

# SPRINTER 320

: 2006 3 20

:

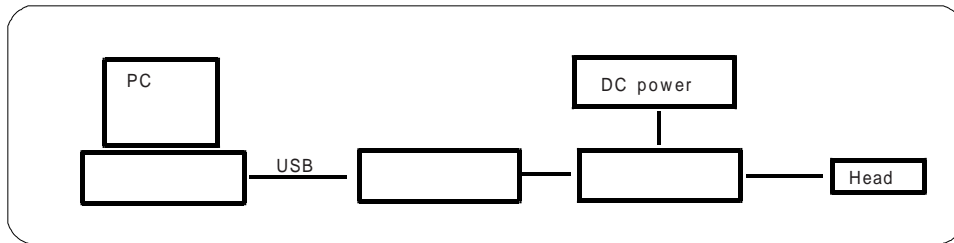
- 1.
- 2.
- 3.
- 4.
- 5.
- 6.

1.

1-1.

SPR NTER320

96 ( 128 )

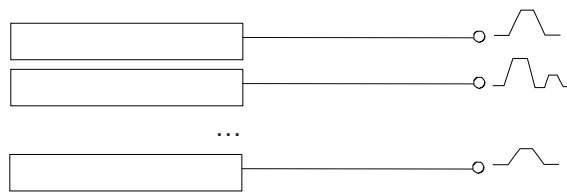


1. SPR NTER 320

1-2.

trigger

가



2.

1-3.

onoff

2.

2-1.

\* : 96 128  
\*

- stirring 4 가 . 가 .
- 1024 points , 0.1 us 102.4 us
- 125 ns jitter
- 12 bits DAC
- ±100V ( 가 )
- 1000 pF 60V/us

2-2.

\*  
= +24V  
= 1.2 A (96 ), 1.6 A (96 )  
1.6 A (128 ), 1.9 A (128 )  
0.5 A .

\* (+) VPP  
= 25V, 135V, 15V .  
= 0.26 A (96 )

VNN  
\* (-) VNN  
= -25V, -135V, 15V .  
= 0.26 A (96 )

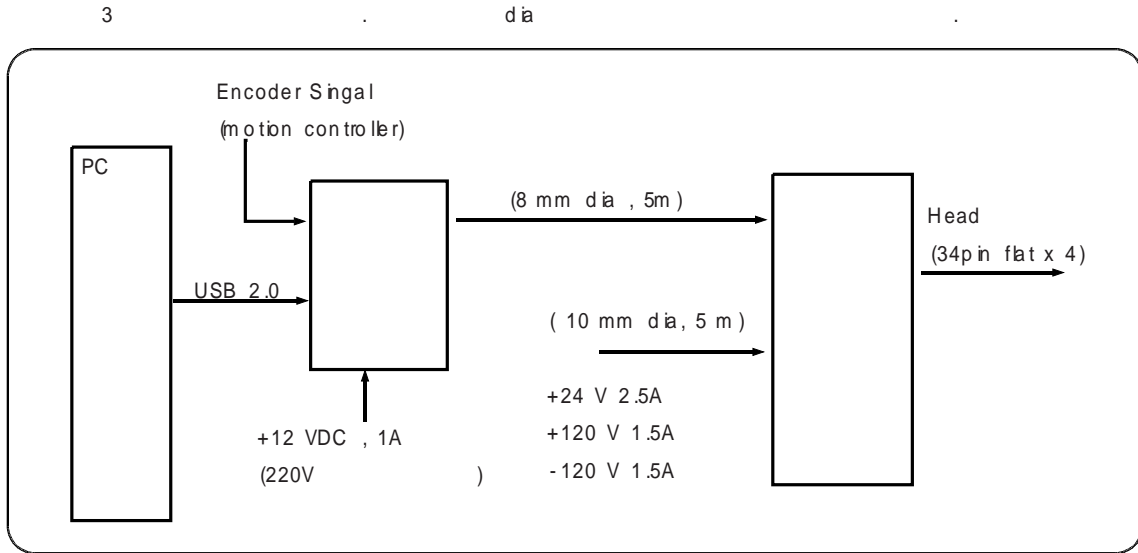
VPP  
)  
96  
1000 pF, -100V , rise/fall time = 1.5 us, duration = 4 us, VPP=+25V, VNN=-125V

