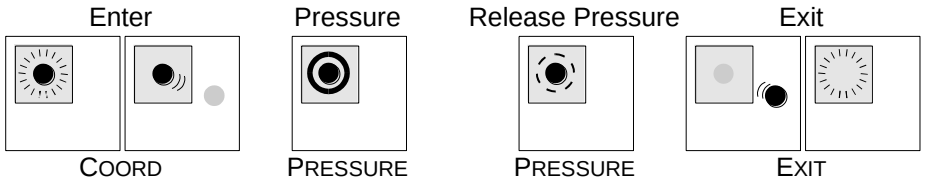
In the Operating Mode AOM_OFF (Area Operating Mode: OFF, **0x00**) no COORD messages (0x19) are reported at a touch of this Area. In this Operating Mode only the events illustrated below result in a message:



EXIT (0x1A) or PRESSURE messages (0x1B) are only created if the corresponding Modifiers have been declared during the Area definition.

2.7.2.1.2 AOM_ENTER

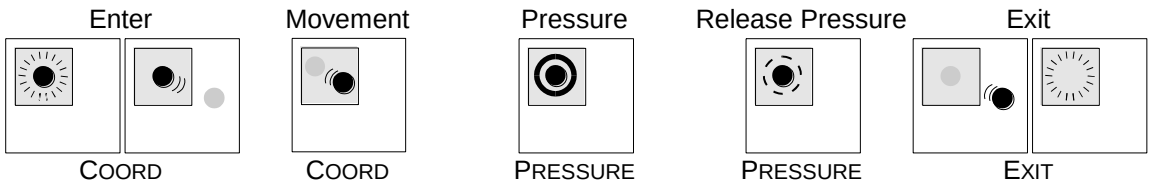
In the Operating Mode AOM_ENTER (Area Operating Mode: ENTER, **0x01**) only the first touch of the Area will report a COORD message (0x19). Further movements within this Area will not result in new COORD messages.
The following illustration shows the events that will create a message in the Operating Mode AOM_ENTER:



EXIT- (0x1A) or PRESSURE messages (0x1B) are only created if the corresponding Modifiers have been declared during the Area definition.

2.7.2.1.3 AOM_TRACK

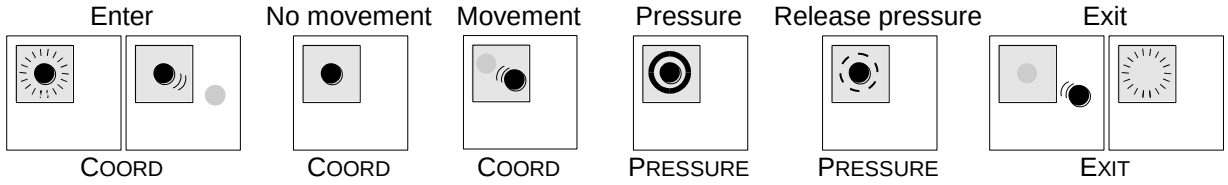
In the Operating Mode AOM_TRACK (Area Operating Mode: TRACKING, **0x02**) the first touch and all further movements within the Area will report a COORD message (0x19).
The following illustration shows the events that will create a message in the Operating Mode AOM_TRACK:



EXIT- (0x1A) or PRESSURE messages (0x1B) are only created if the corresponding Modifiers have been declared during the Area definition.

2.7.2.1.4 AOM_CONT

In the Operating Mode AOM_CONT (Area Operating Mode: CONTINUOUS, **0x03**) a COORD message (0x19) is reported at the first touch and afterwards in regular time intervals as long as the touch spot remains within the Area rectangle. The time interval between the COORD messages is set by the command SETCONTTIME (0xCA). The time interval can be read out from the report CONTTIME (0x42) by means of the command GETCONTTIME-Befehl (0xC2)
The following illustration shows the events that will create a message in the Operating Mode AOM_CONT:



EXIT- (0x1A) or PRESSURE messages (0x1B) are only created if the corresponding Modifiers have been declared during the Area definition.

2.7.2.2 Modifiers

For each Operating Mode there is a set of so-called Modifiers. They determine the exact behaviour for each individual Operating Mode. These Modifiers can be bitwise ORed during the Area definition.

2.7.2.2.1 AOF_ADDEXIT

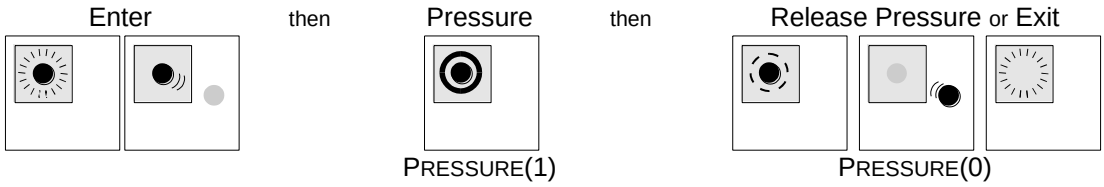
If the AOF_ADDEXIT Modifier (Area Operating Flag: ADD EXIT, **0x01**) was declared during the Area definition, an EXIT message (0x1A) is created as soon as the Touch Zone within the Area rectangle is not interrupted anymore.
The following sequence of events will create an Exit message:

2.7.2.2.2 AOF_ADDCOORD

Only if the AOF_ADDCOORD Modifier (Area Operating Flag: ADD COORDINATES, **0x02**) was declared during the Area definition, the COORD (0x19) and an EXIT message (0x1A) contain coordinates. In case this Modifier was not declared, there are only messages reported when the Area is either touched or released. Without this particular AOF_ADDCOORD Modifier the actual touch position within the Touch zone is not evident.

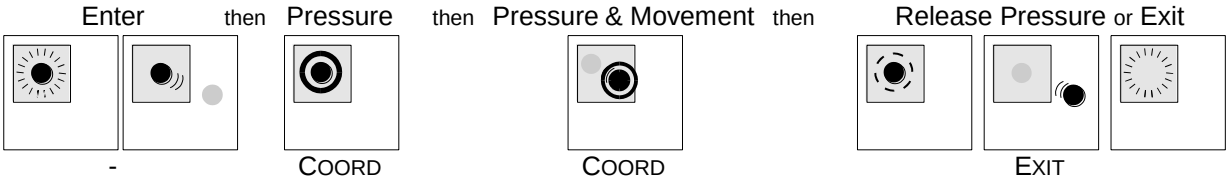
2.7.2.2.3 AOF_ADDPRESS

If the AOF_ADDPRESS Modifier (Area Operating Flag: ADD PRESSURE, **0x04**) was declared during the Area definition, a PRESSURE message (0x1B) is created as soon as both the pressure exerted onto the display screen either exceeds or falls below the Area-specific pressure limit value and a valid touch within the Area rectangle is detected at the same time.
The following sequence of events will create a PRESSURE message:



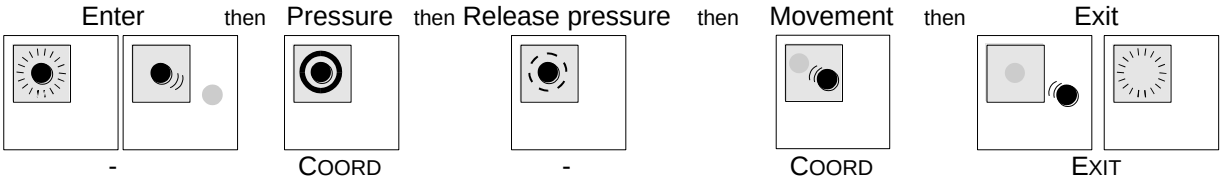
2.7.2.2.4 AOF_PRESSALWAYS

If the AOF_PRESSALWAYS Modifier (Area Operating Flag: PRESSURE ALWAYS, **0x08**) was declared during the Area definition, the respective Area is considered interrupted only if both the pressure exerted onto the display screen exceeds the Area-specific pressure limit value and a valid touch within the Area rectangle is detected at the same time.
As soon as the pressure either falls below the limit value or the touch spot exits the Area rectangle, the respective Area is not considered interrupted anymore. Consequently, according to the Area Operating Mode the COORD message (0x19) is only created as long as the pressure limit value is exceeded. In case the AOF_ADDEXIT Modifier (**0x01**) was additionally declared, a EXIT message (0x1A) is created as soon as the pressure falls below the limit value.
The following illustration shows a typical sequence of events with the AOF_PRESSALWAYS Modifier in the AOM_TRACK Operating Mode:



2.7.2.2.5 AOF_PRESSENTER

If the AOF_PRESSENTER Modifier (Area Operating Flag: PRESSURE ENTER, **0x10**) was declared during the Area definition, the respective Area is considered interrupted if both the pressure exerted onto the display screen exceeds the Area-specific pressure limit value once and a valid touch within the Area rectangle is detected at the same time. Regardless of the pressure exerted onto the display screen, the respective Area is only considered not interrupted anymore when the touch spot exits the Area rectangle. Consequently, according to the Area Operating Mode the first COORD message (0x19) is only created if the pressure limit value is exceeded. If the current Area Operating Mode allows additional COORD messages (0x19) to be reported, they require no further pressure. Only after the Area rectangle was released, for another interruption further pressure is required.
The following illustration shows a typical sequence of events with the AOF_PRESSENTER Modifier in the AOM_TRACK Operating Mode:

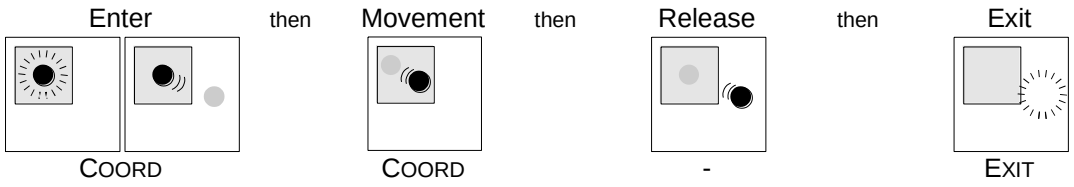


2.7.2.2.6 AOF_PRESSLOCAL

The AOF_PRESSLOCAL Modifier (Area Operating Flag: PRESSURE LOCAL, **0x20**) determines the source of the pressure sensitivity of the respective Area. If this Modifier is declared during the Area definition, the local pressure sensitivity is used. In this case the sensitivity is set during the Area definition and can be changed by means of the commands SETAREASTATE (0xA6) or SETAREAPRESSURE (0xA9) at a later point of time.
In the case of the AOF_PRESSLOCAL Modifier missing, the global pressure sensitivity is used for the respective Area. The global pressure sensitivity corresponds to the pressure sensitivity of AREA0 on the respective Area Page. When the global pressure sensitivity is used, by changing the pressure sensitivity of AREA0 the pressure sensitivity of several Areas on one Area Page can be changed simultaneously.

2.7.2.2.7 AOF_EXTENDED

If the AOF_EXTENDED Modifier (Area Operating Flag: EXTENDED, **0x40**) was declared during the Area definition, the respective Area is only considered not interrupted when the entire Touch Zone is not interrupted anymore by the corresponding touch spot. If an Area is interrupted by the AOF_EXTENDED Modifier and the Touch Zone is then interrupted by a second touch at another position, the second interruption is only reported after the first touch has completely released the Touch Zone.
The following illustration shows a typical sequence of events for an Area with the Operating Mode AOM_TRACK and the Modifiers AOF_ADDEXIT and AOF_EXTENDED:



2.7.2.2.8 AOF_ACTIVE

The AOF_ACTIVE Modifier (Area Operating Flag: PRESSURE LOCAL, **0x80**) is required for the respective Area to be taken into consideration regarding the creation of touch messages. By means of this Modifier single Areas on an Area Page can either be enabled or disabled.

2.7.3 Area0

Among the Area Pages AREA0 holds a special status. After either a reset or the deletion of all Areas by means of the command CLEARAREA (0xA0), there is the AREA0 defined on each Page. It occupies the entire Touch Zone and is set to the Operating Mode AOM_ENTER with the Modifiers AOF_ACTIVE and AOF_ADDCOORD. The pressure sensitivity equals about 100 g.
The Area rectangle of AREA0 can neither be changed with regard to position nor size nor can it be deleted. Alone the Operating Mode, the Modifiers and the pressure sensitivity of AREA0 can be set for each individual Area Page by means of the commands SETAREASTATE (0xA6), SETAREAMODE (0xA7), SETAREAFLAGS (0xA8), or SETAREAPRESSURE (0xA9). By not declaring the AOF_ACTIVE Modifier it is possible to disable AREA0.

2.8 Macros

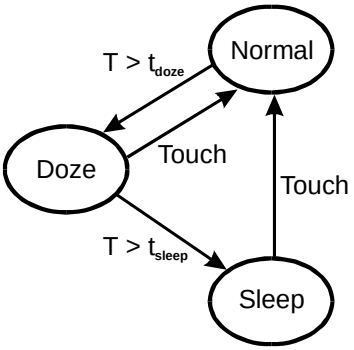
3rd Generation IRT does not support recording and executing of command sequences, so-called macros. Nevertheless the macro commands STARTMACRORECORD (0xE5), ENDMACRORECORD (0xE1), EXECMACRO (0xE2), FREEMACROSPACE (0x63) and GETFREEMACROSPACE (0xE3) are handled without error message.

