

# NHD-C160100AZ-RN-GBW

## COG (Chip-On-Glass) Liquid Crystal Display Module

NHD-	Newhaven Display
C160100-	160 x 100 pixels
AZ-	Model
R-	Reflective
N-	No backlight
G-	STN- Gray
B-	6:00 view
W-	Wide Temp (-20°C ~ +70°C)

**RoHS Compliant**

**Newhaven Display International, Inc.**

2511 Technology Drive, Suite 101

Elgin IL, 60124

Ph: 847-844-8795

Fax: 847-844-8796

[www.newhavendisplay.com](http://www.newhavendisplay.com)

[nhtech@newhavendisplay.com](mailto:nhtech@newhavendisplay.com)

[nhsales@newhavendisplay.com](mailto:nhsales@newhavendisplay.com)

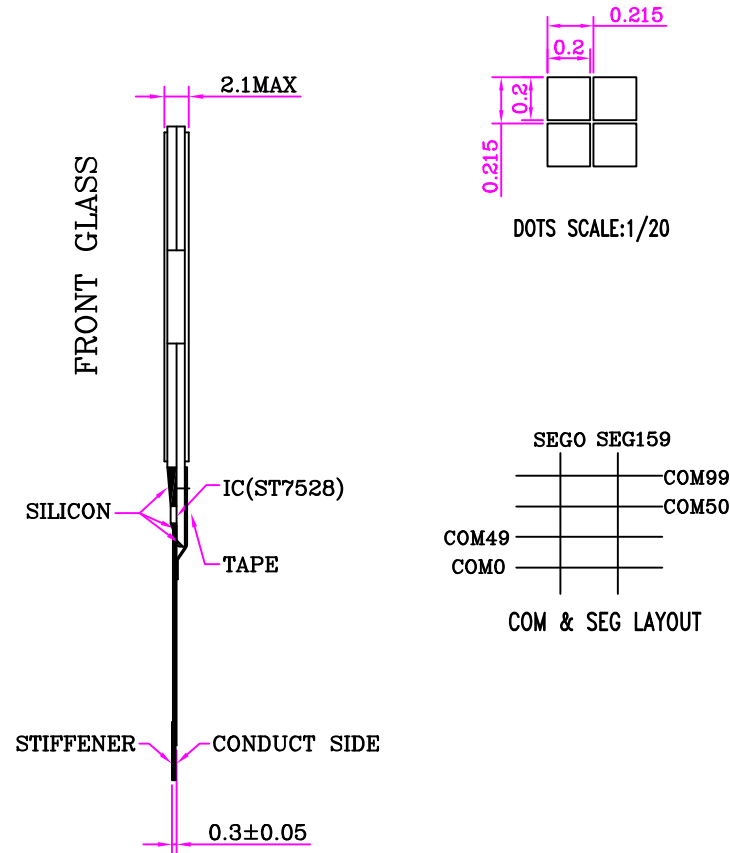
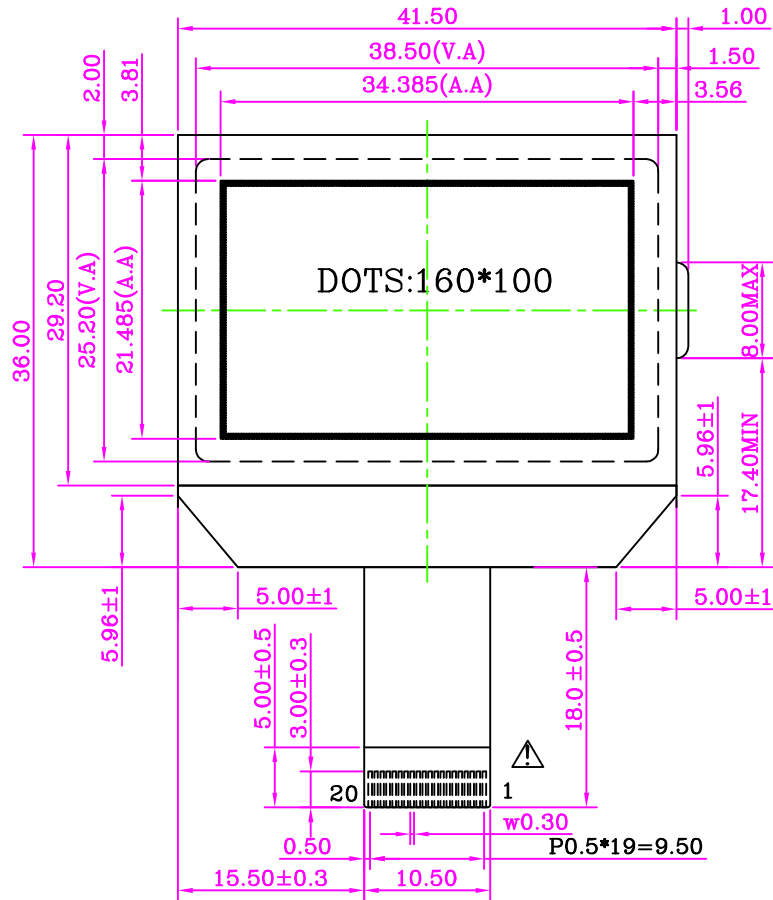
## Document Revision History