	32 ( 가 )	64 ( 가 )
1 kHz	VPP=0.26A (+0), VNN=0.27A (+0.01)	VPP=0.26A (+0), VNN=0.27A (+0.01)
10 kHz	VPP=0.30A (+0.04), VNN=0.33A (+0.07)	VPP=0.33A (+0.07), VNN=0.37A (+0.11)
20 kHz	VPP=0.33A (+0.07), VNN=0.41A (+0.15)	VPP=0.40A (+0.14), VNN=0.48A (+0.22)

2-3.

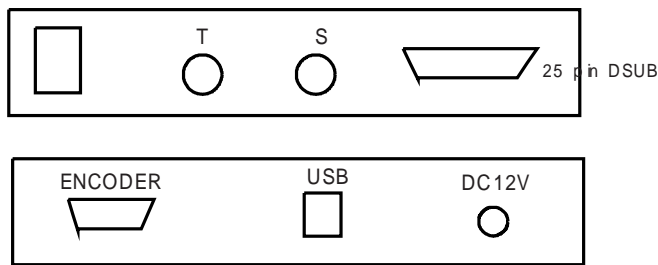
- \* : 5V TTL , 1
- \* : 5 MHz . ( 40MHz ), RS485 A/B ( 5V )
- \* =  $2^{23}$  (8 Mbits) , ( 8 Mbits 가 )
- ) 0.5 um , 가 4 m .
- pixel pixel
- \*
- double buffer , 64 Mbits( $2^{26}$ ) .
- 2 bits (4 ) , 128 , 1 pixel position 256 bits ( $2^8$ ) 가
- pixel position 256 k ( $2^{18}$ ) .
- ) 2 um pixel , 0.5 m .
- \* : 3 Mbytes/sec
- \*
- 12V , 1A

3.



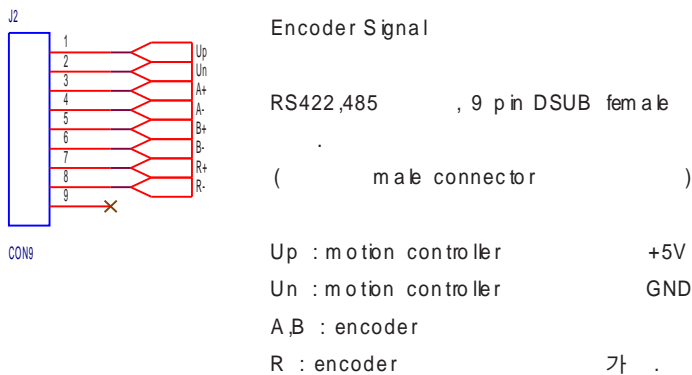
3.

3-1.

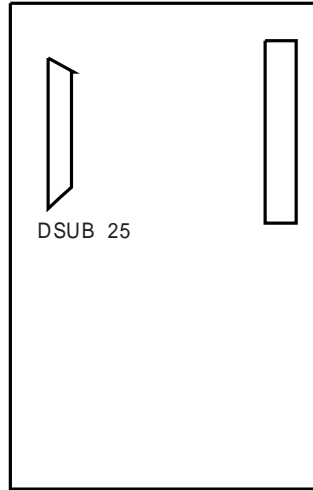
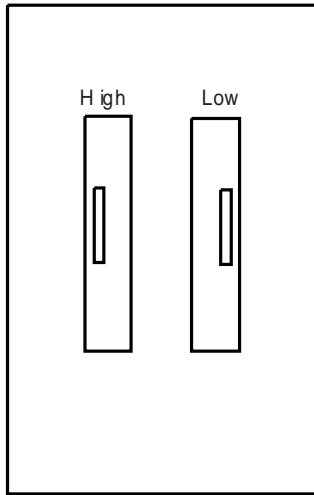


4.