2.9 Peripherals

Besides the beams and pressure sensors there are additional functional components for the touch detection on the IRT. These components are user-programmable and are referred to as "peripherals". What kind of peripherals is actually available can be read out from the report HARDWARE (0x31) by means of the command GETHARDWARE (0xB1)

2.9.1 Power Saving Modes

The IRT implements two power saving modes:Doze-Mode and Sleep-Mode. The Doze-Mode substantially decreases the power consumtion of the IRT without noticable reducing the response time. The Sleep-Mode decreases the power consumption even further, but the response time increases, too. The transistions between the power saving modes are shown in the following diagram:



2.9.1.1 Doze-Mode

If for a certain span of time the Touch Zone is not interrupted, the so-called Doze-Mode is automatically activated. The activated Doze-Mode slightly decreases the scan rate of the beams. This way the power consumption of the IRT is reduced. As soon as an interruption of the Touch Zone is detected, the Doze-Mode is deactivated and the Touch Zone will again be scanned with the maximum speed.
The behaviour of the Doze-Mode can be set by means of the command SETDOZEMODE (0xF9). The current Doze-Mode settings can be read out from the report DOZEMODESTATE (0x78) by means of the command GETDOZEMODE (0xF8).
The following operations can be released either by activating or deactivating the Doze-Mode:

Event	Output	Operation
Doze-Mode is activated:	Report to host	becomes active
Doze-Mode is deactivated:	Report to host	becomes inactive

2.9.1.2 Sleep-Mode

If the IRT is in Doze-Mode and Touch Zone is not interrupted for another certain span of time, the so-called Sleep-Mode is activated. The Sleep-Mode decreases the scan rate of the beams even further than the Doze-Mode does. This way the life expectancy of the beams is prolonged and the power consumption of the IRT is reduced. As soon as an interruption of the Touch Zone is detected, the Sleep-Mode is deactivated and the Touch Zone will again be scanned with the maximum speed.
With the Sleep-Mode activated, depending on the set scan rate the IRT's response time can be considerably longer as in normal operation. If, for example, a scan rate of 500 ms / scan is set, it may last up to a half of a second until the IRT detects the interruption and deactivates the Sleep-Mode.
The behaviour of the Sleep-Mode can be set by means of the command SETSLEEPMODE (0xF7). The current Sleep-Mode settings can be read out from the report SLEEPMODESTATE (0x73) by means of the command GETSLEEPMODE (0xF3).

The following operations can be released either by activating or deactivating the Sleep-Mode:

Event	Output	Operation
Sleep-Mode is activated:	OC_PWM Report to host	"sleep" is set
Sleep-Mode is deactivated:	OC_PWM Report to host	"active" is set

The Sleep-Mode can only be activated if the scanning was set by means of the command SETSCANNING (0xCE) beforehand. The duty-cycle of the OC_PWM output is set separately for active and inactive Sleep-Mode by means of the command SETPWM (0xF5). This way, for instance, the backlight of a TFT display can be dimmed for the case that there was no interruption of the Touch Zone for a longer period of time.

2.9.2 PWM output

The OC_PWM connection of the IRT provides a PWM (pulse-width modulation) signal with a frequency of about 10 kHz. The duty cycle is adjustable in 256 steps by means of the command SETPWM (0xF5). At each step a setting for active and inactive TouchSaver can be determined. This signal can then, for example, be used to control the dimming level of the backlight inverter of a TFT display.

2.9.3 Input Port

All IRTs has a GP_IN (General Purpose INput) input port.
The current status of the input port can be read out from the report PORT (0x70) by means of the command GETPORT (0xF0).
Additionally the port can be used for touch-point validation. To do this set the Area pressure level to maximum (255) and use one of the AOF_ADDPRESS, AOF_PRESSENTER, or AOF_PRESSALWAYS flags. Now the port is used instead of the z-sensors to validate this area.

2.9.4 Sound source

The IRT can be equipped with an optional audio amplifier (LM4861) with 0.5W output power. The outputs of this amplifier are available at the extension connector.
The frequency and the duration of the emitted beep are set by means of the command SETSOUND (0xF6). The actually emitted frequency and the remaining time until the beep stops can be read out from the report SOUND (0x72) by means of the command GETSOUND (0xF2).

3 Reference of the CTS protocol

The following chapter lists both all commands that are accepted by the IRT at the activation of the CTS protocol as well as all messages and reports that are transmitted by the IRT.

3.1 Chart of commands

The following chart lists all commands that are accepted by the IRT arranged in functional groups.

3.1.1 System

Command	Identification	Description	Page
SoftReset	0x80	Initiates a warm start	38
Reset_CTS	0x81	Initializes the IRT in the CTS protocol, at the same time warm start for an already linked IRT	31
SaveSetup	0x83	Saves current settings to the FLASH EPROM	31
DestroySetup	0x84	Destroys a saved Setup	27
BREAK	100 ms TTL-Low applied to RxD	Resets the IRT. A possibly saved Setup will be disregarded.	25

3.1.2 Coordinates

Commands of this category determine the parameters for detecting and reporting coordinates.

Command	Identification	Description	Page
GetBeamMinMax	0xC0	Requests the minimum and maximum number of interrupted beams for a valid interruption	28
GetBeamTimeout	0xC1	Requests the blank-out time for defective beams	28
GetContTime	0xC2	Requests the time interval between two Continuous messages	28
GetDualTouching	0xC3	Requests the behaviour in case of dual touching	28
GetOrigin	0xC4	Requests the coordinates origin	29
GetResolution	0xC5	Requests the coordinates resolution	30
GetScanning	0xC6	Requests whether or not the beams are to be scanned	30
GetTransmission	0xC7	Requests the behaviour set by means of SetTransmission	31
SetBeamMinMax	0xC8	Sets the minimum and maximum number of interrupted beams for a valid interruption.	34
SetBeamTimeout	0xC9	Sets the blank-out time for defective beams	34
SetContTime	0xCA	Sets the time interval between two continuous messages	34
SetDualTouching	0xCB	Sets the behaviour in case of dual touching	35
SetOrigin	0xCC	Sets the coordinates origin	35
SetResolution	0xCD	Sets the coordinates resolution	36
SetScanning	0xCE	Activates or deactivates the scanning of the beams	36
SetTransmission	0xCF	Activates or deactivates the spontaneous transmission of messages	38
GetTouchTime	0xD0	Requests the minimum duration for a valid interruption	31
SetTouchTime	0xD1	Sets the minimum duration for a valid interruption	37

3.1.3 Report requests

Commands of this category request reports about the status and the version of the IRT.

Command	Identification	Description	Page
GetErrors	0xB0	Requests an error report	29
GetHardware	0xB1	Requests a report about the IRT hardware	29
GetRevisions	0xB2	Requests a version report	30

Command	Identification	Description	Page
GetSetup	0xB3	Requests information about saved Setup parameter sets	30
GetSingleMessage	0xB4	Requests a single message	30
GetSingleScan	0xB5	Initiates a scan operation and provides the result	30
GetOemString	0xB8	Requests the OEM string with the serial number	29
GetHWVersions	0xB9	Request the hard coded versions	29

3.1.4 Areas

Commands of this category determine the behaviour of the Touch Areas.

Command	Identification	Description	Page
ClearArea	0xA0	Clears either one or more Area definitions	25
DefineArea	0xA1	Defines a rectangular Area	26
GetAreaDef	0xA2	Requests the definition of an Area	28
GetAreaPage	0xA3	Requests the currently set Area Page	28
GetFreeAreaSpace	0xA4	Requests the available number of Areas	29
SelectAreaPage	0xA5	Selects an Area Page	31
SetAreaState	0xA6	Changes several operating parameters of an Area	32
SetAreaMode	0xA7	Changes the Operating Mode of an already defined Area	33
SetAreaFlags	0xA8	Changes the Operating Mode Flags of an already defined Area	33
SetAreaPressure	0xA9	Changes the pressure sensitivity of an already defined Area	34
DefPolyArea	0xAA	Defines a polygonal Area	27

3.1.5 Macros

Commands of this category either record macros or execute recorded macros.

Command	Identification	Description	Page
ClearMacro	0xE0	Clears either one or more macro definitions	25
EndMacroRecord	0xE1	Ends macro recording	27
ExecMacro	0xE2	Executes a macro	28
GetFreeMacroSpace	0xE3	Requests the available memory for macros	29
StartMacroRecord	0xE5	Starts macro recording	38

3.1.6 Peripherals

Commands of this category control the peripherals of the IRT. The peripherals do not contain the beams and the pressure sensors.

Command	Identification	Description	Page
GetPort	0xF0	Requests the current status of the input port	30
GetPWM	0xF1	Requests the current PWM settings	30
GetSound	0xF2	Requests the current status of the sound source	31
GetSleepMode	0xF3	Requests the current Sleep-Mode status	31
SetPort	0xF4	Sets an output port	36
SetPWM	0xF5	Sets the PWM output	36
SetSound	0xF6	Emits a beep	37
SetSleepMode	0xF7	Sets the Sleep-Mode parameters	37
GetDozeMode	0xF8	Requests the current Doze-Mode status	28
SetDozeMode	0xF9	Sets the Doze-Mode parameters	35

3.2 Charts of reports and messages

The following charts list all messages of the IRT arranged in functional groups.

3.2.1 Messages

Messages are only created if the scanning is activated. The IRT transmits messages without further requests only if the data transmission is enabled.

Message	Identification	Description	Page
DualTouchError	0x18	Dual touch error	41
Coord	0x19	Position of the interruption	40
Exit	0x1A	Exit position	43
Pressure	0x1B	Pressure either exceeded or fell below the limit value	46
SleepMode	0x1C	Sleep-Mode activation or deactivation	48
DozeMode	0x1D	Doze-Mode activation or deactivation	41

3.2.2 Reports

Reports are only transmitted upon request, regardless whether or not the data transmission is enabled.

Report	Identification	Description	Page
AreaDefinition	0x22	Area definition	39
AreaPage	0x23	Currently set Area Page	40
FreeAreaSpace	0x24	Available memory for Area definitions	43
PolyAreaDef	0x2A	Polygonal Area definition	45
ErrorReport	0x30	Error report	42
Hardware	0x31	IRT hardware description	44
Revision	0x32	Versions	47
Setup	0x33	Information about saved Setup parameter sets	48
Idle	0x34	No message available	45
OemString	0x38	OEM string with serial number	45
HWVersion	0x39	Hard coded versions	44
BeamMinMax	0x40	The minimum and maximum number of interrupted beams for a valid interruption	40
BeamTimeout	0x41	Blank-out time for continuously interrupted beams	40
ContTime	0x42	Time interval between two messages in the Continuous Mode	40
DualTouching	0x43	Behaviour in case of dual touching	41
Origin	0x44	Information about the coordinates origin	45
Resolution	0x45	Coordinates resolution	47
Scanning	0x46	Scanning of beams on or off	48
Transmission	0x47	Spontaneous data transmission on/off and flow control	49
TouchTime	0x50	Minimum duration of a valid interruption	49
FreeMacroSpace	0x63	Available memory for macro definitions	43
Port	0x70	Status of the input and output ports	46
PWM	0x71	Settings of PWM Unit	46
Sound	0x72	Current status of sound source	48
SleepModeState	0x73	Settings of Sleep-Mode	48
DozeModeState	0x78	Settings of Doze-Mode	41

3.3 Command reference

The following represents a detailed description of all commands interpretable by the IRT. The commands are listed in alphabetical order. Although essential for the protocol, in the descriptions of the command formats the DC2, DC4 and SYN data are disregarded. Therefore, in order to obtain a command valid to the IRT, first a DC2 must precede each command identification, then - if required - the data must be encoded with SYN and eventually a DC4 must follow the command. Each parameter provided with a name holds a length of 1 byte (8 bit).

BREAK

100 ms TTL-Low
applied to RxD

If the IRT recognizes a Break a hardware reset is carried out. During the following initialization possibly saved Setup parameters will be disregarded. This way an IRT with either unknown or faulty Setup can be initialized anew.

ClearArea

0xA0 *Mode [NumberLo [NumberHi [Page]]]*

Clears either one or more Area definitions.

Parameters:

Mode [0x00..0x02]

Determines what is to be cleared:

0x00	Clears all Area definitions
0x01	Clears a certain Page
0x02	Clears a certain Area

NumberLo [0x00..0xFF]
NumberHi [0x00..0xFF]

Determines either the Area or Page to be cleared.
In case *Mode*=**0x00** *NumberLo* and *NumberHi* are disregarded and do not have to be transmitted.
In case *Mode*=**0x01** *NumberLo* holds the number of the Page to be cleared. *NumberHi* is disregarded and does not have to be transmitted.
In case *Mode*=**0x02** *NumberLo* + 256 × *NumberHi* equals the number of the Area to be cleared.

Page [0x00..0xFF]

Determines the Page on which the Area to be cleared is defined on. If this parameter is not declared, the Area of the currently active Page is cleared. If *Mode* does not equal **0x02** this parameter is disregarded.

ClearMacro

0xE0 *[NumberLo NumberHi]*

Clears either one or all macros.

Parameters:

NumberLo [0x00..0xFF]
NumberHi [0x00..0xFF]

Determines which macro is to be cleared.
Continued for compatibility reasons with previous CTS protocols. Functionless stub.