Revision	Date	Description	Changed by
0	10/31/2007	Initial Release	-
1	9/14/2009	User guide reformat	BE
2	10/14/2009	Updated Electrical Characteristic	MC
3	12/08/2009	Updated Block Diagram, Pins 4 and 5, and Timing Characteristics	MC

## Functions and Features

- 160 x 100 pixels
- Built-in ST7528 controller
- +3.0V power supply
- 1/100 duty; 1/11 bias
- RoHS Compliant

# Mechanical Drawing



## PIN ASSIGNMENT

1	CSB
2	RST
3	A0
4	/WR
5	/RD
6	DB0
7	DB1
8	DB2
9	DB3
10	DB4
11	DB5
12	DB6
13	DB7
14	VDD
15	VSS
16	VOUT
17	V4
18	V3
19	V2
20	V1

## Specification:

- 1). Driving: Duty:1/100, 1/11 Bias, Vlcd: 11.5V
- 2). Viewing Direction: 6 O'clock
- 3). Display mode: STN/Gray Mode/Positive/Reflective
- 4). Operating temp.: -20°C~+70°C  
Storage temp.: -30°C~+80°C
- 5). Driver : ST7528 Vdd:3.0V
- 6). Unmarked Tolerances: ±0.2mm
- 7). RoHS Compliant

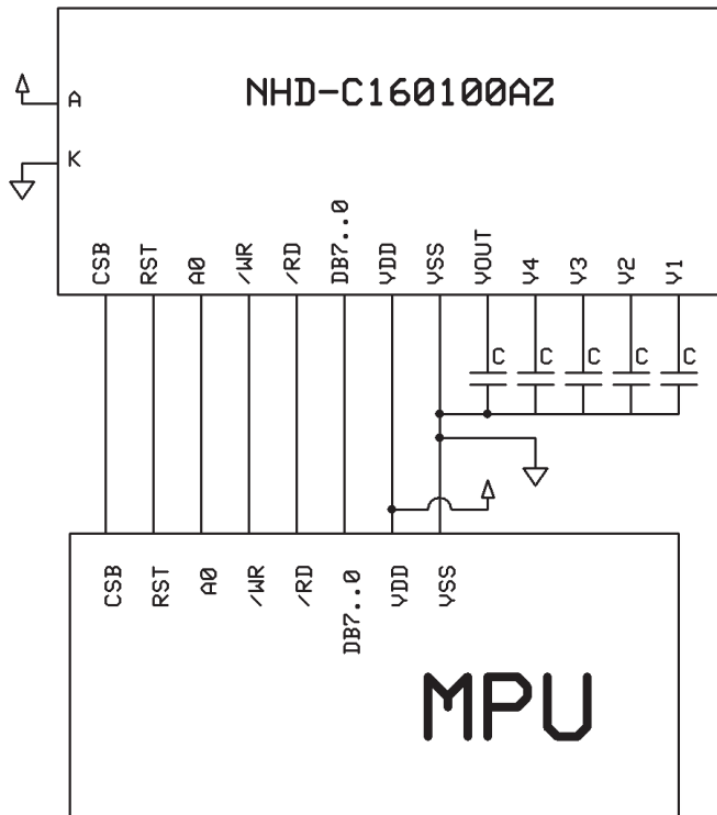
Model Name: NHD-C160100AZ-RN-GBW		<i>Newhaven Display</i>	
GENERAL TOL: ± 0.3			
APPROVALS	DATE	DRAWN NO.	SCALE: 1:1
DWN: YE.GUO.XIANG			
CHK:		SIZE: A4	UNIT: mm
APP:			Page: 1-1

