

Spezifikation

Light Pen (Lichtgriffel)
für die GDP64 oder GDP64 HS



Version 1.0

Idee:

Sascha Neuschl
 Pirolweg 21
 48167 Münster
 Email: scn69@gmx.de

Dokumentenhistorie

Version	Autor(en)	Änderung	Datum
0.9	Neuschl, Sascha	Entwurf	16.02.2021
1.0	Neuschl, Sascha	Erste Version	20.02.2021

Inhaltsverzeichnis

1	Vorwort.....	4
1.1	Idee.....	4
1.2	Ansatz.....	4
1.3	Aktueller Stand	4
2	Beschreibung des Konzepts	4
2.1	Der Light Pen.....	5
2.2	Verwendung des Light Pens mit dem EF9366	5
2.3	Aufbau und Register des EF9366	6
3	Die Schaltung für den Light Pen	8
3.1	Die alte Schaltung von Graf Elektronik Systeme aus 1982 in der Beschreibung der CRT3... 8	
3.2	Die neue Schaltung	8
3.3	Anschluss des Light Pens an die GDP64.....	9
4	Stückliste.....	10
5	Test des Light Pens	11
5.1	Anzeige der X- / Y-Komponente mit einem Programm	11
6	Anwendung des Light Pens	16
6.1	Zeichnen mit dem Light Pen.....	16
6.2	Programm für Auslösen eines Controls.....	19
7	Anhang.....	26
7.1	Bausteine.....	26
7.1.1	Fototransistor SFH229.....	26
7.1.2	Komparator LT1016.....	26
7.2	Quellennachweis:	26

1 Vorwort

1.1 Idee

Schon 1983 (mit 14 Jahren) fand ich diesen Satz mit dem **Lichtgriffel** im Kapitel der GDP64 in dem Buch „Mikrocomputer selbstgebaut und programmiert“ höchst interessant. Das musste so ähnlich funktionieren wie bei den Montagsmalern mit Frank Elstner im ersten Programm. Aber damals fehlten mir die Mittel zur Umsetzung. Und letztens kam mir das mal wieder in den Sinn, und ich habe das dann schnell mal umgesetzt.

1.2 Ansatz

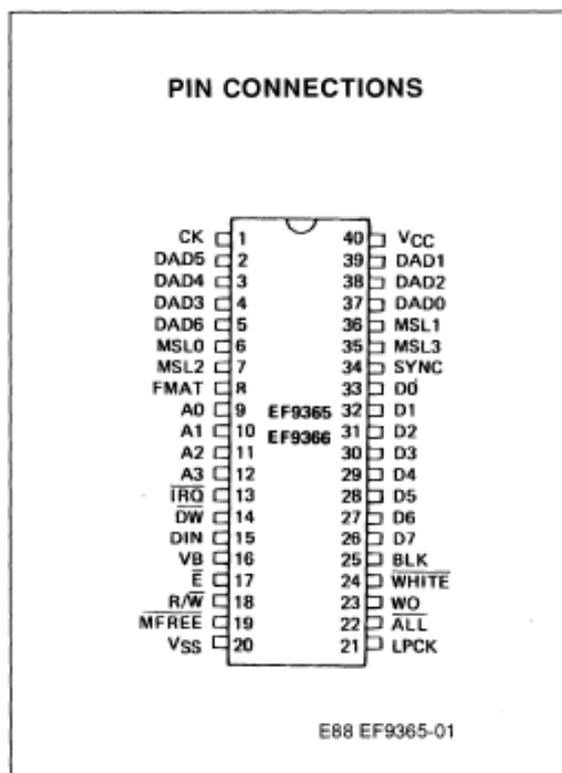
Ein Light Pen oder Lichtgriffel sollte gebaut werden, um damit zu zeichnen und damit ein Zeigergerät zu haben, um bestimmte Programmfunktionen / Controls auslösen zu können.

1.3 Aktueller Stand

Mit dem von mir gebauten Light Pen kann man in einem Programm zeichnen und in einem anderen Controls bzw. Buttons, wie bei Maus-Betrieb, auslösen.

2 Beschreibung des Konzepts

Der Grafikprozessor EF9366, der auf der GDP64 bzw. GDP64HS eingesetzt ist, besitzt einen Anschluss für einen Light Pen an Pin 21 (LPCK). Auf den GDP-Platinen ist dieser mit einem 330 Ohm Widerstand gegen 0V geschaltet.



2.1 Der Light Pen

Ein Light Pen besteht aus einem sehr schnellen Fototransistor und einem Verstärker. Mit dem Verstärker lässt sich auch die Empfindlichkeit des Light Pens einstellen. Er erzeugt ein Signal mit logisch 1, wenn ein heller Punkt auf einem **Röhrenbildschirm** erkannt wird. Bei diesem hellen Punkt befindet sich zu diesem Zeitpunkt der **Elektronenstrahl** der Bildröhre.

Der Grafikprozessor bestimmt die Position des Elektronenstrahls bzw. des Light Pens so, dass ein **Zähler** gestartet wird, wenn der **Elektronenstrahl in der oberen linken Ecke der Bildröhre** positioniert ist. Der Zähler wird gestoppt, wenn es einen 0-1-Übergang an Pin 21 (LPCK) gibt.

Effektiv wird also eine Zeitmessung durchgeführt. Das Ergebnis wird als **Bildschirmposition mit X- und Y-Komponente in jeweils einem Register** eingetragen.