DefineArea

0xA1 *NumberLo NumberHi [Page [Xlo Xhi [Ylo Yhi [Wlo Whi [Hlo Hhi [Mode [Flags [Press]]]]]]]]*

Defines a rectangular Area.

Parameters:

NumberLo [0x00..0xFF]
NumberHi [0x00..0xFF]

NumberLo + 256 × *NumberHi* determines the number of the Area to be defined.

Page [0x00..0xFF]

Number of Page the Area is to be defined on. If this parameter is not declared, the page of the recently defined Area is used. If there was no Area previously defined, Page 0 is assumed.

Xlo [0x00..0xFF]
Xhi [0x00..0xFF]
Ylo [0x00..0xFF]
Yhi [0x00..0xFF]

Top left corner of the Area rectangle. If these parameters are omitted, the top left corner of the recently defined Area is used. If there was no Area previously defined, (0,0) is assumed.

Wlo [0x00..0xFF]
Whi [0x00..0xFF]
Hlo [0x00..0xFF]
Hhi [0x00..0xFF]

Width and height of the Area rectangle. If these parameters are omitted, width and height of the recently defined Area are used. If there was no Area previously defined, the maximum width and height are assumed.

Mode [0x00..0xFF]

Operating Mode of the Area. By means of one of the parameters listed below the behaviour of each single Area can be set separately. If this parameter is not declared, the Operating Mode of the recently defined Area is used. If there was no Area previously defined, AOM_ENTER is assumed.

0x00 AOM_OFF: For this Area no coordinate messages are created. However, in case the AddExit Flag is set, a message is issued when exiting the Area.
0x01 AOM_ENTER: Reports only the interruption of the Area.
0x02 AOM_TRACK: Reports an interruption and each further change of the coordinates within the Area.
0x03 AOM_CONT: Reports the first interruption of the Area and afterwards the current position in regular time intervals (refer to SETCONTTIME, 0xCA).

Flags [0x00..0xFF]

The Flags listed below can modify the Operating Mode of the Area. If this parameter is not declared, the Flags of the recently defined Area are used. If there was no Area previously defined, 0x82 (standard setting AREA0) is assumed. Several Flags can be Ored together.

0x01 AOF_ADDEXIT: Reports the exiting of the Area.
0x02 AOF_ADDCOORD: Adds the coordinate of the touch spot to the message.
0x04 AOF_ADDPRESS: Issues a message if the pressure exerted onto the front screen either exceeds or falls below the limit value that was predetermined for this Area.
0x08 AOF_PRESSALWAYS: For all Area messages to be received, a certain pressure has to be exceeded.
0x10 AOF_PRESSENTER: For the very first Area message a certain pressure has to be exceeded. For any additional messages a regular interruption of the Touch Zone is sufficient.
0x20 AOF_PRESSLOCAL: There is a local pressure value used for this Area. If this Flag is not explicitly declared, the pressure value of AREA0 is used.
0x40 AOF_EXTENDED: Enables the "N-Key-Rollover" emulation for this Area. In this case messages of other Areas will not be issued until the interrupted Touch Zone of this Area is not completely released yet.
0x80 AOF_ACTIVE: The Area is activated. Only active Areas create messages.

Press [0x00..0xFF]

Pressure sensitivity of this Area. If this parameter is not declared, the pressure sensitivity of the recently defined Area is used. If there was no Area previously defined, a pressure sensitivity of 0 (no pressure) is assumed. A pressure sensitivity of 0xFF uses an external switch at the GP_IN input to provide the pressure information.

Default:
AREA0: AOM_ENTER, AOF_ADDCOORD, AOF_ACTIVE, No additional Areas defined

DefPolyArea

0xAA *NumberLo NumberHi CountLo CountHi X_0Lo X_0Hi Y_0Lo Y_0Hi X_1Lo X_1Hi Y_1Lo Y_1Hi X_2Lo X_2Hi Y_2Lo Y_2Hi [... X_nLo X_nHi Y_nLo Y_nHi] [Page [Mode [Flags [Press]]]]*
Defines a polygonal Area.
Parameters:
NumberLo [0x00..0xFF]
NumberHi [0x00..0xFF]
NumberLo + 256 × *NumberHi* determines the number of the Area to be defined.
CountLo [0x00..0xFF]
CountHi [0x00..0xFF]
CountLo + 256 × *CountHi* determines the number of polygone edges that will follow. At least 3 edges have to be defined for a valid polygonal Area. The polygone is automatically closed by connecting the last edge to the first one.
X_n*Lo* [0x00..0xFF]
X_n*Hi* [0x00..0xFF]
X_n*Lo* + 256 × X_n*Hi* defines the X-coordinate of a polygone edge.
Y_n*Lo* [0x00..0xFF]
Y_n*Hi* [0x00..0xFF]
Y_n*Lo* + 256 × Y_n*Hi* defines the Y-coordinate of a polygone edge.
Page [0x00..0xFF]
See description of the DEFINEAREA command.
Mode [0x00..0xFF]
See description of the DEFINEAREA command.
Flags [0x00..0xFF]
See description of the DEFINEAREA command.
Press [0x00..0xFF]
See description of the DEFINEAREA command.
Default:
AOM_ENTER, AOF_ADDCOORD, AOF_ACTIVE, No polygonal Areas defined

DestroySetup

0x84 *Selection*
Destroys one or more previously saved Setup parameters.
Parameters:
Selection
Determines the Setup parameter set to be destroyed. If there are several parameter sets to be destroyed at the same time, the respective constants need to be ORed.

0x01	SERIALSETUP: Baud rate, parity, emulation mode
0x04	AREADEFINITIONS: Currently defined Areas and the active Area Page
0x08	PERIPHERALSETTINGS: PWM, /OUT0, TouchSaver
0x10	COORDINATESETTINGS: Minimum and maximum width of the interruption, blank-out time for defective beams, Continuous Time, behaviour in case of multiple touching, coordinates origin, coordinates resolution, number of scans per coordinate, data transmission.

EndMacroRecord

0xE1 *NumberLo [NumberHi]*
Ends the macro recording.
Parameters:
NumberLo [0x00..0xFF]
NumberHi [0x00..0xFF]
Continued for compatibility reasons with previous CTS protocols. Functionless stub.

ExecMacro

0xE2 *NumberLo [NumberHi]*

Executes a macro.

Parameters:

NumberLo [0x00..0xFF]

NumberHi [0x00..0xFF]

Continued for compatibility reasons with previous CTS protocols. Functionless stub.

GetAreaDef

0xA2 *NumberLo [NumberHi [Page]]*

Requests a AREADEFINITION report (0x22) or an POLYAREADEF report (0x2A) about an Area definition.

Parameters:

NumberLo [0x00..0xFF]

NumberHi [0x00..0xFF]

Number of Area whose definition is to be provided. If *NumberHi* is not declared, it is assumed to be 0.

Page [0x00..0xFF]

Page on which the Area is situated. If this parameter is not declared, the currently active Area Page is used.

GetAreaPage

0xA3

Requests the AREAPAGE report (0x23) about the currently active Area Page.

Parameters:

None

GetBeamMinMax

0xC0

Requests the BEAMMINMAX report (0x40) about the minimum and maximum permitted width of an interruption.

GetBeamTimeout

0xC1

Requests the BEAMTIMEOUT report (0x41) about the blank-out time for continuously interrupted beams.

GetContTime

0xC2

Requests the CONTTIME report (0x42) about the time interval between Continuous messages.

GetDozeMode

0xF8

Requests the DOZEMODESTATE report (0x78) about the current Doze-Mode status.

GetDualTouching

0xC3

Requests the DUALTOUCHING report (0x43) about the behaviour at dual touching.

GetErrors

0xB0 *Selection*

Requests an ERRORREPORT report (0x30) about errors that may have occurred.

Parameters:

Selection

By means of this parameter the desired report is selected. If more than one error report is to be transmitted, you can characterize a combination of classifications by ORing these constants.

0x01	INITIALERRORS: All errors detected during a system initialization after a reset.
0x02	DEFECTBEAMS: Numbers of the blanked-out beams
0x04	COMMUNICATIONS: Transmission error (overrun, protocol violation, ...).
0x08	COMMAND STATUS: Result of the recently transmitted command (except for this GetErrors command).
0x80	CLEARERRORS: The errors are cleared after being reported.

GetFreeAreaSpace

0xA4

Requests a FREEAREASPACE report (0x24) about the number of Areas that are still available to be defined.

GetFreeMacroSpace

0xE3

Continued for compatibility reasons with previous CTS protocols. Functionless stub.

GetHardware

0xB1 *Selection*

Requests a HARDWARE report (0x31).

Parameters:

Selection [0x00..0x07]

This parameter selects the desired report. If more than one hardware report is to be transmitted, you can characterize a combination of classifications by ORing these constants

0x01	BEAMCOUNT: Number of X/Y-beams.
0x02	PRESSURESENSORS: Number of pressure sensors.
0x04	PERIPHERALS: Available peripherals.

GetHWVersions

0xB9 *Selection*

Requests a HWVERSION report (0x39).

Parameters:

Selection [0x00..0x07]

This parameter selects the desired report. If more than one hardware report is to be transmitted, you can characterize a combination of classifications by ORing these constants

0x01	SILICONSERIALNUMBER: The unique silicon serial number assembled to the IRT.
0x02	HARDWARECODE: A code for automatic assembly identification.
0x04	FPGAREVISION: The revision of the FPGA-configuration data stream.

GetOemString

0xB8

Requests a OEMSTRING report (0x38) about the serial number of the IRT.

GetOrigin

0xC4

Requests an ORIGIN report (0x44) about the position of the coordinates origin.

GetPort

0xF0 *Selection*

Requests the peripheral ports of the IRT and provides a PORT report (0x70).

Parameters:

Selection

Selects the peripheral ports to be requested. Possible peripheral ports:

0x08	GP_IN

GetPWM

0xF1

Requests a PWM report (0x71) about the current PWM settings.

GetResolution

0xC5

Requests a RESOLUTION report (0x45) about the current coordinates resolution.

GetRevisions

0xB2 *Selection*

Requests a REVISION report (0x32) about the version of the IRT soft- and hardware.

Parameters:

Selection

This parameter selects the desired revision. If more than one revision number is to be provided, the respective constants can be ORed.

0x01	Revision number of the system manager module
0x02	Revision number of the hardware driver module
0x04	Revision number of the process module
0x08	Revision number of the Protocol module
0x10	Revision number of the hardware parameters module
0x20	Designation of the IRT hardware
0x40	Revision number of the Burn-In module
0x80	Revision number of the FPGA module

GetScanning

0xC6

Requests a SCANNING report (0x46). This report indicates whether or not the beams are scanned.

GetSetup

0xB3

Requests a SETUP report (0x33) about the saved Setup parameter sets.

GetSingleMessage

0xB4

Requests a single message. In case the data transmission is disabled, the IRT saves the recently occurred message. This message can then explicitly be requested by means of GETSINGLEMESSAGE.

GetSingleScan

0xB5

The IRT scans the beams once and as a result reports the positions of the interrupted beams.

GetSleepMode

0xF3

Requests a SLEEPMODESTATE report (0x73) about the current Sleep-Mode status.

GetSound

0xF2

Requests a SOUND report (0x72) about the current status of the sound source.

GetTouchTime

0xD0

Requests a TOUCHTIME report (0x50) about the minimum duration of a valid interruption.

GetTransmission

0xC7

Requests a TRANSMISSION report (0x47) about the settings of the serial data transmission.

Reset_CTS

0x81

Initializes the CTS instruction set.
If the command RESET_CTS is transmitted again at a later point of time, and then with the protocol taken into consideration, all settings are reset to the values they have had after the initialization.

SaveSetup

0x83 *Selection*

Saves one or more Setup parameter sets. After either a cold start or a soft reset of the IRT, these parameters are used to initialize the IRT again.

ATTENTION:	ALL AREA DEFINITIONS ARE DELETED BY THIS COMMAND!

Parameters:

Selection

Determines the Setup parameter set to be saved. If several parameter sets are to be saved simultaneously the respective constants can be bitwise ORed.

0x01	SERIALSETUP: Baud rate, parity, emulation mode
0x04	AREADEFINITIONS: Currently defined Areas and the active Area Page
0x08	PERIPHERALSETTINGS: PWM, TouchSaver
0x10	COORDINATESETTINGS: Minimum and maximum width of the interruption, blank-out time for defective beams, Continuous Time, behaviour in case of multiple touching, coordinates origin, coordinates resolution, number of scans per coordinate, data transmission.

SelectAreaPage

0xA5 *Page*

Selects a Page with Area definitions.

Parameters:

Page [0x00..0xFF]

Determines the number of Page to be activated.

Default:

Page 0

SetAreaState

0xA6 *NumberLo NumberHi Flags [Press]*

Either enables or disables an Area and changes the pressure sensitivity of this Area. The Area that is changed is always situated on the currently active Page.

Parameters:

NumberLo [0x00..0xFF]

NumberHi [0x00..0xFF]

Number of Area.