- \* PC : USB 2.0
- \* : T ( ) S ( ) , BNC
- T trigger 5 V TTL
- S trigger 5V TTL
- \* : DSUB 25 , 25 pin cable . (5m )
- \* ENCODER : RS485 +5V
- +5V opto-coupler



3-2.



- 1.VPP
- 2.AGND
- 3.VNN
- 4.AGND
- 5.+24V
- 6.DGND
- 7.CASE

- \* 96 2 50 , Low 1~ 48 .
- \* DSUB 25 .
- \* VPP (+) , VNN (-) , AGND analog ground
- \* +24V DGND .
- \* CASE . ( 가 ...)

4.

HeadDriver class DLL  
Borland C++ Builder 6.0

4-1.

```

class wHeadDriver
{
public:
    wHeadDriver();
    ~wHeadDriver();
    //
    bool inHardware(void);

    //
    bool resetPosition(void); //set origin
    bool setPixelPosition(unsigned long PixelCount,unsigned long *forward,unsigned long *backward=0); // takes 1~2 seconds

    //
    bool setPixelData(unsigned char* data); //sizeof(data)=PixelCount*NumberOfNozzle,value=0,1,2,3
                                                // index=(N th pixel) - 1)*NumberOfNozzle + (N th Nozzle - 1)
    void enablePrint(void);
    void disablePrint(void);

    //
    bool setWaveformLength(int length); //          1024,512,256,128 points
    void setFG(int NozzleIndex,int WaveIndex,int NumberOfControlPoints,double *ControlPointsCoordinate); //

    // Spitting
    void setFireWave(int index); // index = 0, 1, 2, 3, spitting
    bool fireNozzleSelect(int nz,bool onoff);
    bool fireDrop(double kHz,int count);
    bool isDropEnd(void);
    bool continuousSpitting(bool onoff,double FireFrequency=1.0);

    //debug
    void getErrorMessage(char* msg); //          256
    void getEncoder(long& count);

    // trigger
    void setTriggerLength(double TriggerLength);
    void setTriggerDelay(double TriggerDelay);
    void enableExternalTrigger(bool onoff,bool invert=false);

    // tilting :          , pixel position
    void setControlCounter(int n,int k);
    void setTrigPos(int nz,int k);
    void programTrigPos(void);
};
    
```

4-2.

4-2-1.

initHardware() , false

4-2-1.

spitting , printing waveform

waveform 4 , 2

waveform 0 : off , 0V DC , stirring

waveform 1 : on

waveform 2, 3 : waveform 1

spitting , setFireWave(int index) waveform , off 0 waveform

printing , setPixelData waveform

4-2-2. SPITTING

fireNozzleSelect(int nz, bool onoff)

nz 0-base

onoff bool true

fireDrop(double kHz, int count) , (nonblocking function)

isDropEnd . (true )

continuousSpitting(bool onoff, double FireFrequency=1.0)

4-2-3. PRINTING

A/B 23 bit up/down counter

가 , mask memory

mask memory bit , print mode 가 trigger 가  
pixel , pixel 128 (96 128 가 .) onoff

0~3 waveform index

) 1um , 1000 um , 200 um 3 ,

(1 waveform ), pixel 0,2,4.. ,

0,2,4,... 1 waveform , 1,3,5,... 2 waveform

resetPosition(void); //set origin

setPixelPosition(unsigned long PixelCount, unsigned long \*forward, unsigned long \*backward=0);

PixelCount=5, forward[]={1000,1200,1400,1600,1800}, backward=forward

setPixelData(unsigned char\* data)

data[5\*128];

data[0,1,...,127] = 1;