Der **Bildpunkttakt beträgt 71 ns**. Zum Zeitpunkt der Entwicklung des Grafikprozessors stellte das noch ein Problem für die Schaltung des Light Pens dar, weil weder der Fototransistor mit seiner Reaktionszeit noch der Verstärker (Komparator) mit seiner Schaltzeit schnell genug waren. Heute erreicht man bei geringen Kosten für die Bauteile 10 ns.

2.2 Verwendung des Light Pens mit dem EF9366

USE OF LIGHT PEN CIRCUITRY

A rising edge on the LPCK input is used to sample the current display address in the XLP and YLP registers, provided that this edge is present in the frame immediately following loading of the 08₁₆ or 09₁₆ code into the CMD register.

Here, the frame origin is counted starting with the VB falling edge. With code 08₁₆, the WHITE output recopies the BLK signal from the frame origin up to the rising edge on the LPCK input, or when VB starts rising again, if the LPCK input remains low for the entire frame. With code 09₁₆, the WHITE output is not activated.

The YLP address is 8-bit coded since there are 256 display lines in each frame. The XLP address is 6-bit coded since there are 64 display cycles in each line.

These 6 bits are left justified in the XLP register. XLP and YLP register contents match the write address if FMAT is low (or for the EF9366), but should be multiplied by 2 if FMAT is high, so as to be able to match the write address.

The address sampled into XLP corresponds to the current memory cycle. Dots detected by the light pen were addressed in the memory during the previous cycle. Hence, 1 should be subtracted from bit 2 in XLP register where the light pen electronic circuitry does not produce any additional delay.

If the rising edge on input LPCK occurs while VB is low, then the LSB in XLP is set high. This bit acts as

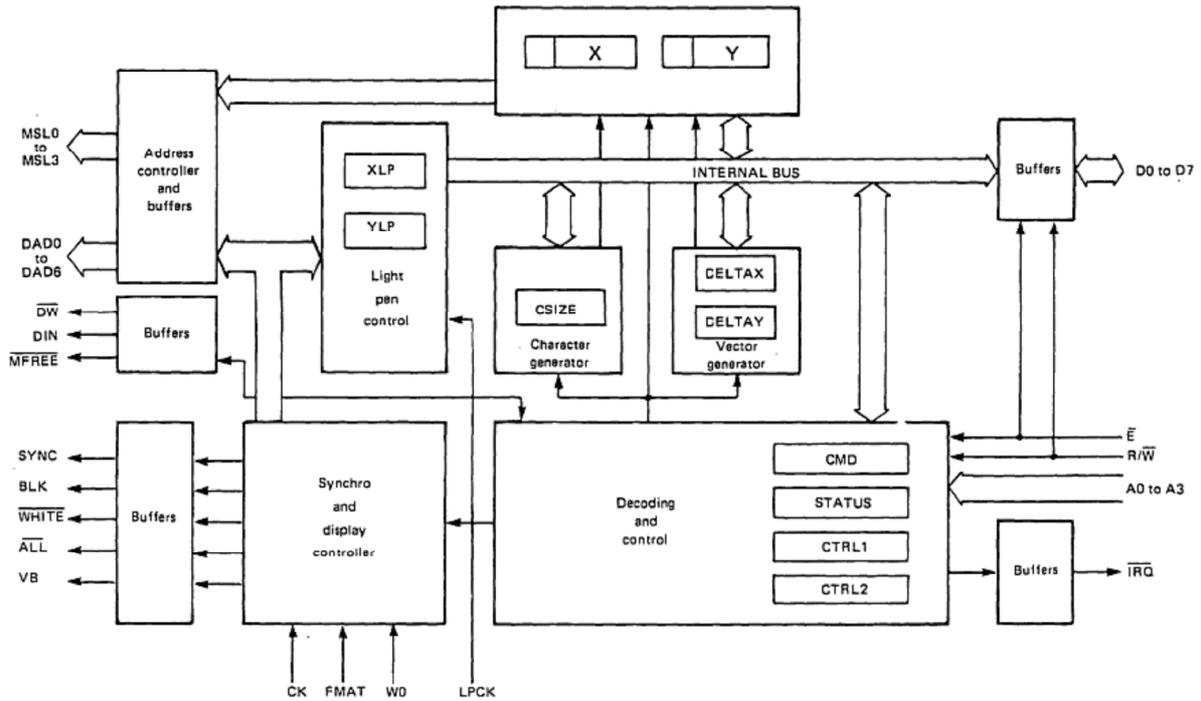
a status signal which is reset to the low state by reading register XLP or YLP.

The rising edge first received (LPCK or VB) sets bit 0 in STATUS register high. An interrupt is initiated if bit 4 bin CTRL1 is high.

When commands 08₁₆ or 09₁₆ have been decoded, bit 2 of the status register goes high (circuit ready for any further command) and bit 0 goes low (light pen operating sequence underway).

2.3 Aufbau und Register des EF9366

Aufbau des Grafikprozessors:



Registerstruktur:

Address Register					Register Functions		Number of Bits
Binary				Hexa	Read R/W = 1	Write R/W = 0	
A3	A2	A1	A0				
0	0	0	0	0	STATUS	CMD	8
0	0	0	1	1	CTRL 1 (write Control and Interrupt Control)		7
0	0	1	0	2	CTRL 2 (Vector and Symbol Type Control)		4
0	0	1	1	3	CSIZE (Character Size)		8
0	1	0	0	4	Reserved		–
0	1	0	1	5	DELTAX		8
0	1	1	0	6	Reserved		–
0	1	1	1	7	DELTAY		8
1	0	0	0	8	X MSBs		4
1	0	0	1	9	X LSBs		8
1	0	1	0	A	Y MSBs		4
1	0	1	1	B	Y LSBs		8
1	1	0	0	C	XLP (light-pen)	Reserved	7
1	1	0	1	D	YLP (light-pen)	Reserved	8
1	1	1	0	E	Reserved		–
1	1	1	1	F	Reserved		–

Reserved : These addresses are reserved for future versions of the circuit. In read mode, output buffers D0-D7 force a high state on the data bus.