Flags [0x00..0xFF]

The new Operating Mode Modifier Flags for the declared Area:

0x01	AOF_ADDEXIT: Reports the exiting of an Area.
0x02	AOF_ADDCOORD: Reports the coordinates of the touch spot.
0x04	AOF_ADDPRESS: Issues a message if the pressure exerted onto the front screen either exceeds or falls below the limit value that was predetermined for this Area.
0x08	AOF_PRESSALWAYS: For all Area messages to be received, a certain pressure has to be exceeded.
0x10	AOF_PRESSENTER: For the very first Area message a certain pressure has to be exceeded. For any additional messages a regular interruption of the Touch Zone is sufficient.
0x20	AOF_PRESSLOCAL: There is a local pressure value used for this Area. If this Flag is not explicitly declared, the pressure value of Area-0 is used.
0x40	AOF_EXTENDED: Enables the "N-Key-Rollover" emulation for this Area. In this case messages of other Areas will not be issued until the interrupted Touch Zone of this Area is not completely released yet.
0x80	AOF_ACTIVE: The Area is activated. Only active Areas create messages.

Press [0x00..0xFF]

Pressure sensitivity of this Area. If this parameter is not declared, the pressure sensitivity of the Area remains unchanged.

Default:

AREA0: AOM_ENTER, AOM_ADDCOORD, AOM_ACTIVE

SetAreaFlags

0xA8 *Flags [Page [NumberLo [NumberHi]]]*

Changes the Operating Mode Modifier Flags of an already defined Area.

Parameters:

Flags [0x00..0xFF]

The new Operating Mode Modifier Flags for the declared Area:

- **0x01** AOF_ADDEXIT: Reports the exiting of an Area.
- **0x02** AOF_ADDCOORD: Reports the coordinates of the touch spot.
- **0x04** AOF_ADDPRESS: Issues a message if the pressure exerted onto the front screen either exceeds or falls below the limit value that was predetermined for this Area.
- **0x08** AOF_PRESSALWAYS: For all Area messages to be received, a certain pressure has to be exceeded.
- **0x10** AOF_PRESSENTER: For the very first Area message a certain pressure has to be exceeded. For any additional messages a regular interruption of the Touch Zone is sufficient.
- **0x20** AOF_PRESSLOCAL: There is a local pressure value used for this Area. If this Flag is not explicitly declared, the pressure value of Area-0 is used.
- **0x40** AOF_EXTENDED: Enables the "N-Key-Rollover" emulation for this Area. In this case messages of other Areas will not be issued until the interrupted Touch Zone of this Area is not completely released yet.
- **0x80** AOF_ACTIVE: The Area is activated. Only active Areas create messages.

Page [0x00..0xFF]

Page on which the Area to be changed is defined. If this parameter is omitted, an Area situated on the currently active Page is changed.

NumberLo [0x00..0xFF]

NumberHi [0x00..0xFF]

Number of Area. If *NumberHi* is omitted, it is assumed to be 0. If both *NumberHi* and *NumberLo* are omitted, AREA0 is changed.

SetAreaMode

0xA7 *Mode [Page [NumberLo [NumberHi]]]*

Changes the Operating Mode of an already defined Area.

Parameters:

Mode

Operating Mode of the Area. By means of one of the parameters listed below the behaviour of each single Area can be set separately.

- **0x00** AOM_OFF: For this Area there are no coordinate messages created. However, in case the AddExit Flag is set, a message is issued when exiting the Area.
- **0x01** AOM_ENTER: Reports the interruption of the Area only.
- **0x02** AOM_TRACK: Reports an interruption and each further change of the coordinates within the Area.
- **0x03** AOM_CONT: Reports the first interruption of the Area and afterwards the current position in regular time intervals (refer to SETCONTTIME, 0xCA).

Page [0x00..0xFF]

Page on which the Area to be changed is defined. If this parameter is omitted, an Area situated on the currently active Page is changed.

NumberLo [0x00..0xFF]

NumberHi [0x00..0xFF]

Number of Area. If *NumberHi* is omitted, it is assumed to be 0. If both *NumberHi* and *NumberLo* are omitted, AREA0 is changed.

SetAreaPressure

0xA9 *Pressure [Page [NumberLo [NumberHi]]]*

Changes the pressure sensitivity of an already defined Area.

Parameters:

Pressure [0x00..0xFF]

The new pressure sensitivity of the Area.

Page [0x00..0xFF]

Page on which the Area to be changed is defined. If this parameter is omitted, an Area situated on the currently active Page is changed.

NumberLo [0x00..0xFF]

NumberHi [0x00..0xFF]

Number of Area. If *NumberHi* is omitted, it is assumed to be 0. If both *NumberHi* and *NumberLo* are omitted, AREA0 is changed.

SetBeamMinMax

0xC8 *MinX [MaxX [MinY [MaxY]]]*

Determines the minimum and maximum number of beams that are permitted to be interrupted next to each other in order to receive a valid coordinates message.

Parameters:

MinX [0x00..0xFF]

MinY [0x00..0xFF]

Minimum number of interrupted beams in X- and Y-direction.

MaxX [0x00..0xFF]

MaxY [0x00..0xFF]

Maximum number of interrupted beams in X- and Y-direction.

If one of the parameters is omitted, a default value according to the following table is used:

Input parameters	Used parameters for MaxX	MinY	MaxY
MinX	*MinX*	*MinX*	*MinX*
MinX, MaxX	-	*MinX*	*MaxX*
MinX, MaxX, MinY	-	-	*MaxX*
MinX, MaxX, MinY, MaxY	-	-	-

Default:

MinX = 1, *MinY* = 1, *MaxX* = 5, *MaxY* = 5

SetBeamTimeout

0xC9 *TimeLo [TimeHi]*

Determines the time span that has to elapse before a beam is considered defective, blanked-out and excluded from the coordinates evaluation.

Parameters:

TimeLo [0x00..0xFF]

TimeHi [0x00..0xFF]

Time value in seconds (0x00 = never blank-out). If *TimeHi* is not explicitly declared, it is assumed to be 0x00.

Default: **10 seconds**

SetContTime

0xCA *TimeLo [TimeHi]*

Determines the time interval between two COORD messages (0x19) in the Continuous Mode.

Parameters:

TimeLo [0x00..0xFF]

TimeHi [0x00..0xFF]

Time value in milliseconds. If *TimeHi* is not explicitly declared, it is assumed to be 0x00.

Default: **55 milliseconds**

SetDozeMode

0xF9 *Mode TimeLo [TimeHi [ScanLo [ScanHi]]]*

Sets the functions of the Doze-Mode.

Parameters:

Mode [0x00..0x03]

Determines the behaviour of the Doze-Mode.

0x00	No message at either activation or deactivation
0x01	Message at activation
0x02	Message at deactivation
0x03	Message at activation and deactivation

TimeLo [0x00..0xFF]

TimeHi [0x00..0xFF]

Activation time in seconds (0x0000 = immediately activated, 0xFFFF = always deactivated). If *TimeHi* is omitted, the value 0 is assumed for *TimeHi*.

ScanLo [0x00..0xFF]

ScanHi [0x00..0xFF]

Time interval between two scan operations while in Doze-Mode. The time interval is set in steps of milliseconds. If *ScanHi* is omitted, the value 0 is assumed for *ScanHi*. If both *ScanLo* and *ScanHi* are omitted, the recently set value is used.

Default:

Doze-Mode after 10 seconds, *Scan* (=*ScanLo*+*256*×*ScanHi*) = 25 ms.

SetDualTouching

0xCB *Mode*

Determines the behaviour of the IRT in case of dual touching.

Parameters:

Mode [0x00..0x02]

The IRT recognizes the following constants. The constants can be bitwise ORed.

0x00	Dual touching is disregarded.
0x01	Inadmissible dual touching results in an DualTouchError message (0x18).
0x02	The coordinates of the second touch spot are reported.

Default:

Dual touching is disregarded.

SetOrigin

0xCC *Selection*

Sets the coordinates origin to one of the four corners of the screen.

Parameters:

Selection [0x00..0x03]

The following values are accepted by the IRT:

0x00	TOPLEFT: Origin set to the left-hand side top corner.
0x01	TOPRIGHT: Origin set to the right-hand side top corner.
0x02	BOTTOMRIGHT: Origin set to the right-hand side bottom corner.
0x03	BOTTOMLEFT: Origin set to the left-hand side bottom corner.

Default:

TOPLEFT

SetPort

0xF4 *Port Level*

This command is still supported by 3RD generation IRT's, but has no effect any more since there is no /OC_OUT0 output available for these touches.

SetPWM

0xF5 *Active [Sleep]*

Determines the mark-to-space ratio of the PWM output. Higher values result in longer pulse widths.

Parameters:

Active [0x00..0xFF]

Duty cycle in normal operation (Sleep-Mode inactive).

Sleep [0x00..0xFF]

Duty cycle while in Sleep-Mode. If *Sleep* is omitted, the value determined for *Active* is also used for the *Sleep* parameter.

Default:

Active = 0xFF

Sleep = 0xFF

SetResolution

0xCD *XLo [XHi [YLo [YHi]]]*

Determines the value range for coordinates messages.

Parameters:

XLo [0x00..0xFF]

XHi [0x00..0xFF]

YLo [0x00..0xFF]

YHi [0x00..0xFF]

Maximum coordinates in X- or Y-direction, respectively. The minimum coordinate always equals 0. In order to utilize the preset value range to the best possible extent, an interpolation of the signal values is carried out.

In case some of the optional parameters are omitted, the values for X and Y are determined according to the following table:

Input parameters	used parameters for X	Y
Xlo	XLo	XLo
XLo, Xhi	XLo+256×Xhi	XLo+256×Xhi
XLo, XHi, Ylo	XLo+256×Xhi	YLo
XLo, XHi, YLo, YHi	XLo+256×Xhi	YLo+256×Yhi

Default:

640 × 480

SetScanning

0xCE *Mode*

Either enables or disables the scanning of the beams.

Parameters:

Mode [0x00..0x01]

0x00 scanning disabled

0x01 scanning enabled

Default:

Scanning disabled

SetSleepMode

0xF7 *Mode TimeLo [TimeHi [ScanLo [ScanHi]]]*

Sets the functions of the Sleep-Mode.

Parameters:

Mode [0x00..0x03]

Determines the behaviour of the Sleep-Mode.

0x00	No message at either activation or deactivation
0x01	Message at activation
0x02	Message at deactivation
0x03	Message at activation and deactivation

TimeLo [0x00..0xFF]
TimeHi [0x00..0xFF]

Activation time in seconds (0x0000 = immediately activated, 0xFFFF = always deactivated). If *TimeHi* is omitted, the value 0 is assumed for *TimeHi*.

ScanLo [0x00..0xFF]
ScanHi [0x00..0xFF]

Time interval between two scan operations while in Sleep-Mode. The time interval is set in steps of milliseconds. If *ScanHi* is omitted, the value 0 is assumed for *ScanHi*. If both *ScanLo* and *ScanHi* are omitted, the recently set value is used.

Default:

Sleep-Mode disabled, *Scan* (=*ScanLo+256*×*ScanHi*) = 500 ms.

SetSound

0xF6 *[FreqLo [FreqHi [TimeLo [TimeHi]]]]*

Emits a beep with a certain frequency and a certain duration.

Parameters:

FreqLo [0x00..0xFF]
FreqHi [0x00..0xFF]

Frequency of the beep in Hertz. Frequencies of 20 Hz up to 3,5 kHz are permitted.

TimeLo [0x00..0xFF]
TimeHi [0x00..0xFF]

Duration of the beep in milliseconds (0x0000=immediately off, 0xFFFF = continuous beep). If *TimeHi* is not explicitly declared, the value 0 is assumed for *TimeHi*. If the command SETSOUND is used without any further parameters, a currently emitted beep is turned off.

Default:

Sound source disabled

SetTouchTime

0xD1 *Time*

Determines the minimum time span for a valid interruption. In order for an interruption to be reported to the host computer as valid, it needs to remain at the same spot for at least the time span declared here.

Parameters:

Time [0x00..0xFF]

Minimum duration of a valid interruption in steps of 10 ms.

Default:

Maximum touch speed (≈ 20 ms).

SetTransmission

0xCF *Mode [TimeLo [TimeHi]]*

Controls both the spontaneous data transmission and the flow control.
In case the data transmission is disabled, messages are only transmitted upon request by means of the command GETSINGLEMESSAGE (0xB4). The most recent message in stored in a buffer and can be requested at any later point of time. In case the data transmission is enabled, the messages are immediately transmitted. In the moment the data transmission becomes enabled, all messages collected so far are deleted.
The flow control can be set separately for reception and transmission. For the control XON/XOFF is used. If the IRT receives a XOFF (0x13), a timer is started. If the IRT does not receive a XON (0x11) within a predetermined period of time, it automatically changes into the XON Mode and sets a XOFF-Timeout Error Flag.

Parameters:

Mode [0x00..0xFF]

In order to set the desired behaviour, the values listed below must be ORed accordingly.

0x00	Data transmission disabled
0x01	Data transmission enabled
0x10	Flow control for transmission (**IRT** transmits XON/XOFF to the host)
0x20	Flow control for reception (Host transmits XON/XOFF to the **IRT**)

TimeLo [0x00..0xFF]
TimeHi [0x00..0xFF]

XOFF timeout in milliseconds. In case *TimeHi* is not explicitly declared, it is assumed to be 0. In case neither *TimeLo* nor *TimeHi* are declared, the XOFF timeout is disabled.

Default:

Data transmission disabled, no flow control.