data[128,129,130,...,255]={1,0,1,0,...,0}

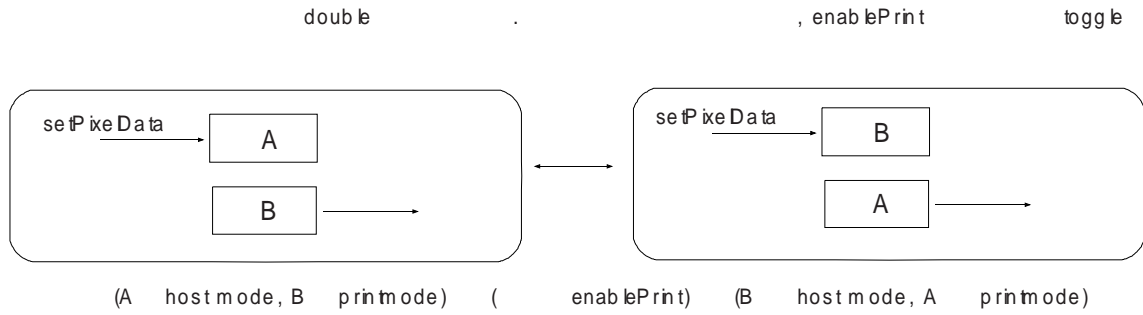
data[256,257,258,...,511]={1,2,1,2,...,2}

...

, enablePrint ,

disablePrint 1 . setPixelData+enablePrint+disablePrint

4-2-4. double buffer



- a. A 가 host mode
- b. setPixelData host mode A 가
- c. enablePrint 가 A 가 print mode가
- d. setPixelData B host mode 가 가
- e. disablePrint 가 , 가 A

setPixelData thread

e. disablePrint 가

spitting fireDrop continuousSpitting

- host mode on off 1 pixel
- enablePrint
- pixel counter 0 , 1 pixel trigger

spitting enablePrint

4-3.

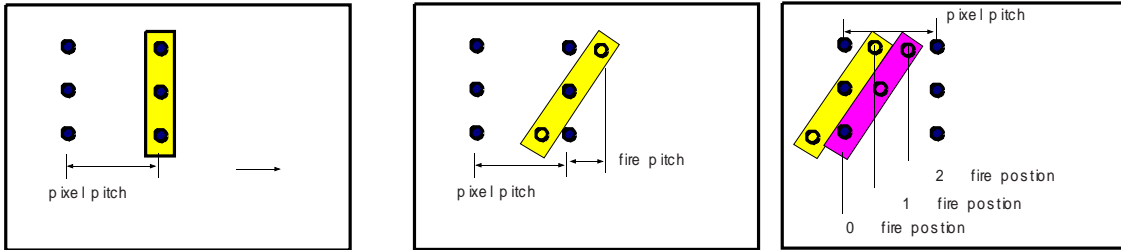
```

*
initHardware      , boolean
*
- resetPosition DPN
- setPixelPosition :
    resetPosition
    ,resetPosition 2000 um      100 um      10
    PixelCount=10,forward[0]=(2000)/(      ),forward[1]=(2000+100)/(      ),...
    forward backward
    backward
*
-
setPixelData(unsigned char*data):data      swath      ,(      )
    (      )x(      )
,
index= PixelCount*(      )+(      )
    waveform index ,0,1,2,3
-
setPixelData swath      ,enablePrint
    waveform
    double buffer
    ,disablePrint
* waveform
- setFG (in NozzleIndex,in WaveIndex,in NumberOfControlPoints,double *ControlPointsCoordinate);
    (NozzleIndex,0 base) 4 (0 base) waveform      waveform
    control point      ,0      0 waveform
0 V      80 V      1 us      rise      2 us      가 1 us      fall
NumberOfControlPoints = 4
ControlPointsCoordinate[]={ 0,0, 1,80,3,80,4,0 }
    x,y      ,x micro-second      ,y voltage
x      0.1 us, y      0.1 volt
-----
* spitting
- fireNozzleSelect
    spitting      index 0 base      , on=true,off=false
- fireDrop + isDropEnd
    fireDrop nonblocking
    isDropEnd
* continuousSpitting
    onoff      , nonblocking
*
,256
*