## Pin Description and Wiring Diagram

Pin No.	Symbol	External Connection	Function Description
1	CSB	MPU	Active Low chip select
2	RST	MPU	Active Low Reset signal
3	A0	MPU	Register select signal. A0=1: Data, A0=0: Command
4	/WR	MPU	Active LOW write signal
5	/RD	MPU	Active LOW read signal
6-13	DB0-DB7	MPU	Bi-directional 8-bit data bus.
14	VDD	Power Supply	Power supply for LCD and logic (+3.0V)
15	Vss	Power Supply	Ground
16	VOUT	Power Supply	Connect to 1uF cap to VSS or VDD
17	V4	Power Supply	1.0uF-2.2uF cap to Vss
18	V3	Power Supply	1.0uF-2.2uF cap to Vss
19	V2	Power Supply	1.0uF-2.2uF cap to Vss
20	V1	Power Supply	1.0uF-2.2uF cap to Vss

**Recommended LCD connector:** 0.5mm pitch pins. Molex p/n: 52746-2070

**Backlight connector:** --- **Mates with:** ---



## Electrical Characteristics

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Operating Temperature Range	Top	Absolute Max	-10	-	+60	°C
Storage Temperature Range	Tst	Absolute Max	-20	-	+70	°C
Supply Voltage	VDD		2.8	3.0	3.3	V
Supply Current	IDD	Ta=25°C, VDD=3.0V	-	1.5	2.5	mA
Supply for LCD (contrast)	VDD-V0	Ta=25°C	-	11.5	-	V
"H" Level input	Vih		2.2	-	VDD	V
"L" Level input	Vil		0	-	0.6	V
"H" Level output	Voh		2.4	-	-	V
"L" Level output	Vol		-	-	0.4	V

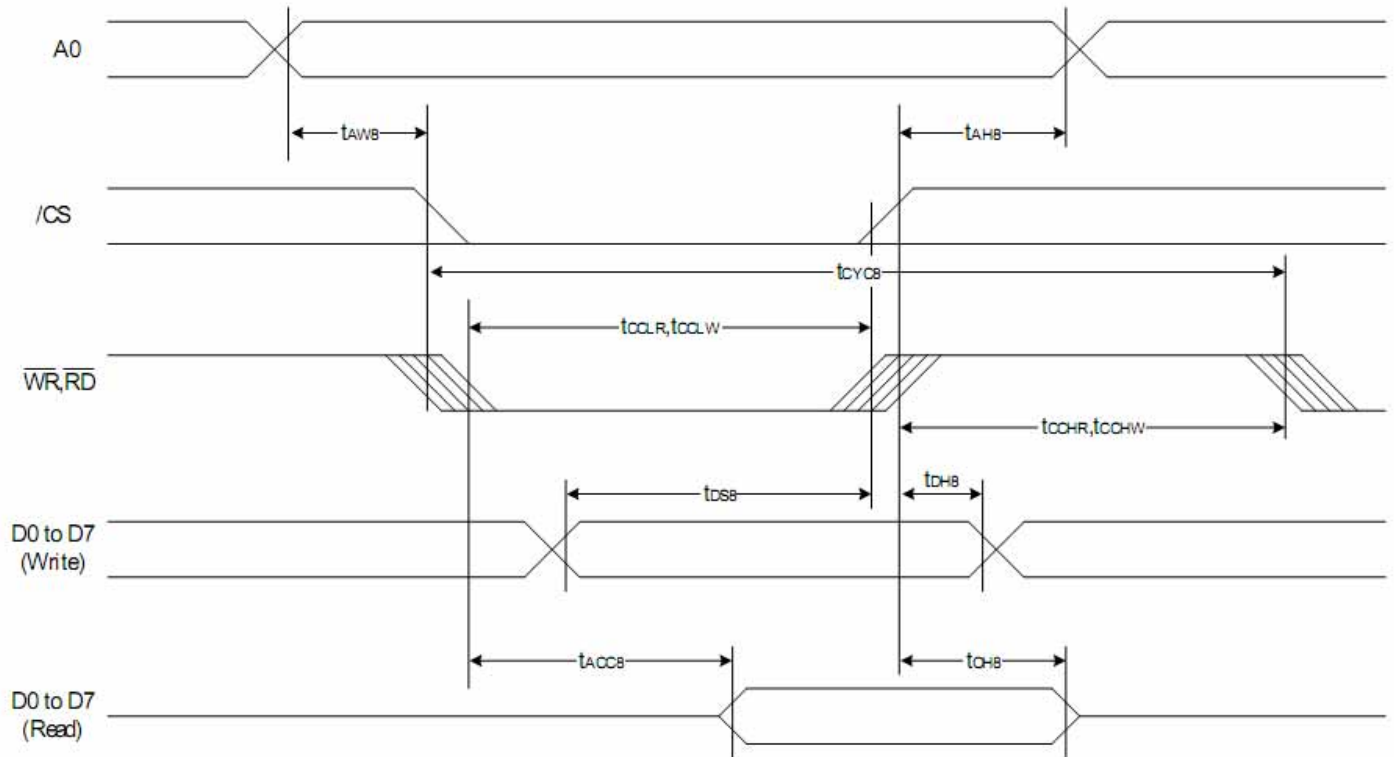
## Optical Characteristics

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Viewing Angle - Vertical		Cr≥2	-60	-	+35	°
Viewing Angle - Horizontal	Φ		-40	-	+40	°
Contrast Ratio	CR		-	6	-	-
Response Time (rise)	Tr	-	-	150	250	ms
Response Time (fall)	Tf	-	-	150	250	ms