SoftReset

0x80

Carries out a cold start of the IRT.
The IRT carries out a real hardware reset. Afterwards the process is just like after turning on the power supply. Therefore all saved Setup parameter sets are used.

StartMacroRecord

0xE5 *NumberLo [NumberHi]*

Starts a macro recording.
Succeeding to this command all following commands will not be executed by the IRT but stored in the macro memory. The COMMAND EndMacroRecord (0xE1) ends the macro recording mode.

Parameters:

NumberLo [0x00..0xFF]
NumberHi [0x00..0xFF]

Continued for compatibility reasons with previous CTS protocols. Functionless stub.

3.4 Message and report references

The following represents a detailed description of all reports and messages transmitted by the IRT. Although essential for the protocol, in the descriptions of the report formats the DC2, DC4 and SYN data are disregarded. The IRT first places a preceding DC2 to each report or message, then - if required - the data must be encoded with SYN and eventually a DC4 is transmitted. Each parameter provided with a name holds a length of 1 byte (8 bit).

AreaDefinition

0x22 *NumberLo NumberHi Page Xlo Xhi Ylo Yhi Wlo Whi Hlo Hhi Mode Flags Press*

Returns an Area definition.

Parameters:

NumberLo [0x00..0xFF]

NumberHi [0x00..0xFF]

NumberLo + 256 * *NumberHi* determines the number of the Area.

Page [0x00..0xFF]

Number of Page the Area is defined on.

Xlo [0x00..0xFF]

Xhi [0x00..0xFF]

Ylo [0x00..0xFF]

Yhi [0x00..0xFF]

Left-hand side top corner of Area rectangle.

Wlo [0x00..0xFF]

Whi [0x00..0xFF]

Hlo [0x00..0xFF]

Hhi [0x00..0xFF]

Width and height of Area rectangle.

Mode [0x00..0xFF]

Operating Mode of Area:

0x00 AOM_OFF: For this Area there are no coordinate messages created. However, in case the AddExit Flag is set, a message is issued when exiting the Area.

0x01 AOM_ENTER: Reports only the interruption of the Area.

0x02 AOM_TRACK: Reports an interruption and each further change of the coordinates within the Area

0x03 AOM_CONT: Reports the first interruption of the Area and afterwards the current position in regular time intervals (refer to SETCONTTIME, 0xCA).

Flags [0x00..0xFF]

The Flags listed below can modify the Operating Mode of the Area:

0x01 AOF_ADDEXIT: Reports the exiting of the Area.

0x02 AOF_ADDCOORD: Reports the coordinate messages of the touch spot.

0x04 AOF_ADDPRESS: Issues a message if the pressure exerted onto the front screen either exceeds or falls below the limit value that was predetermined for this Area.

0x08 AOF_PRESSALWAYS: For all Area messages to be received, a certain pressure has to be exceeded.

0x10 AOF_PRESSENTER: For the very first Area message a certain pressure has to be exceeded. For any additional messages a regular interruption of the Touch Zone is sufficient.

0x20 AOF_PRESSLOCAL: There is a local pressure value used for this Area. If this Flag is not explicitly declared, the pressure value of AREA0 is used.

0x40 AOF_EXTENDED: Enables the "N-Key-Rollover" emulation for this Area. In this case messages of other Areas will not be issued until the interrupted Touch Zone of this Area is not completely released yet.

0x80 AOF_ACTIVE: The Area is activated. Only active Areas create messages.

Press [0x00..0xFF]

Pressure sensitivity of this Area.

AreaPage

0x23 *page*

Returns the currently active Area Page.

Parameters:

Page [0x00..0xFF]

Number of currently active Area Page.

BeamMinMax

0x40 *MinX MaxX MinY MaxY*

Returns the minimum and maximum permitted width of a valid interruption.

Parameters:

MinX [0x00..0xFF]

Minimum number of X-beams interrupted next to each other.

MaxX [0x00..0xFF]

Maximum number of X-beams interrupted next to each other.

MinY [0x00..0xFF]

Minimum number of Y-beams interrupted next to each other.

MaxY [0x00..0xFF]

Maximum number of Y-beams interrupted next to each other.

BeamTimeout

0x41 *TimeLo TimeHi*

Returns the blank-out time for continuously interrupted beams.

Parameters:

TimeHi [0x00..0xFF]

TimeLo [0x00..0xFF]

Blank-out time in seconds.

ContTime

0x42 *TimeLo TimeHi*

Returns the time interval between two COORD messages in the Continuous Mode.

Parameters:

TimeHi [0x00..0xFF]

TimeLo [0x00..0xFF]

Time interval between messages in milliseconds.

Coord

0x19 *[X1Lo X1Hi Y1Lo Y1Hi] [X2Lo X2Hi Y2Lo Y2Hi] [NumberLo [NumberHi]]*

Reports the position of an interruption.

Parameters:

X1Lo [0x00..0xFF]

X1Hi [0x00..0xFF]

Y1Lo [0x00..0xFF]

Y1Hi [0x00..0xFF]

Coordinates of the first interruption. In case the interrupted Area does not have a set ADDCOORD Flag, there are no coordinates reported.

X2Lo [0x00..0xFF]

X2Hi [0x00..0xFF]

Y2Lo [0x00..0xFF]

Y2Hi [0x00..0xFF]

Coordinates of the second interruption.

NumberLo [0x00..0xFF]

NumberHi [0x00..0xFF]

Number of the interrupted Area. If no Area number is transmitted, then there is an interruption at AREA0. If the number of the interrupted Area is smaller than 256, *NumberHi* is not transmitted and can be assumed to be 0.

DozeMode

0x1D *Mode*

Reports either the activation or deactivation of the Doze-Mode.

Parameters:

Mode [0x00..0x01]

0x00 Doze-Mode was deactivated.

0x01 Doze-Mode was activated.

DozeModeState

0x78 *Mode TimeLo TimeHi ScanLo ScanHi*

Provides information about the function of the Doze-Mode.

Parameters:

Mode [0x00..0x83]

Behaviour of the Doze-Mode

0x00 No message at either activation or deactivation

0x01 Message at activation

0x02 Message at deactivation

0x03 Message at activation and deactivation

0x80 If this bit is set, the IRT is in Doze-Mode.

TimeLo [0x00..0xFF]

TimeHi [0x00..0xFF]

Activation time in seconds (0x0000 = always activated, 0xFFFF = always deactivated).

ScanLo [0x00..0xFF]

ScanHi [0x00..0xFF]

Time interval between two scan operations while in Doze-Mode. The time interval is set in steps of milliseconds.

DualTouchError

0x18

Multiple touch error.

Is transmitted when the corresponding behaviour was set by means of SETDUALTOUCHING (0xCB) and an invalid multiple touching was detected.

DualTouching

0x43 *Mode*

Behaviour at dual touching.

Response to the command GETDUALTOUCHING (0xC3). The parameter *Mode* reflects the currently set behaviour at dual touching. The listed values are bitwise ORed by the IRT.

0x00 Dual touching is disregarded.

0x01 Invalid multiple touching results in a DualTouchError message.

0x02 A second touch is also reported.

ErrorReport

0x30 *Selection ...*

Response to the request of an error report by means of the command GETERRORS (0xB0). The length of the report and the transmitted data depend on the kind of report.

Parameters:

Selection

Kind of error report

0x00	no error report available
0x01	INITIALERRORS: All errors that occurred after a reset and were detected by the Built-in-Test.
0x02	DEFECTBEAMS: Numbers of the blanked-out beams.
0x04	COMMUNICATIONS: Transmission error (overrun, protocol violation, ...)
0x08	COMMANDSTATUS: Result of the last transmitted command prior to the command GetErrors.

Initialization error (0x01):

FlagsLL FlagsML FlagsMH FlagsHH

A 32 bit error status is transmitted. Each set bit represents one detected error. The error bits can only be erased by a new initialization of the IRT.

0x00000001	System manager module, check sum error
0x00000002	System manager module, initialization error
0x00000004	Hardware driver module, check sum error
0x00000008	Hardware driver module, initialization error
0x00000010	Process module, check sum error
0x00000020	Process module, initialization error
0x00000040	Protocol module, check sum error
0x00000080	Protocol module, initialization error
0x00000100	During the initialization one ore more beams were interrupted
0x00000200	PSU listed in the hardware parameters, however, PSU either not available or not functional.
0x00000400	CPU test faulty.
0x00000800	Internal RAM test faulty.
0x00001000	External RAM test faulty.

Defective beams (0x02):

NxLo NxHi NyLo NyHi LsX... LsY...

Quantity and numbers of blanked-out X- and Y-beams are reported.

Transmission error (0x04):

FlagsLL FlagsML FlagsMH FlagsHH

A 32 bit error status is transmitted. Each set bit represents one detected error.

0x00000001	More DC2 than DC4 received.
0x00000002	More DC2 than DC4 received.
0x00000004	Non-control character received out of a DC2/DC4 block.
0x00000008	Control character (except SYN, XON or XOFF) received within a DC2/DC4 block.
0x00000010	Receiver overflow
0x00000020	Faulty serial data frame received (Framing Error).
0x00000040	Data parity error
0x00000080	XOFF timeout. The IRT received a XOFF, however, it did not receive a XON within the predetermined time interval.
0x00000100	Command buffer overflow.
0x00000200	Overflow of the serial receive buffer.

Command error (0x08):

FlagsLL FlagsML FlagsMH FlagsHH

A 32 bit error status is transmitted. Each set bit represents one detected error. The error bits always refer to the last transmitted command prior to the command GetErrors, only.

0x00000001	Unknown command
0x00000002	Wrong number of parameters
0x00000004	One parameter violated the value range

Exit

0x1A *[X1Lo X1Hi Y1Lo Y1Hi] [X2Lo X2Hi Y2Lo Y2Hi] [NumberLo [NumberHi]]*

Reports the position when exiting the IRT. This report is transmitted only if the ADDEXIT Flag is set.

Parameters:

X1Lo [0x00..0xFF]
X1Hi [0x00..0xFF]
Y1Lo [0x00..0xFF]
Y1Hi [0x00..0xFF]

Coordinate at which the first touch has left the Touch Zone. If the ADDCOORD Flag of the respective Area is not set, no coordinates are transmitted.

X2Lo [0x00..0xFF]
X2Hi [0x00..0xFF]
Y2Lo [0x00..0xFF]
Y2Hi [0x00..0xFF]

Coordinate at which the second touch has left the Touch Zone.

NumberLo [0x00..0xFF]
NumberHi [0x00..0xFF]

Number of the interrupted Area. If no Area number is transmitted, then there is an interruption at AREA0. If the number of the interrupted Area is smaller than 256, *NumberHi* is not transmitted and can be assumed to be 0.

FreeAreaSpace

0x24 *SpaceLo SpaceHi*

Reports the number of Areas that are still available to be defined.

Parameters:

SpaceHi [0x00..0xFF]
SpaceLo [0x00..0xFF]

Number of Areas that are still available to be defined.

FreeMacroSpace

0x63 *SpaceLo SpaceHi*

Reports the still available macro memory.

Parameters:

SpaceHi [0x00..0xFF]
SpaceLo [0x00..0xFF]

Available macro memory in bytes.

Continued for compatibility reasons with previous CTS protocols. Report is not sent by this touch generation.

Hardware

0x31 *Selection ...*

Provides information about the actual IRT hardware.

Parameters:

Selection [0x00..0x07]

This parameter determines the kind of the Hardware report. The length of the report and the transmitted data depend on the kind of parameters. In case more than one hardware report was requested at the same time, the constants listed below can be ORed together and the respective data belonging to it are lined up in the order listed below.

0x01 BEAMCOUNT: Number of X/Y-beams
0x02 PRESSURESENSORS: Number of pressure sensors
0x04 PERIPHERALS: Available peripherals

BeamCount (0x01):

BeamXlo BeamsXhi BeamsYlo BeamsYhi

Reports the number of physically present X- and Y-beams.

PressureSensors (0x02):

SensorCount

Reports the number of pressure sensors.

Peripherals (0x04):

FlagsLL FlagsML FlagsMH FlagsHH

Provides a 32 bit vector containing the available peripherals of IRT. Each set bit represents one available peripheral unit:

0x00000004 /OC_OUT0 output available (never set for 3RD generation IRTs)
0x00000008 OC_PWM output available
0x00000010 Sound source available
0x00000040 Run-LED available
0x00000080 GP_IN input available

HWVersion

0x39 *Selection ...*

Provides information about the IRT hardware versions.

Parameters:

Selection [0x00..0x07]

This parameter determines the kind of the HWVersion report. The length of the report and the transmitted data depend on the kind of parameters. In case more than one hardware report was requested at the same time, the constants listed below can be ORed together and the respective data belonging to it are lined up in the order listed below.

0x01 SILICONSERIALNUMBER: The unique silicon serial number.
0x02 HARDASSY: The hardcoded assembly number.
0x04 FPGAREVISION: The revision of the FPGA-configuration data stream.

SiliconSerialNumber (0x01):

SSNoLoLo SSNoLoHi SSNoHiLo SSNoHiHi

The unique silicon serial number of this IRT as a 32-bit unsigned value, transmitted LSB first.

HardwareCode (0x02):

HWCode

This is an assembly dependent hardware code that is used during manufacturing tests to identify the IRT version.