NDR-Klein-Computer – Light Pen für GDP64 /GDP64HS

Command Register:

b7	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
b6	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	1
b5	0	0	1	1	0	0	1	1	0	0	1	0	0	1	1	1
b4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1
b3 b2 b1 b0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Vector Generation

(for b2, b1, b0 see small vector definition)

0 0 0 0 0	0	Set Bit 1 of CTRL 1 : Pen Selection	Space	0	@	P	`	p
0 0 0 1 1	1	Clear Bit 1 of CTRL 1 : Eraser selection	!	1	A	Q	a	q
0 0 1 0 2	2	Set Bit 0 of CTRL 1 : Pen/Eraser Down Selection	"	2	B	R	b	r
0 0 1 1 3	3	Clear Bit 0 of CTRL 1 : Pen/Eraser up Selection	#	3	C	S	c	s
0 1 0 0 4	4	Clear screen	\$	4	D	T	d	t
0 1 0 1 5	5	X and Y Registers Reset to 0	%	5	E	U	e	u
0 1 1 0 6	6	X and Y Reset to 0 and Clear Screen	&	6	F	V	f	v
0 1 1 1 7	7	Clear Screen, set CSIZE to code "minsize". All other registers reset to 0. (except XLP, YLP)	'	7	G	W	g	w

Small Vector Definition

b7	b6 b5	b4 b3	b2 b1 b0
1	\DeltaX	\DeltaY	Direction

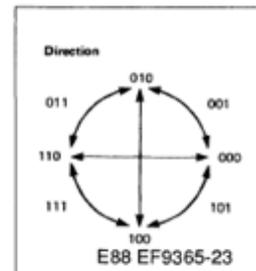
Dimension

ΔX or ΔY	Vector Length
0 0	0 Step
0 1	1 Step
1 0	2 Steps
1 1	3 Steps

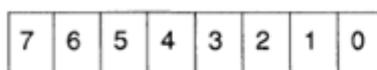
Special Direction Vectors

(for b2, b1, b0 see small vector definition)

1 0 0 0 8	8	Light-pen initialization (WHITE forced low)	(8	H	X	h	x
1 0 0 1 9	9	Light-Pen initialization)	9	I	Y	i	y
1 0 1 0 A	A	5 x 8 Block Drawing (size according to CSIZE)	*	:	J	Z	j	z
1 0 1 1 B	B	4 x 4 Block Drawing (size according to CSIZE)	+	;	K	[k	{
1 1 0 0 C	C	Screen Scanning : Pen or Eraser as defined by CTRL1	,	<	L	\	l	
1 1 0 1 D	D	X Register Reset to 0	-	=	M]	m	}
1 1 1 0 E	E	Y Register Reset to 0	.	>	N	↑	n	~
1 1 1 1 F	F	Direct Image Memory access request for the next free cycle.	/	?	O	←	o	⊗



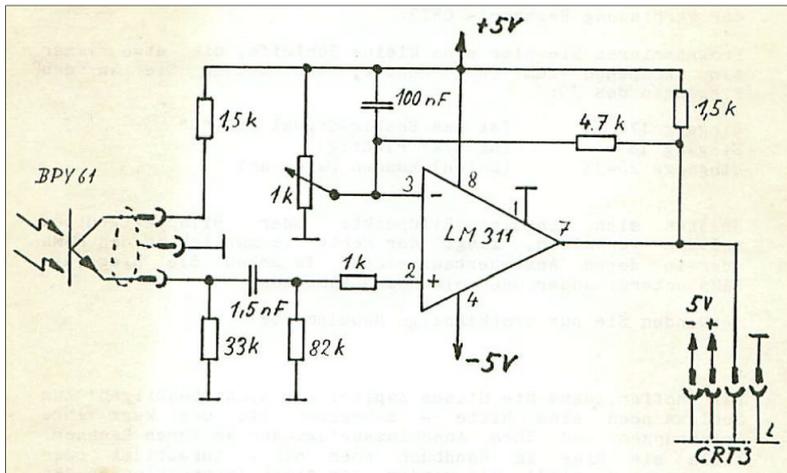
Statusregister:



- HIGH = light-pen sequences ended
 - HIGH = vertical blanking (idem on pin VB)
 - HIGH = ready for a new command ; LOW = busy
 - HIGH = pen out of display window (logical OR of the 6 MSBs of the X and Y registers)
 - HIGH = light-pen sequence ended IRQ (if enabled)
 - HIGH = vertical blanking IRQ (if enabled)
 - HIGH = ready for a new command IRQ (if enabled)
 - IRQ : logical OR of bits 4, 5, 6 ; HIGH when IRQ output is low.
- } These 3 bits are not latched and not masked
 } These 3 bits are reset after a read cycle of the status register at