```

\*

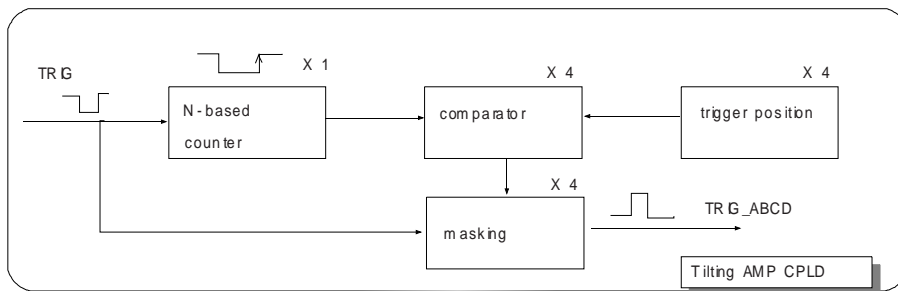
A. , 가 pixel pitch fire . , pixel pitch fire 가  
 fire pitch pixel pitch ,



A. B. C. pixel pitch fire position

B.

pixel pitch , pixel pitch N fire position  
 pixel position N  
 3 N=3 , 0 0 , 1 1 2 2  
 0 , 0 fire position on/off  
 fire position trigger  
 ( trigger ) = ( pixel ) x ( pixel fire position : N)  
 N trigger 가  
 4 fire position



D. tilting

C.

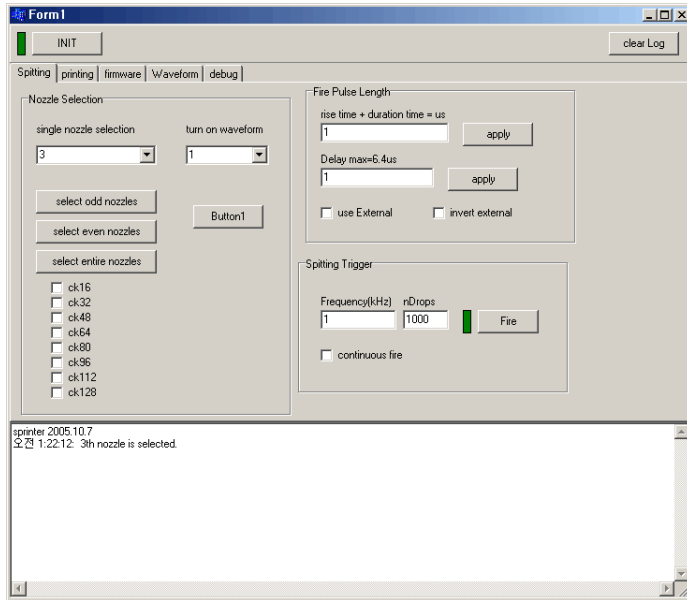
```

C-1. void setControlCounter(int n,int k);
      N k trigger N
      trigger
C-2. void setTrigPos(int nz,int k); void programTrigPos(void);
      (nz) N trigger (k) fire 가
      programTrigPos
C-3. setPixelPosition fire position , setPixelData k=0
      k=1,2,...N-1 k=0 on/off
    
```

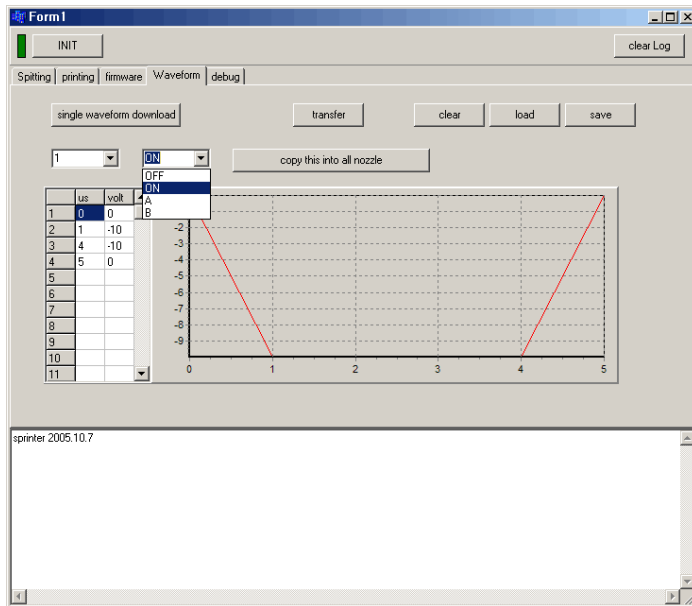
5.

SPRINTER320

가



spitting



4