FPGAVersion (0x04):

A zero-terminated ASCII string with the following format is transmitted:
„*VVVVVVVV_TT.MM.YYYY_HH:MM:SS*“ (the ‘_’ represents a blank)
Meaning if the individual fields:

V Name of the configuration data stream.
TT Day of creation
MM Month of creation
YYYY Year of creation
HH Hour of creation
MM Minute of creation
SS Second of creation

Idle

0x34

Is transmitted as response to the command GETSINGLEMESSAGE (0xB4), if there are no messages in the buffer.

OemString

0x38 *Char0 Char1 ...*

Provides the OEM string with the serial number.

Parameters:

Char0 Char1 ... [0x01..0xFF]

The OEM string. The OEM string represents a byte order within the value range of 0x01 up to 0xff. The string end is marked by the value 0x00.

Origin

0x44 *Position*

Provides the current position of the coordinates origin.

Parameters:

Position [0x00..0x03]

- **0x00** TOPLEFT: Origin set to the left-hand side top corner.
- **0x01** TOPRIGHT: Origin set to the right-hand side top corner.
- **0x02** BOTTOMRIGHT: Origin set to the right-hand side bottom corner.
- **0x03** BOTTOMLEFT: Origin set to the left-hand side bottom corner.

PolyAreaDef

0x2A *NumberLo NumberHi CountLo CountHi X_0Lo X_0Hi Y_0Lo Y_0Hi X_1Lo X_1Hi Y_1Lo Y_1Hi X_2Lo X_2Hi Y_2Lo Y_2Hi [... X_nLo X_nHi Y_nLo Y_nHi] [Page [Mode [Flags [Press]]]]*

Returns a polygonal Area definition.

Parameters:

NumberLo [0x00..0xFF]

NumberHi [0x00..0xFF]

NumberLo + 256 × *NumberHi* is the number of the polygonal Area.

CountLo [0x00..0xFF]

CountHi [0x00..0xFF]

CountLo + 256 × *CountHi* is the number of polygone edges that will follow.

X_nLo [0x00..0xFF]

X_nHi [0x00..0xFF]

X_nLo + 256 × *X_nHi* is the X-coordinate of a polygone edge.

Y_nLo [0x00..0xFF]

Y_nHi [0x00..0xFF]

Y_nLo + 256 × *Y_nHi* is the Y-coordinate of a polygone edge.

Page [0x00..0xFF]

Number of Page the Area is defined on.

Mode [0x00..0xFF]

Operating Mode of the Area.

- **0x00** AOM_OFF: For this Area no coordinate messages are created. However, in case the AddExit Flag is set, a message is issued when exiting the Area.
- **0x01** AOM_ENTER: Reports only the interruption of the Area.
- **0x02** AOM_TRACK: Reports an interruption and each further change of the coordinates within the Area.
- **0x03** AOM_CONT: Reports the first interruption of the Area and afterwards the current position in regular time intervals (refer to SETCONTTIME, 0xCA).

Flags [0x00..0xFF]

The Operating Mode modification flags of the Area.

- **0x01** AOF_ADDEXIT: Reports the exiting of the Area.
- **0x02** AOF_ADDCOORD: Reports the coordinate messages of the touch spot.
- **0x04** AOF_ADDPRESS: Issues a message if the pressure exerted onto the front screen either exceeds or falls below the limit value that was predetermined for this Area.

0x08 AOF_PRESSALWAYS: For all Area messages to be received, a certain pressure has to be exceeded.
0x10 AOF_PRESSENTER: For the very first Area message a certain pressure has to be exceeded. For any additional messages a regular interruption of the Touch Zone is sufficient.
0x20 AOF_PRESSLOCAL: There is a local pressure value used for this Area. If this Flag is not explicitly declared, the pressure value of AREA0 is used.
0x40 AOF_EXTENDED: Enables the "N-Key-Rollover" emulation for this Area. In this case messages of other Areas will not be issued until the interrupted Touch Zone of this Area is not completely released yet.
0x80 AOF_ACTIVE: The Area is activated. Only active Areas create messages.

Press [0x00..0xFF]
Pressure sensitivity of this Area.

Port

0x70 *Selection StateXX...*
Reports the status of either an input or output port of the IRT.

Parameters:

Selection [0x00..0xFF]
Determines the respective port. In case the status of more than one peripheral port was requested at the same time, the constants listed below can be bitwise ORed and the status values of the individual ports are transmitted one after the other in the order listed below. Possible peripheral ports:
0x08 GP_IN input

StateXX [0x00..0xFF]
Reports the status of a port.
0x00 Port is inactive
0xFF Port is active

Pressure

0x1B *Mode [NumberLo [NumberHi]]*
Reports if the pressure exerted onto the front screen either exceeds or falls below the limit value that was predetermined for a certain Area.

Parameters:

Mode [0x00..0x01]
0x00 Pressure fell below the limit value
0x01 Pressure exceeded limit value

NumberLo [0x00..0xFF]
NumberHi [0x00..0xFF]
Number of Area on which the pressure either exceeded or fell below the limit value. If AREA0 is concerned, no number is transmitted. If the number of the concerned Area is smaller than 256, only *NumberLo* is transmitted.

PWM

0x71 *Active Sleep*
Reports the set mark-to-space ratio of the PWM output. Higher values result in longer pulse widths.

Parameters:

Active [0x00..0xFF]
Duty cycle in normal operation (Sleep-Mode inactive).

Sleep [0x00..0xFF]
Duty cycle while in Sleep-Mode.

Resolution

0x45 *XLo XHi YLo YHi*

Reports the value range of coordinates messages.

Parameters:

XLo	[0x00..0xFF]
XHi	[0x00..0xFF]
YLo	[0x00..0xFF]
YHi	[0x00..0xFF]

Maximum coordinates in X- or Y-direction. The minimum coordinate always equals 0.

Revision

0x32 *Selection Data...*

Version report. The data depend on the kind of version report.

Parameters:

Selection [0x01..0x3f]

Kind of version report. In case several version reports were requested at the same time, the constants listed below can be bitwise ORed and the report data are lined up in the order listed below.

0x01	Version number of the system manager module
0x02	Version number of the hardware driver module
0x04	Version number of the process module
0x08	Version number of the protocol module
0x10	Version number of the hardware parameters module
0x20	Designation of the IRT hardware

System manager version (0x01):
Hardware driver version (0x02):
Process module version (0x04):
Protocol module version (0x08):
Hardware parameters module version (0x10):

A zero-terminated ASCII string with the following format is transmitted:
„*V.HH.RRR.X_TT.MM.YYYY_HH:MM*“ *(*the '_' represents a blank)
Meaning if the individual fields:

V	Main version, all modules holding the same main version number are compatible to each other.
HH	Indicates the IRT hardware that is required for the module in an encoded form („00“ means hardware independent).
RRR	Indicates the revision number of the module.
X	Indicates the release status of the module:

' ' = Final
'B' = Beta
'A' = Alpha
'D' = Debug
'T' = Internal test software

TT	Day of last modification
MM	Month of last modification
YYYY	Year of last modification
HH	Hour of last modification
MM	Minute of last modification

IRT hardware (0x20):

ASCII characters with the following format are transmitted:

Dsc1..Dsc32	Hardware designation (exactly 32 characters, filled up with zero characters)
Asy1..Asy16	ASSY number (exactly 16 characters, filled up with zero characters)

Scanning

0x46 *Mode*

Permanent scanning of the beams either on or off.

Parameters:

Mode [0x00..0x01]

0x00 Scanning off

0x01 Scanning on

Setup

0x33 *Selection*

Provides information about the saved Setup parameter sets.

Parameters:

Selection [0x00..0xFF]

Each set bit represents one saved Setup parameters set.

0x01 SERIALSETUP: Baud rate, parity, emulation mode

0x04 AREADEFINITIONS: Currently defined Areas and the active Area Page

0x08 PERIPHERALSETTINGS: PWM, /OUT0, TouchSaver

0x10 COORDINATESETTINGS: Minimum and maximum width of the interruption, blank-out time for defective beams, Continuous Time, behaviour in case of multiple touching, coordinates origin, coordinates resolution, number of scans per coordinate, data transmission.

SleepMode

0x1C *Mode*

Reports either the activation or deactivation of the Sleep-Mode.

Parameters:

Mode [0x00..0x01]

0x00 Sleep-Mode was deactivated.

0x01 Sleep-Mode was activated.

SleepModeState

0x73 *Mode TimeLo TimeHi ScanLo ScanHi*

Provides information about the function of the Sleep-Mode.

Parameters:

Mode [0x00..0x83]

Behaviour of the Sleep-Mode

0x00 No message at either activation or deactivation

0x01 Message at activation

0x02 Message at deactivation

0x03 Message at activation and deactivation

0x80 If this bit is set, the IRT is in Sleep-Mode.

TimeLo [0x00..0xFF]

TimeHi [0x00..0xFF]

Activation time in seconds (0x0000 = always activated, 0xFFFF = always deactivated).

ScanLo [0x00..0xFF]

ScanHi [0x00..0xFF]

Time interval between two scan operations while in Sleep-Mode. The time interval is set in steps of milliseconds.

Sound

0x72 *FreqLo FreqHi TimeLo TimeHi*

Provides information about the sound emission.

Parameters:

FreqLo [0x00..0xFF]
FreqHi [0x00..0xFF]

Frequency of the most recently emitted beep in Hertz.

TimeLo [0x00..0xFF]
TimeHi [0x00..0xFF]

Remaining duration of the beep in milliseconds (0x0000=no beep emission, 0xFFFF = continuous beep emission).

TouchTime

0x50 *Time*

Minimum duration of a valid interruption before it is reported to the host computer.

Parameters:

Time [0x00..0xFF]

Minimum duration of the interruption in steps of 10 milliseconds.

Transmission

0x47 *Mode TimeLo TimeHi*

Provides information about the settings for the data transmission and the flow control.

Parameters:

Mode [0x00..0xFF]

- **0x00** Data transmission disabled
- **0x01** Data transmission enabled
- **0x10** Flow control for transmission (**IRT**→ host)
- **0x20** Flow control for receive (host→**IRT**)

TimeLo [0x00..0xFF]
TimeHi [0x00..0xFF]

XOFF timeout in milliseconds.

3.5 Alphabetical summary

The following charts provide an alphabetical summary of all commands, reports and messages.

3.5.1 Commands

Commands are transmitted from the host computer to the IRT. They either determine a certain behaviour of the IRT or initiate a definite function.

Command	Identification	Description	Page
BREAK	100 ms TTL-Low applied to RxD	Resets the IRT. A possibly saved Setup will be disregarded.	25
ClearArea	0xA0	Clears either one or more Area definitions	25
ClearMacro	0xE0	Clears either one or more macro definitions	25
DefineArea	0xA1	Defines a rectangular Area	26
DefPolyArea	0xAA	Defines a polygonal Area	27
DestroySetup	0x84	Destroys a saved Setup	27
EndMacroRecord	0xE1	Ends macro recording	27
ExecMacro	0xE2	Executes a macro	28
GetAreaDef	0xA2	Requests the definition of an Area	28
GetAreaPage	0xA3	Requests the currently set Area Page	28
GetBeamMinMax	0xC0	Requests the minimum and maximum number of interrupted beams for a valid interruption	28
GetBeamTimeout	0xC1	Requests the blank-out time for defective beams	28
GetContTime	0xC2	Requests the time interval between two Continuous messages	28
GetDozeMode	0xF8	Requests the current Doze-Mode status	28
GetDualTouching	0xC3	Requests the behaviour in case of dual touching	28
GetErrors	0xB0	Requests an error report	29
GetFreeAreaSpace	0xA4	Requests the available number of Areas	29
GetFreeMacroSpace	0xE3	Requests the available memory for macros	29
GetHardware	0xB1	Requests a report about the IRT hardware	29
GetHWVersions	0xB9	Request the hard coded versions	29
GetOemString	0xB8	Requests the OEM string with the serial number	29
GetOrigin	0xC4	Requests the coordinates origin	29
GetPort	0xF0	Requests the current status of the input port	30
GetPWM	0xF1	Requests the current PWM settings	30
GetResolution	0xC5	Requests the coordinates resolution	30
GetRevisions	0xB2	Requests a version report	30
GetScanning	0xC6	Requests whether or not the beams are to be scanned	30
GetSetup	0xB3	Requests information about saved Setup parameter sets	30
GetSingleMessage	0xB4	Requests a single message	30
GetSingleScan	0xB5	Initiates a scan operation and provides the result	30
GetSleepMode	0xF3	Requests the current Sleep-Mode status	31
GetSound	0xF2	Requests the current status of the sound source	31
GetTouchTime	0xD0	Requests the minimum duration for a valid interruption	31
GetTransmission	0xC7	Requests the behaviour set by means of SetTransmission	31
Reset_CTS	0x81	Initializes the IRT in the CTS protocol, at the same time warm start for an already linked IRT	31
SaveSetup	0x83	Saves current settings to the FLASH EPROM	31
SelectAreaPage	0xA5	Selects an Area Page	31
SetAreaFlags	0xA8	Changes the Operating Mode Flags of an already defined Area	33
SetAreaMode	0xA7	Changes the Operating Mode of an already defined Area	33

Command	Identification	Description	Page
SetAreaPressure	0xA9	Changes the pressure sensitivity of an already defined Area	34
SetAreaState	0xA6	Changes several operating parameters of an Area	32
SetBeamMinMax	0xC8	Sets the minimum and maximum number of interrupted beams for a valid interruption.	34
SetBeamTimeout	0xC9	Sets the blank-out time for defective beams	34
SetContTime	0xCA	Sets the time interval between two continuous messages	34
SetDozeMode	0xF9	Sets the Doze-Mode parameters	35
SetDualTouching	0xCB	Sets the behaviour in case of dual touching	35
SetOrigin	0xCC	Sets the coordinates origin	35
SetPort	0xF4	Sets an output port	36
SetPWM	0xF5	Sets the PWM output	36
SetResolution	0xCD	Sets the coordinates resolution	36
SetScanning	0xCE	Activates or deactivates the scanning of the beams	36
SetSleepMode	0xF7	Sets the Sleep-Mode parameters	37
SetSound	0xF6	Emits a beep	37
SetTouchTime	0xD1	Sets the minimum duration for a valid interruption	37
SetTransmission	0xCF	Activates or deactivates the spontaneous transmission of messages	38
SoftReset	0x80	Initiates a warm start	38
StartMacroRecord	0xE5	Starts macro recording	38

3.5.2 Messages and reports

Messages are transmitted spontaneously from the IRT to the host computer, reports only upon request.