3 Die Schaltung für den Light Pen

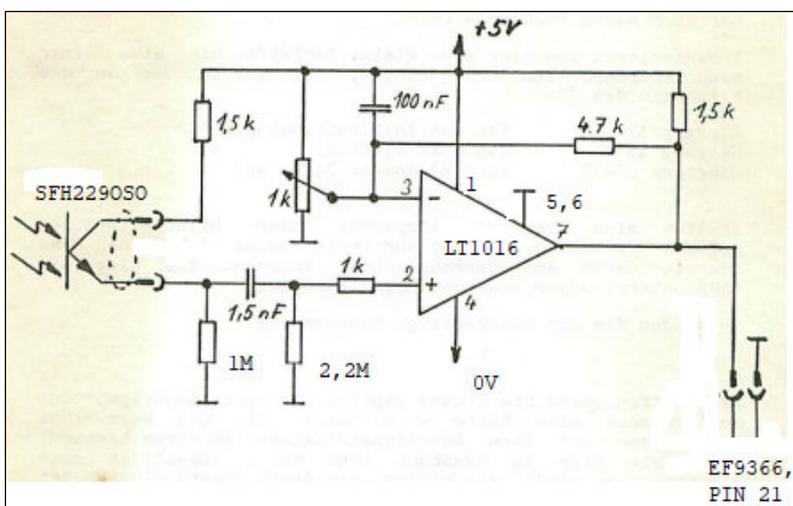
3.1 Die alte Schaltung von Graf Elektronik Systeme aus 1982 in der Beschreibung der CRT3



3.2 Die neue Schaltung

In der neuen Schaltung wurden die aktiven Bauelemente durch wesentlich schnellere ersetzt. Dazu mussten auch Widerstandswerte angepasst werden.

- Reaktionszeit Fototransistor:
 - o BPY61 = 5000 ns
 - o **SFH229 = 10 ns**
- Schaltzeit Komparator:
 - o LM311 = 200 ns
 - o **LT1016 = 10 ns**



NDR-Klein-Computer – Light Pen für GDP64 /GDP64HS

Folgende aktiven Komponenten werden einsetzt:

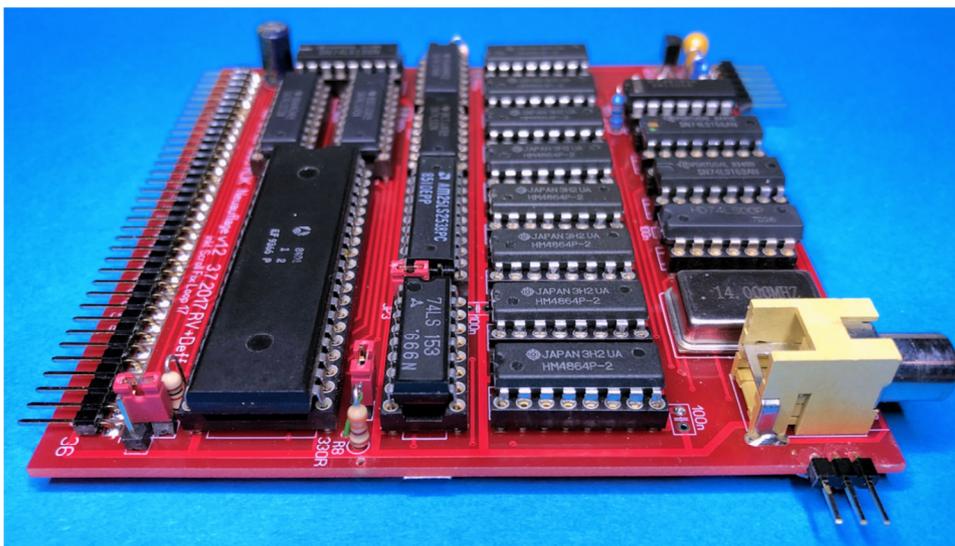
Funktion	Baustein	Beschreibung	Datenblatt
Fototransistor	SFH229	Aufnehmer des Elektronenstrahls der Bildröhre	SFH229.pdf
Komparator	LT1016	Einstellung der Empfindlichkeit des Aufnehmers und Signalverstärkung	Lt1016.pdf

Die Schaltung samt Fixierung des Fototransistors passt sehr gut in das Gehäuse eines Textmarkers!



3.3 Anschluss des Light Pens an die GDP64

Der Light Pen muss einerseits mit +5 Volt und 0 Volt versorgt werden und gibt andererseits das LPCK-Signal ab. Dazu habe ich neben der BAS-Ausgangsbuchse drei Löcher gebohrt und eine abgewinkelte Steckerleiste mit 3 Pins mit Sekundenkleber eingeklebt.



NDR-Klein-Computer – Light Pen für GDP64 /GDP64HS

Die Verdrahtung auf der Rückseite der Platine ist einfach:



4 Stückliste

Lochrasterplatine	1 Stück
Gewinkelte Stiftleiste – 3 polig (Anschluss auf GDP-Platine)	1 Stück
IC-Fassung – 4-polig	1 Stück

Widerstände:

1,0 K Ω	1 Stück
1,0 K Ω Trimmer	1 Stück
1,5 K Ω	2 Stück
4,7 K Ω	1 Stück
1,0 M Ω	1 Stück
2,2 M Ω	1 Stück

Kondensatoren:

1,5 nF	1 Stück
100 nF	1 Stück

Transistoren:

SFH229	1 Stück
--------	---------

ICs:

LT 1016	1 Stück
---------	---------

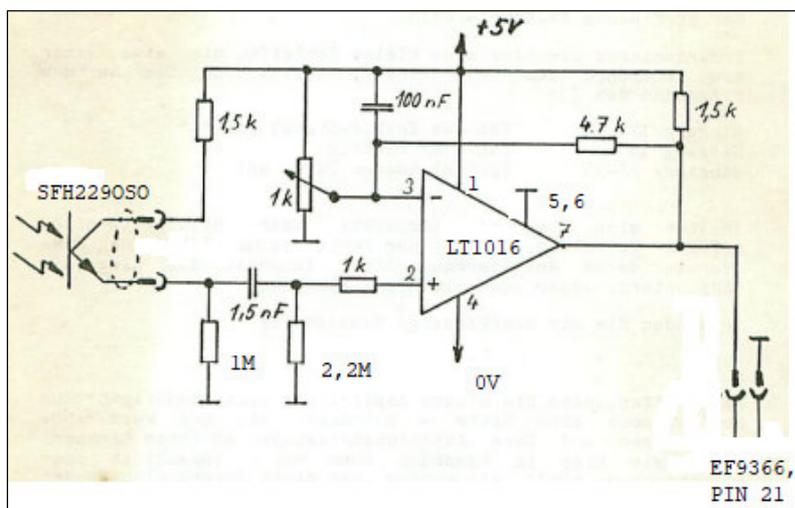
5 Test des Light Pens

5.1 Anzeige der X- / Y-Komponente mit einem Programm

Um den Light Pen testen zu können, braucht man zunächst ein Programm, das die die Werte des X- und des Y-Registers des EF9366 für den Light Pen ausgibt. Das Programm steuert den Bildschirm komplett hell, denn sonst könnte der Light Pen keine Position erkennen.

Dann berührt man mit dem Light Pen verschiedene Positionen auf der Bildröhre und prüft die Anzeigen für die X- und die Y-Komponente auf Plausibilität.

Mit dem 1 Kiloohm-Trimmer wird in der Schaltung des Light Pens die Empfindlichkeit eingestellt. Bei mir konnte ich die Empfindlichkeit so hoch einstellen, dass der Light Pen schon reagiert, bevor die Spitze auf der Bildröhre aufsetzt.



NDR-Klein-Computer – Light Pen für GDP64 /GDP64HS

```

*****
*                               NKC - Light Pen - Test by sEn                               *
*****
*
*   1) Anschluss an PIN 21 LPCK des EF9366                                           *
*   2) Verwendeter Fototransistor ist SFH 229 OSO (10 ns Reaktionszeit)             *
*   3) Verwendeter Komparator ist LT 1016 (10 ns Schaltzeit)                       *
*
*****
*
*   Version 0.1 - Test der Schaltung und Response in XLP- und YLP-Register          *
*   Version 1.0 - Dezimale Anzeige von X- und Y-Position des Light Pens            *
*
*****

;* Programmstart

start:
    bra lipe
    rts

;* Konstanten

    CPU      EQU 1                               ; (68008)

    GDPBASE  EQU $FFFFFF70+CPU                   ;GDP64 Basisadresse / Status / CMD
    GDPXLP   EQU GDPBASE+$0C                     ;X Light Pen - 6 bit
                                                    ;left justified, bit 0 = Status
    GDPYLP   EQU GDPBASE+$0D                     ;Y Light Pen - 8 bit

    COMSCRW  EQU $C                               ;Bildschirm hell steuern
    COMLPIW  EQU $08                             ;Init. Light Pen m. weissem Bildsch.
    COMLPI   EQU $09                             ;Initialisierung Light Pen - nur

;* Variablen

    buffer:  ds.b 10
    lbuff:   dc.b ' ', 0 ; SPACES

```

NDR-Klein-Computer – Light Pen für GDP64 /GDP64HS

```

;* Info zum Programm

;* Zu den Koordinatenwerten aus dem XLP- und dem YLP-Register:

;* Das YLP-Register zeigt in seinen 8 Bit die Werte von Zeile 0 bis 255 an.
;* Diesen Wert kann man direkt für MOVETO oder DRAWTO verwenden.
;* Das XLP-Register zeigt in den linken 6 Bit einen Wert von 0-63.
;* Dies entspricht 64 moeglichen Speicherzyklen mit jeweils 8 Bit.
;* Der GDP waehlt 1 Punkt immer mit einem Speicherzyklus und innerhalb der 8 Bit
;* mit den Signalen MSL0, MSL1 und MSL2 aus.
;* Bit 1 ist fix '0' gesetzt. Bit 0 wird als Light Pen Zyklus Status genutzt.
;* Laesst man die 6 Bit der Positionsinformation da stehe, wo sie sind, koennen
;* sie Werte von 1-253 annehmen.
;* Beruecksichtigt man das Status-Bit 0 nicht, erhaelt man durch ein links
;* Rotieren um 1 Werte von 2-506, die jetzt fuer MOVETO und DRAWTO eingesetzt
;* werden koennen.
;* Hier muss jetzt noch die Laufzeit von Light Pen-Signal bis zur Bildschirmaus-
;* gabe korrigiert werden. Damit wird auch um den Wert des Status-Bits korri-
;* giert.
;* Fuer die Y-Komponente ist keine Korrektur notwendig.

;* Achtung: Bei meinem Aufbau habe ich festgestellt, dass am rechten Rand nach
;* dem Maximalwert '11111100' fuer XLP wieder ein kleinerer Wert ent-
;* steht. Man muss also die hell gesteuerte Bildschirmflaeche am
;* rechten Rand beschneiden.

;* Programm

;* Vorbereitung fuer den Light Pen:
;* Der Schreibstift wird gesenkt und der Bildschirm hell gesteuert.
;* Der Schreibstift wird nun gehoben, da die helle Fläche für die Funktion des
;* Light Pen benoetigt wird und man darauf nur loeschend zeichnen kann.

lipe:
moveq #!setpen, d7                ;Schreibstift auf "Schreiben"
trap #1
move.b #comscrw, d0              ;Bildschirm hell steuern
moveq #!cmd, d7
trap #1
moveq #!erapen, d7              ;Schreibstift auf "Löschen"
trap #1
moveq #40, d1                    ;Dunkle Zone fuer Koordinatenanzeige
move #190, d2
move #230, d3

```

NDR-Klein-Computer – Light Pen für GDP64 /GDP64HS

```

moveq #50, d4
moveq #6, d0
moveq #!Grafik, d7
trap #1

;* Position des Light Pens ermitteln.

initpos:
clr d3                ;X/Y-Position für Ausgabe initialisieren
clr d4
move.b #comlpi, d0    ;Initialisierung Light Pen
moveq #!cmd, d7
trap #1
move.b gdpxlp, d3    ;X-Position Light Pen und Status
btst.b #0, d3        ;Light Pen Zyklus zu Ende?
beq.s initpos        ;Wenn nicht, erneut ermitteln ...
subq #1, d3
rol.w #1, d3         ;X-Komponente auf 0-511 umrechnen
sub #56, d3          ;Laufzeitkorrektur zwischen Stift
                        ;und Anzeige (bei mir)
move.b gdpyp, d4     ;Y-Position Light Pen

;* Koordinatenanzeige X/Y zum Test des Light Pen

lea lbuff, A0        ;Anzeige loeschen
moveq #$22, d0
moveq #50, d1
moveq # 100, d2
moveq #!write, d7
trap #1
move d3, d0          ;Dezimalausgabe X-Position
lea buffer, A0
move.b #'X', (A0)+
move.b #':', (A0)+
move.b # ' ', (A0)+
moveq #!print4d, d7
trap #1
lea buffer, A0
move #$22, d0
move #50, d1
move #100, d2
moveq #!write, d7
trap #1
move.b d4, d0        ;Dezimalausgabe Y-Position
lea buffer, A0

```

NDR-Klein-Computer – Light Pen für GDP64 /GDP64HS

```

move.b #'Y', (A0)+
move.b #':', (A0)+
move.b # ' ', (A0)+
moveq #!print4d, d7
trap #1
lea buffer, A0
move #$22, d0
move #180, d1
move #100, d2
moveq #!write, d7
trap #1

;* Tastaturabfrage

tast:
moveq #!csts, d7
trap #1
beq initpos
moveq #!ci, d7
trap #1
cmp.b #'e', d0                ;Programmende
bne initpos

rts

end.
```

6 Anwendung des Light Pens

6.1 Zeichnen mit dem Light Pen

Mit dem folgenden Programm kann man auf dem Bildschirm zeichnen. Es gibt folgende Funktionen durch Tastendruck:

- c = Löschen des Bildschirms (komplett hell steuern)
- n = Neue Position setzen (ohne zu zeichnen)
- e = Programmende



NDR-Klein-Computer – Light Pen für GDP64 /GDP64HS

```
*****
*                               NKC - Light Pen - Malprogramm by sEn                               *
*****
*
*   1) Anschluss an PIN 21 LPCK des EF9366
*   2) Verwendeter Fototransistor ist SFH 229 OSO (10 ns Schaltzeit)
*   3) Verwendeter Komparator ist LT 1016 (10 ns Schaltzeit)
*
*****
*
*   Version 0.1 - Test der Schaltung und Response in XLP- und YLP-Register
*   Version 1.0 - Malen auf dem Bildschirm, Positionieren und Löschen
*
*****
```

```
;* Programmstart
```

```
start:
```

```
bra lipe
rts
```

```
;* Konstanten
```

```

CPU      EQU 1                ; (68008)

GDPBASE  EQU $FFFFFF70*CPU    ; GDP64 Basisadresse / Status / CMD
GDPXLP   EQU GDPBASE+$0C     ; X Light Pen - 6 bit
                          ; left justified, bit 0 = Status
GDPYLP   EQU GDPBASE+$0D     ; Y Light Pen - 8 bit

COMSCRW  EQU $C              ; Bildschirm hell steuern
COMLPIW  EQU $08            ; Init. Light Pen m. weissem Bildsch.
COMLPI   EQU $09            ; Initialisierung Light Pen - nur
```

```
;* Programm
```

```
;* Vorbereitung fuer den Light Pen:
;* Der Schreibstift wird gesenkt und der Bildschirm hell gesteuert.
;* Der Schreibstift wird nun gehoben, da die helle Fläche für die Funktion des
;* Light Pen benoetigt wird und man darauf nur loeschend zeichnen kann.
```

NDR-Klein-Computer – Light Pen für GDP64 /GDP64HS

```

lipe:
    moveq #!setpen, d7                ;Schreibstift auf "Schreiben"
    trap #1
    move.b #comscrw, d0              ;Bildschirm hell steuern
    moveq #!cmd, d7
    trap #1
    moveq #!erapen, d7              ;Schreibstift auf "löschen"
    trap #1

;* Erste Position des Light Pens ermitteln, von der aus dann gezeichnet wird.

initpos:
    clr d1                          ;Initialisierung X- und Y-Komponente
    clr d2                          ;zum Zeichnen
    move.b #comlpi, d0              ;Initialisierung Light Pen
    moveq #!cmd, d7
    trap #1
    move.b gdpxlp, d1               ;X-Position Light Pen und Status
    btst.b #0, d1                   ;Light Pen Zyklus zu Ende?
    beq.s initpos                   ;Wenn nein, dann nochmal pruefen
    rol.w #1, d1                    ;X-Komponente auf 0-511 umrechnen
    sub #56, d1                     ;Laufzeitkorrektur zwischen Stift
                                    ;und Anzeige (bei mir)
    move.b gdpylp, d2               ;Y-Position Light Pen
    moveq #!moveto, d7              ;Position Schreibstift neu setzen
    trap #1

;* Ab hier malen ...

schreiben:
    clr d1                          ;Initialisierung X- und Y-Komponente
    clr d2                          ;zum Zeichnen
    move.b #comlpi, d0              ;Initialisierung Light Pen
    moveq #!cmd, d7
    trap #1
    move.b gdpxlp, d1               ;X-Position Light Pen und Status
    btst.b #0, d1                   ;Light Pen Zyklus zu Ende?
    beq.s tast                      ;Nein, dann Tastaturabfrage
    rol.w #1, d1                    ;X-Komponente auf 0-511 umrechnen
    sub #56, d1                     ;Laufzeitkorrektur zwischen Stift
                                    ;und Anzeige (bei mir)
    move.b gdpylp, d2               ;Y-Position Light Pen