Message	Identification	Description	Page
AreaDefinition	0x22	Area definition	39
AreaPage	0x23	Currently set Area Page	40
BeamMinMax	0x40	The minimum and maximum number of interrupted beams for a valid interruption	40
BeamTimeout	0x41	Blank-out time for continuously interrupted beams	40
ContTime	0x42	Time interval between two messages in the Continuous Mode	40
Coord	0x19	Position of the interruption	40
DozeMode	0x1D	Doze-Mode activation or deactivation	41
DozeModeState	0x78	Settings of Doze-Mode	41
DualTouchError	0x18	Dual touch error	41
DualTouching	0x43	Behaviour in case of dual touching	41
ErrorReport	0x30	Error report	42
Exit	0x1A	Exit position	43
FreeAreaSpace	0x24	Available memory for Area definitions	43
FreeMacroSpace	0x63	Available memory for macro definitions	43
Hardware	0x31	IRT hardware description	44
HWVersion	0x39	Hard coded versions	44
Idle	0x34	No message available	45
OemString	0x38	OEM string with serial number	45
Origin	0x44	Information about the coordinates origin	45
PolyAreaDef	0x2A	Polygonal Area definition	45
Port	0x70	Status of the input and output ports	46
Pressure	0x1B	Pressure either exceeded or fell below the limit value	46
PWM	0x71	Settings of PWM Unit	46
Resolution	0x45	Coordinates resolution	47
Revision	0x32	Versions	47

Message	Identification	Description	Page
Scanning	0x46	Scanning of beams on or off	48
Setup	0x33	Information about saved Setup parameter sets	48
SleepMode	0x1C	Sleep-Mode activation or deactivation	48
SleepModeState	0x73	Settings of Sleep-Mode	48
Sound	0x72	Current status of sound source	48
TouchTime	0x50	Minimum duration of a valid interruption	49
Transmission	0x47	Spontaneous data transmission on/off and flow control	49

3.6 Numerical summary

The following charts provide a numerical summary of all commands, reports and messages.

3.6.1 Commands

Commands are transmitted from the host computer to the IRT. They either determine a certain behaviour of the IRT or initiate a definite function.

Command	Identification	Description	Page
BREAK	100 ms TTL-Low applied to RxD	Resets the IRT. A possibly saved Setup will be disregarded.	25
SoftReset	0x80	Initiates a warm start	38
Reset_CTS	0x81	Initializes the IRT in the CTS protocol, at the same time warm start for an already linked IRT	31
SaveSetup	0x83	Saves current settings to the FLASH EPROM	31
DestroySetup	0x84	Destroys a saved Setup	27
ClearArea	0xA0	Clears either one or more Area definitions	25
DefineArea	0xA1	Defines a rectangular Area	26
GetAreaDef	0xA2	Requests the definition of an Area	28
GetAreaPage	0xA3	Requests the currently set Area Page	28
GetFreeAreaSpace	0xA4	Requests the available number of Areas	29
SelectAreaPage	0xA5	Selects an Area Page	31
SetAreaState	0xA6	Changes several operating parameters of an Area	32
SetAreaMode	0xA7	Changes the Operating Mode of an already defined Area	33
SetAreaFlags	0xA8	Changes the Operating Mode Flags of an already defined Area	33
SetAreaPressure	0xA9	Changes the pressure sensitivity of an already defined Area	34
DefPolyArea	0xAA	Defines a polygonal Area	27
GetErrors	0xB0	Requests an error report	29
GetHardware	0xB1	Requests a report about the IRT hardware	29
GetRevisions	0xB2	Requests a version report	30
GetSetup	0xB3	Requests information about saved Setup parameter sets	30
GetSingleMessage	0xB4	Requests a single message	30
GetSingleScan	0xB5	Initiates a scan operation and provides the result	30
GetOemString	0xB8	Requests the OEM string with the serial number	29
GetHWVersions	0xB9	Request the hard coded versions	29
GetBeamMinMax	0xC0	Requests the minimum and maximum number of interrupted beams for a valid interruption	28
GetBeamTimeout	0xC1	Requests the blank-out time for defective beams	28
GetContTime	0xC2	Requests the time interval between two Continuous messages	28
GetDualTouching	0xC3	Requests the behaviour in case of dual touching	28
GetOrigin	0xC4	Requests the coordinates origin	29
GetResolution	0xC5	Requests the coordinates resolution	30

Command	Identification	Description	Page
GetScanning	0xC6	Requests whether or not the beams are to be scanned	30
GetTransmission	0xC7	Requests the behaviour set by means of SetTransmission	31
SetBeamMinMax	0xC8	Sets the minimum and maximum number of interrupted beams for a valid interruption.	34
SetBeamTimeout	0xC9	Sets the blank-out time for defective beams	34
SetContTime	0xCA	Sets the time interval between two continuous messages	34
SetDualTouching	0xCB	Sets the behaviour in case of dual touching	35
SetOrigin	0xCC	Sets the coordinates origin	35
SetResolution	0xCD	Sets the coordinates resolution	36
SetScanning	0xCE	Activates or deactivates the scanning of the beams	36
SetTransmission	0xCF	Activates or deactivates the spontaneous transmission of messages	38
GetTouchTime	0xD0	Requests the minimum duration for a valid interruption	31
SetTouchTime	0xD1	Sets the minimum duration for a valid interruption	37
ClearMacro	0xE0	Clears either one or more macro definitions	25
EndMacroRecord	0xE1	Ends macro recording	27
ExecMacro	0xE2	Executes a macro	28
GetFreeMacroSpace	0xE3	Requests the available memory for macros	29
StartMacroRecord	0xE5	Starts macro recording	38
GetPort	0xF0	Requests the current status of the input port	30
GetPWM	0xF1	Requests the current PWM settings	30
GetSound	0xF2	Requests the current status of the sound source	31
GetSleepMode	0xF3	Requests the current Sleep-Mode status	31
SetPort	0xF4	Sets an output port	36
SetPWM	0xF5	Sets the PWM output	36
SetSound	0xF6	Emits a beep	37
SetSleepMode	0xF7	Sets the Sleep-Mode parameters	37
GetDozeMode	0xF8	Requests the current Doze-Mode status	28
SetDozeMode	0xF9	Sets the Doze-Mode parameters	35

3.6.2 Reports and messages

Messages are transmitted spontaneously from the IRT to the host computer, reports only upon request.

Message	Identification	Description	Page
AreaDefinition	0x22	Area definition	39
DualTouchError	0x18	Dual touch error	41
Coord	0x19	Position of the interruption	40
Exit	0x1A	Exit position	43
Pressure	0x1B	Pressure either exceeded or fell below the limit value	46
SleepMode	0x1C	Sleep-Mode activation or deactivation	48
DozeMode	0x1D	Doze-Mode activation or deactivation	41
AreaPage	0x23	Currently set Area Page	40
FreeAreaSpace	0x24	Available memory for Area definitions	43
PolyAreaDef	0x2A	Polygonal Area definition	45
ErrorReport	0x30	Error report	42
Hardware	0x31	IRT hardware description	44
Revision	0x32	Versions	47
Setup	0x33	Information about saved Setup parameter sets	48
Idle	0x34	No message available	45
OemString	0x38	OEM string with serial number	45
HWVersion	0x39	Hard coded versions	44

Message	Identification	Description	Page
BeamMinMax	0x40	The minimum and maximum number of interrupted beams for a valid interruption	40
BeamTimeout	0x41	Blank-out time for continuously interrupted beams	40
ContTime	0x42	Time interval between two messages in the Continuous Mode	40
DualTouching	0x43	Behaviour in case of dual touching	41
Origin	0x44	Information about the coordinates origin	45
Resolution	0x45	Coordinates resolution	47
Scanning	0x46	Scanning of beams on or off	48
Transmission	0x47	Spontaneous data transmission on/off and flow control	49
TouchTime	0x50	Minimum duration of a valid interruption	24
FreeMacroSpace	0x63	Available memory for macro definitions	43
Port	0x70	Status of the input and output ports	46
PWM	0x71	Settings of PWM Unit	46
Sound	0x72	Current status of sound source	48
SleepModeState	0x73	Settings of Sleep-Mode	48
DozeModeState	0x78	Settings of Doze-Mode	41

4 Default values

The following charts contain the default values after either a successful initialization or a reset dependent on the used protocol.

4.1 CTS protocol

Parameters	Default	Remark
Scanning	OFF	or depending on coord. setup
Report format	coordinates	fixed
Operating Mode AREA0	AOM_ENTER	or depending on Area setup
Add Exit Point	OFF	or depending on coord. setup
Data transmission	OFF	or depending on coord. setup
Serial flow control	none	or depending on coord. setup
Origin (zero point)	left-hand side top corner	or depending on coord. setup
Blank-out time for defective beams	10 s	or depending on coord. setup
Macros	none	
Areas	only AREA0, maximum size	or depending on Area setup
Max. edges per polygone	64	fixed
Active Area Page	0	or depending on Area setup
Min. interrupted beam	X=1, Y=1	or depending on coord. setup
Max. interrupted beam	X=5, Y=5	or depending on coord. setup
Min. duration of interruption	20 ms	or depending on coord. setup
Continuous Time	55 ms	or depending on coord. setup
Multiple touching	disregarded	or depending on coord. setup
PWM output	constant HIGH	or depending on periph. setup
Coordinates resolution	X=640, Y=480	or depending on coord. setup
Doze-Mode	after 10 s, scan each 25 ms	or depending on periph. setup
Sleep-Mode	disabled, scan each 500 ms	or depending on periph. setup

4.2 Carrol Touch Emulation

Parameters	Default	Remark
Scanning	OFF	
Report format	coordinates	
Operating Mode AREA0	tracking	
Add Exit Point	OFF	
Data transmission	OFF	
Serial flow control	none	fixed
Origin (zero point)	left-hand side top corner	
Blank-out time for defective beams	10 s	fixed

5 Technical specifications

5.1 Electrical Specs

Voltage: +5.0 V_{DC} (±5%)

IRT	Current (typical operating)	Current (worst case)
IRT65-V3.2	135 mA_{RMS}	155 mA_{RMS}
IRT84-V2.0	135 mA_{RMS}	170 mA_{RMS}
IRT104-V5.x	135 mA_{RMS}	170 mA_{RMS}
IRT121-V3.x	140 mA_{RMS}	215 mA_{RMS}
IRT151-V2.x	150 mA_{RMS}	215 mA_{RMS}
IRT170-V1.0	150 mA_{RMS}	245 mA_{RMS}
IRT180-V1.1	210 mA_{RMS}	280 mA_{RMS}
IRT190-V1.0	215 mA_{RMS}	295 mA_{RMS}

5.2 Communication Specs

Communication	Bi-directional, asynchronous, EIA-232-D and TTL
Baud Rate, Parity	1200 to 57600 bps. Automatic baud rate and parity detection
Protocol	XON/XOFF

5.3 Operational Specs

Active Touch Area	
IRT65-V3.2	132.1 x 96.5 mm²
IRT84-V2.0	172,7 x 127.0 mm²
IRT104-V5.x	208,3 x 157,5 mm²
IRT121-V3.x	243,8 x 182,9 mm²
IRT151-V2.x	309,9 x 233,7 mm²
IRT170-V1.0	340,4 x 274,3 mm²
IRT180-V1.1	365,8 x 294,6 mm²
IRT190-V1.0	381,0 x 305,0 mm²

Touchpoint Density	16 tps/cm²; 64 tps/cm² interpolated
Response time	20 ms
Touch Points	Simultaneous tracking of two touchpoints
Stylus Diameter	≥ 6 mm, >8 mm for interpolation
Touch Modes	Up to 71 polygonal areas with individual touch modes: enter, exit, tracking, continuous, Z-Press. Up to 16 area pages
Software Drivers	MS-DOS, Win3.11/9x/Me/NT/2000/XP/CE, Windows 7, Windows 8, Windows 8.1, XFree86, X.Org
Diagnostics	Complete system test at power-on, cyclic beam test during operation

5.4 Environmental Specs

Operating Temperature	0°C to +70°C -20°C to +85°C (optional)
Storage Temperature	-20°C to +85°C
Humidity	90% RH @ 70°C, non-condensing
Altitude	T.B.D.
Shock (MIL-STD-810E)	T.B.D.
Vibration (MIL-STD-810E)	T.B.D.
Sealing (EN 60529)	> IP65
Peak Output Wavelenght	950 nm, infrared
Ambient light	Unaffected
Transmissivity	Up to 100%, depending on filter screen
MTBF (@25°C) MIL-HDBK-217F	> 500.000 h
EMI	EN 50081-1,2 EN 55022, Class B
ESD	EN 50082-1,2

5.5 Mechanical Specs

Total size Controller	
IRT65-V3.2	W 186.1 x H 163.0 mm^2
IRT84-V2.0	W 227.0 x H 194.0 mm^2
IRT104-V5.x	W 262.8 x H 210.0 mm^2
IRT121-V3.x	W 300.4 x H 249.0 mm^2
IRT151-V2.x	W 361.5 x H 295.0 mm^2
IRT170-V1.0	W 401.2 x H 369.6 mm^2
IRT181-V1.1	W 422.4 x H 358.1 mm^2
IRT190-V1.0	W 439.8 x H 372.1 mm^2
Maximum component height	2.5 mm
Maximum IR-Element height	5.5 mm
Weight Controller	
IRT65-V3.2	65 g
IRT84-V2.0	80 g
IRT104-V5.x	90 g
IRT121-V3.x	105 g
IRT151-V2.x	125 g
IRT170-V1.0	130 g
IRT181-V1.1	145 g
IRT190.V1.0	150 g
Connectors	X1: 20-pin, Hirose DF14-20P X2: 5-pin, Hirose DF14-5P
Peripherals	1 opto-isolated PWM output for backlight dimming. 1 opto-isolated input for touchpoint validation or GP use.