```

NDR-Klein-Computer – Light Pen für GDP64 /GDP64HS

```

moveq #!drawto, d7          ;Dunkle Linie malen
trap #1
bra.s schreiben

;* Tastaturabfrage

tast:
moveq #!csts, d7           ;Wenn keine Taste gedrueckt, weiter mit
trap #1                   ;'schreiben'
beq.s schreiben
moveq #!ci, d7            ;Wenn Taste gedrueckt, dann ...
trap #1
cmp.b #'n', d0
beq initpos               ;Neue Position ohne Schreiben
cmp.b #'c', d0            ;Bildschirm löschen
beq lipe
cmp.b #'e', d0           ;Programmende
bne.s schreiben

rts

end.
```

6.2 Programm für Auslösen eines Controls

Mit dem folgenden Programm kann man Controls / Buttons erzeugen, in eine Tabelle eintragen und dann mit dem Light Pen diese auslösen und entsprechende Aktionen umsetzen. Es gibt folgende Funktionen:

- Berührung Control 1 zeichnet einen Kreis
- Berührung Control 2 löscht den Kreis
- Tastendruck ,e' beendet das Programm



NDR-Klein-Computer – Light Pen für GDP64 /GDP64HS



```

*****
*
*                               NKC - Light Pen Control by sEn
*
*****
*
*
*   1) Anschluss an PIN 21 LPCK des EF9366
*   2) Verwendeter Fototransistor ist SFH 229 OSO (10 ns Reaktionszeit
*   3) Verwendeter Komparator ist LT 1016 (10 ns Schaltzeit)
*
*****
*
*   Version 0.1 - Test der Schaltung und Response in XLP- und YLP-Register
*   Version 1.0 - Anlegen eines Controls, Ausloesen eines Controls
*
*****

; Programmstart

start:
  bra lipe
  rts
  
```

NDR-Klein-Computer – Light Pen für GDP64 /GDP64HS

;* Konstanten

```

CPU      EQU 1                                ; (68008)

GDPBASE  EQU $FFFFFF70*CPU                    ; GDP64 Basisadresse / Status / CMD
GDPXLP   EQU GDPBASE+$0C                      ; X Light Pen - 6 bit
                                                ; left justified, bit 0 = Status
GDPYLP   EQU GDPBASE+$0D                      ; Y Light Pen - 8 bit

COMSCRW  EQU $C                                ; Bildschirm hell steuern
COMLPIW  EQU $08                              ; Init. Light Pen m. weissem Bildsch.
COMLPI   EQU $09                              ; Initialisierung Light Pen - nur

```

;* Variablen

;Control-Beschriftungen:

```

c1:      dc.b 'con 1', 0
c2:      dc.b 'con 2', 0

```

;Control Tabelle:

```

contab:  df.w 100, $0000

```

;* Programm

lipe:

;* Button / Click Control 1 für NKC Light Pen anlegen

```

move #1, d6                                ; Nummer des Controls im Programm

```

;Control malen

```

moveq #6, d0                                ; Ausgefülltes Rechteck zeichnen
moveq #100, d1                               ; X-Position
moveq #100, d2                               ; Y-Position
moveq #50, d3                                ; Positive Kantenlaenge X
moveq #25, d4                                ; Positive Kantenlaenge Y
moveq #!Grafik, d7                           ; Control als Rechteck malen
trap #1

```

;Control in Tabelle eintragen

NDR-Klein-Computer – Light Pen für GDP64 /GDP64HS

```

;Aufbau: W1 = X0, W2 = Y0, W3 = X1, W4 = Y1, W5 = Nr. Control
lea contab (PC), A1          ;Control-Tabelle laden - nur beim 1. Control
move d1, (A1)+              ;X0
move d2, (A1)+              ;Y0
add d1, d3
move d3, (A1)+              ;X1
add d2, d4
move d4, (A1)+              ;Y1
move d6, (A1)+              ;Control Nr.

;Control beschriften
moveq #!erapen, d7          ;GDP auf Loeschen
trap #1
moveq #23, d0                ;Textausgabe
lea c1 (PC), A0
addq #5, d1                  ;Neue X-Position
sub d2, d4                   ;Alte Kantenlaenge Y
ror.w #1, d4                 ;Alte Kantenlaenge Y / 2
add d4, d2                   ;Mitte des Controls
subq #8, d2                  ;Neue Y-Position 4 Zeilen unter der Mitte
moveq #11, d3                ;Schriftgroesse
moveq #!grafik, d7
trap #1
moveq #!setpen, d7           ;GDP auf Schreiben
trap #1

;* Button / Click Control 2 für NKC Light Pen anlegen

move #2, d6                  ;Nummer des Controls im Programm

;Control malen
move #6, d0                  ;Ausgefuehltes Rechteck zeichnen
move #175, d1                ;X-Position
move #100, d2                ;Y-Position
move #50, d3                 ;Positive Kantenlaenge X
move #25, d4                 ;Positive Kantenlaenge Y
moveq #!Grafik, d7           ;Control als Rechteck malen
trap #1

;Control in Tabelle eintragen
;Aufbau: W1 = X0, W2 = Y0, W3 = X1, W4 = Y1, W5 = Nr. Control
move d1, (A1)+              ;X0

```

NDR-Klein-Computer – Light Pen für GDP64 /GDP64HS

```

move d2, (A1)+           ;Y0
add d1, d3
move d3, (A1)+           ;X1
add d2, d4
move d4, (A1)+           ;Y1
move d6, (A1)+           ;Control Nr.

;Control beschriften
moveq #!erapen, d7       ;GDP auf Loeschen
trap #1
moveq #23, d0            ;Textausgabe
lea c2 (PC), A0
addq #5, d1              ;Neue X-Position
sub d2, d4               ;Alte Kantenlaenge Y
ror.w #1, d4             ;Alte Kantenlaenge Y / 2
add d4, d2               ;Mitte des Controls
subq #8, d2              ;Neue Y-Position 4 Zeilen unter der Mitte
moveq #11, d3           ;Schriftgroesse
moveq #!grafik, d7
trap #1
moveq #!setpen, d7      ;GDP auf Schreiben
trap #1

;* Position des Light Pens ermitteln.

lppos:
clr d1                  ;X/Y Position für Ausgabe initialisieren
clr d2
move.b #comlpi, d0      ;Initialisierung Light Pen
moveq #!cmd, d7
trap #1
move.b gdpxlp, d1      ;X-Position Light Pen und Status
btst.b #0, d1           ;Light Pen Zyklus zu Ende?
beq.s tast            ;Wenn nicht, erneut ermitteln ...
subq #1, d1            ;und Tastatur abfragen
rol.w #1, d1           ;X-Komponente auf 0-511 umrechnen
sub #56, d1            ;Laufzeitkorrektur zwischen Stift
                        ;und Anzeige (bei mir)
move.b gdpylp, d2      ;Y-Position Light Pen
rol.w #1, d2           ;Y-Komponente auf 0-511 umrechnen,
                        ;weil Grafik-Befehl bei Y so arbeitet!

```

NDR-Klein-Computer – Light Pen für GDP64 /GDP64HS

```

;* Abfrage der X/Y-Position des Light Pens und Zuordnung des Controls
;* Liefert die Control Nr. in d0.w oder aber $0000, wenn nicht gefunden.
;* Uebergeben werden XLP als d1.w und YLP als d2.w

chkcon:
    clr d0                ;Ausgaberegister initialisieren
    lea contab, A0       ;Control-Tabelle laden

scon:
    move (A0)+, d3        ;X0
    move (A0)+, d4        ;Y0
    move (A0)+, d5        ;X1
    move (A0)+, d6        ;Y1
    move (A0)+, d0        ;Control Nr.

    cmp #$0000, d0       ;Tabellenende - k e i n  valides Control
    beq.s endcon

    cmp d3, d1            ;Check XLP gegen X0
    ble.s scon           ;Wenn XLP kleiner als X0, dann naechstes Contr
    cmp d5, d1            ;Check XLP gegen X1
    bgt.s scon           ;Wenn XLP groesser als X1, dann naechstes Cont
    cmp d4, d2            ;Check YLP gegen Y0
    ble.s scon           ;Wenn YLP kleiner als Y0, dann naechstes Contr
    cmp d6, d2            ;Check YLP gegen X1
    bgt.s scon           ;Wenn YLP groesser als X1, dann naechstes Cont
endcon:                  ;Passendes Control gefunden oder Tabellenende!

;* Auswertung der Control Nr. in d0.w

    cmp #1, d0            ;Control 1
    beq.s conact1
    cmp #2, d0            ;Control 2
    beq.s conact2
    bra lppos             ;k e i n  bekanntes Control
conact1:                 ;Aktion Control 1
    moveq #!setpen, d7
    trap #1
    moveq #7, d0           ;Kreis zeichnen
    move #255, d1
    move #300, d2
    move #75, d3

```

NDR-Klein-Computer – Light Pen für GDP64 /GDP64HS

```

moveq#!Grafik, d7
trap #1
bra lppos
conact2:                                ;Aktion Control 2
moveq #!erapen, d7
trap #1
moveq #7, d0                            ;Kreis loeschen
move #255, d1
move #300, d2
move #75, d3
moveq#!Grafik, d7
trap #1
bra lppos

rts

;* Tastaturabfrage

tast:
moveq #!csts, d7
trap #1
beq lppos
moveq #!ci, d7
trap #1
cmp.b #'e', d0                          ;Programmende
bne lppos
rts

end.
```

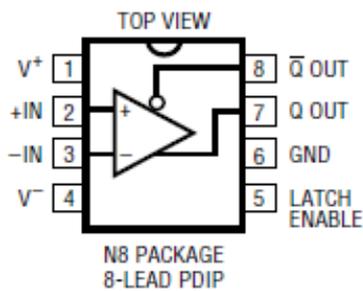
7 Anhang

7.1 Bausteine

7.1.1 Fototransistor SFH229



7.1.2 Komparator LT1016



7.2 Quellennachweis:

Funktion	Baustein	Datenblatt
Fototransistor	SFH229	SFH229.pdf
Komparator	LT1016	Lt1016.pdf
Grafikprozessor	EF9366	Graphic_Processors_Data_Book_Mar89.pdf
Grafikkarte CRT3 – Schaltungsvorschlag Light Pen	CRT3	CRT3_BAU.pdf