There are 3 types of Bezels available: Standard (overall height 10.5 mm), UniBezel (height about 7 mm) and ThinLine (height about 3 mm). ThinLine IRTs are equipped with SMD-IR elements. Please ask us for actual implementation proposals.

5.6 Options

USB-Interface:	
Voltage internal	+3.3 V_{DC}
Current (worst case)	90 mA_{RMS}
Device type	Full speed, 12 Mbit/s
USB-Drivers	Windows 9x/Me/2000/XP, Windows 7, Windows 8, Windows 8.1, Linux
Z-axis:	256 levels
Audio amplifier (key click):	0.5 W @ 8 Ω

5.7 Glass Specification

Bezel is made of PMMA plastics, covers the touch controller electronics and is used as a mechanical mounting device for the glass and touch controller.
For the IRT two glass options are available: 3.0mm coated standard glass and 4.4mm coated and laminated security glass.
Glass is delivered according to following specification:

Standard glass:

Thickness:	3,0 mm (+ 0,6 mm / -0,5 mm)
Tooling:	Edges grinded, corners broken
Scratches:	5 × 0,16 mm wide with a cummulated length of 12 mm
Bubbles:	No visible bubbles
Inclusions:	3 × 0,4 mm
Coating damages:	3 × 0,6 – 1,3 mm (*(length + width)/2*)
Color:	Undefined
Spalling / flaking:	Only on edges and corners. Not visible from front
Tests according to:	DIN 10110-7
Throw Ridgity	Not specified

Security glass:

Thickness:	4,4 mm (+ 0,6 mm / -0,5 mm)
Tooling:	Edges grinded, corners broken
Scratches:	5 × 0,16 mm wide with a cummulated length of 12 mm
Bubbles:	No visible bubbles
Inclusions:	3 × 0,4 mm
Coating damages:	3 × 0,6 – 1,3 mm (*(length + width)/2*)
Color:	Undefined
Spalling / flaking:	Only on edges and corners. Not visible from front
Tests according to:	DIN 10110-7
Throw Ridgity	EN 356, Klasse P1A

6 Connector:

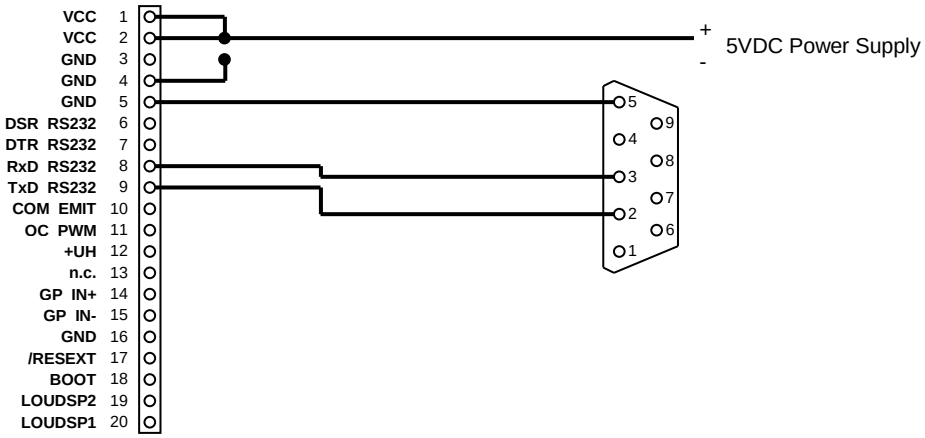
6.1 Pinout

X1/		X1/		X1/		X1/	
1	+5V	6	n.c.	11	OC_PWM	16	GND
2	+5V	7	n.c.	12	+Uh	17	/RESEXT
3	GND	8	RxD_RS232	13	n.c.	18	BOOT
4	GND	9	TxD_RS232	14	GP_IN+	19	LOUDSP1
5	GND	10	COM_EMIT	15	GP_IN-	20	LOUDSP2

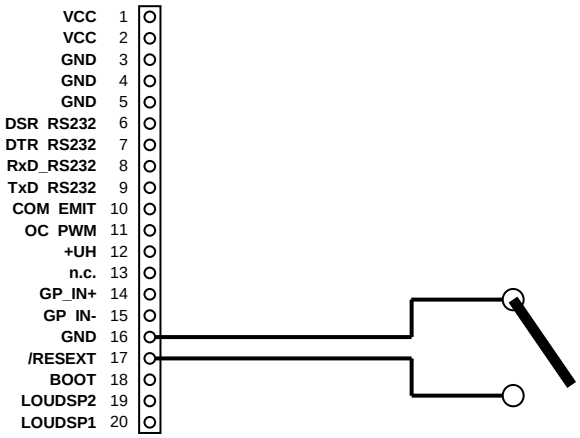
X2/	
1	USB_GND
2	USB_DATA+
3	USB_DATA-
4	+5V
5	Internal 4 V (out)

6.2 Usage

Use the following wiring diagram to connect the IRT with an EIA-232-D compatible serial port (e.g. COM-port of an IBM compatible PC)



Use the following wiring diagram to disable the IRT by using a key switch. The IRT is in a permanent reset while the switch is closed. After the switch is opened again, the IRT has the be reconnected. The Citron mouse emulation drivers for DOS and Windows handle the reconnection automatically.



6.3 Electrical Data

6.3.1 RXD_RS232

Description:

Serial data input at EIA-232-D level.

Implementation:

Corresponds to a MAX232 input with a 2.61 kΩ pull-up resistor.

Electrical Data:

Parameter	Symbol	min.	nom.	max.	Unit
Input Low Voltage	V_{IL}	-30		0.8	V
In put High Voltage	V_{IH}	2.4		30	V

6.3.2 TXD_RS232

Description:

Serial data output at EIA-232-D level.

Implementation:

Corresponds to a MAX232 output.

Electrical Data:

Parameter	Symbol	min.	nom.	max.	Unit
Output Low Voltage	V_{OL}	-5		-9	V
Output High Voltage	V_{OH}	5		9	V
Short Circuit Current	I_S		±10		mA

6.3.3 OC_PWM

Description:

PWM Output for Backlight Dimming. Here the PWM signal is provided that can be set by means of the command SETPWM (0xF5). The PWM duty cycle can be switched automatically between two individually definable levels for regular operation and sleep mode.

Implementation:

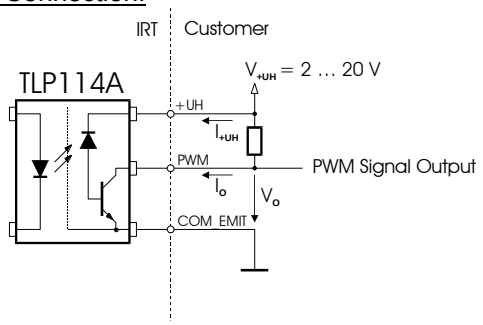
Power supply for the opto-coupler at the +UH pin.
Signal ground at the COM_EMIT pin.
PWM signal output at the open collector terminal of a TLP114A opto-coupler.

Electrical Data:

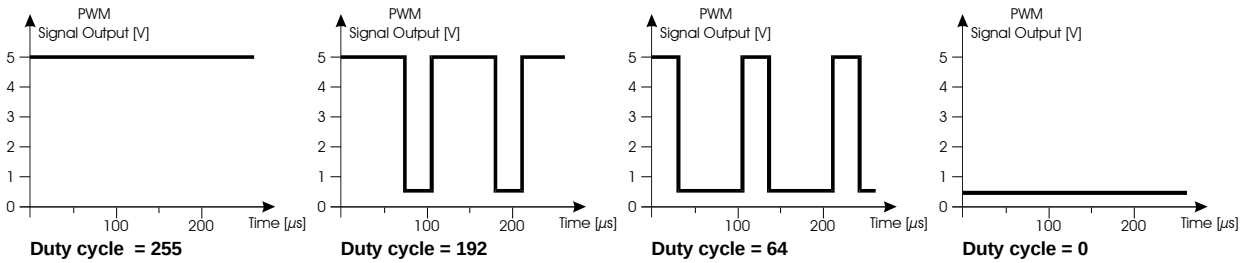
Parameter (1)	Symbol	min.	nom.	max.	Unit
Output Low Voltage	$V_{O,Lo}$ (I_O = 2.4 mA)			0.4	V
Output High Current	$I_{O,Hi}$ (V_{+UH} = 5.5 V, V_O = 5.5 V)		3	500	nA
	(V_{+UH} = 30 V, V_O = 20 V)			5	µA
Output Low Current	$I_{O,Lo}$		2.5		mA
Output Voltage (2)	V_O	-0.5		20	V
Output Power Dissipation (3)	P_O			100	mW
PWM Frequency (3) @ f_{ext} = 40 MHz			9.766		kHz
Supply Voltage	V_{+UH}	2		20	V
Supply Current (4)	I_{+UH}		0.01	1	µA

Notes:
(1) all parameters with reference to COM_EMIT
(2) absolute maximum ratings
(3) The PWM Frequency is calculated as follows: $f_{PWM} = f_{ext} / 4096$
(4) PWM output not connected

Typical Connection:



Signal Definition:



Default:

duty cycle = 255

6.3.4 GP_IN

Description:

General purpose input. Can be used for touchpoint validation.

Implementation:

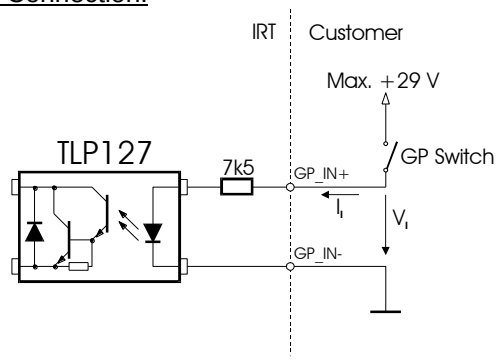
Anode of a TLP127 opto-coupler input with a 7.5 kΩ series resistor at GP_IN+. Cathode of a TLP127 opto-coupler at GP_IN-.

Electrical Data:

Parameter	Symbol	min.	nom.	max.	Unit
Input Low Voltage	$V_{I,Lo}$	-5 (1)		1	V
Input High Voltage	$V_{I,Hi}$	2.2		29	V
Input Low Current	$I_{I,Lo}$ (V_I = -5 V)			10	µA
Input High Current	$I_{I,Hi}$ (V_I = 29 V)			4	mA
	$I_{I,Hi}$ (V_I = 5 V)			0.6	mA

Notes:
(1) absolute maximum rating

Typical Connection:



Signal Level Definition:

GP Switch	Port State	touchpoint
Open	0x00	invalid
Closed	0xFF	valid

6.3.5 /RESEXT

Description:

Low active external reset input for the IRT.

Implementation:

This input is connected to the INH-pin of a MB3793-42 Reset generator with a 10 kΩ pull-up-resistor

Electrical Data:

Parameter	min	nom	max	Unit
Input Threshold		0.7		V
Input Duration	15			ms

6.3.6 /BOOT

Description:

For internal use by Citron only. **DO NOT CONNECT!**

Implementation:

Corresponds to a 74FCT244 (alt. ABT) input with a 10 kΩ pull-up resistor.

6.3.7 LOUDSPEAKER

Description:

Output of a 0.5 W amplifier for programmable frequency output to an 8 Ohm speaker.

Implementation:

LOUDSP1 and LOUDSP2 are the direct outputs of a LM4861 audio amplifier.

7 Index