## Controller Information

Built-in ST7528. Download specification at [http://www.newhavendisplay.com/app\\_notes/ST7528.pdf](http://www.newhavendisplay.com/app_notes/ST7528.pdf)

# Timing Characteristics



(VDD = 3.3V , Ta =25°C)

Item	Signal	Symbol	Condition	Rating		Units
				Min.	Max.	
Address hold time	A0	tAH8		0	—	ns
Address setup time		tAW8		0	—	
System cycle time		tCYC8		240	—	
Enable L pulse width (WRITE)	WR	tCCLW		80	—	
Enable H pulse width (WRITE)		tCCHR		80	—	
Enable L pulse width (READ)	RD	tCCLR		140	—	
Enable H pulse width (READ)		tCCHW		80	—	
WRITE Data setup time	D0 to D7	tDS8		40	—	
WRITE Data hold time		tDH8		10	—	
READ access time		tACC8	CL = 100 pF	—	70	
READ Output disable time		tOH8	CL = 100 pF	5	50	

## Table of Commands

Instruction	A0	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Description
<b>EXT=0 or 1</b>											
Mode Set	0	0	0	0	1	1	1	0	0	0	2-byte instruction to set Mode and FR( Frame frequency control) BE( Booster efficiency control)
	0	0	FR3	FR2	FR1	FR0	0	BE	x'	EXT	
<b>EXT=0</b>											
Read display data	1	1	Read data							Read data into DDRAM	
Write display data	1	0	Write data							Write data into DDRAM	
Read status	0	1	BUSY	ON	RES	MF2	MF1	MF0	DS1	DS0	Read the internal status
ICON control register ON/OFF	0	0	1	0	1	0	0	0	1	ICON	ICON=0: ICON disable(default) ICON=1: ICON enable & set the page address to 16
Set page address	0	0	1	0	1	1	P3	P2	P1	P0	Set page address
Set column address MSB	0	0	0	0	0	1	Y9	Y8	Y7	Y6	Set column address MSB
Set column address LSB	0	0	0	0	0	0	Y5	Y4	Y3	Y2	Set column address LSB
Set modify-read	0	0	1	1	1	0	0	0	0	0	Set modify-read mode
Reset modify-read	0	0	1	1	1	0	1	1	1	0	release modify-read mode
Display ON/OFF	0	0	1	0	1	0	1	1	1	D	D=0: Display OFF D=1: Display ON
Set initial display line register	0	0	0	1	0	0	0	0	x'	x'	2-byte instruction to specify the initial display line to realize vertical scrolling
	0	0	x'	S6	S5	S4	S3	S2	S1	S0	
Set initial COM0 register	0	0	0	1	0	0	0	1	x'	x'	2-byte instruction to specify the initial COM0 to realize window scrolling
	0	0	x'	C6	C5	C4	C3	C2	C1	C0	
Set partial display duty ration	0	0	0	1	0	0	1	0	x'	x'	2-byte instruction to set partial display duty ratio
	0	0	D7	D6	D5	D4	D3	D2	D1	D0	
Set N-line inversion	0	0	0	1	0	0	1	1	x'	x'	2-byte instruction to set N-line inversion register
	0	0	x'	x'	x'	N4	N3	N2	N1	N0	
Release N-line inversion	0	0	1	1	1	0	0	1	0	0	Release N-line inversion mode
Reverse display ON/OFF	0	0	1	0	1	0	0	1	1	REV	REV=0: normal display REV=1: reverse display
Entire display ON/OFF	0	0	1	0	1	0	0	1	0	EON	EON=0: normal display EON=1: entire display ON

## Example Initialization Program

```
//-----  
void write_command(unsigned char datum)  
{  
    A0=0; /*Instruction register*/  
    E=1; /*Read inactive*/  
    P1 = datum; /*put data on port 1*/  
    CS1=0; /*Chip select active*/  
    RW=0; /*Write active*/  
    RW=1; /*Write inactive; latch in data*/  
    CS1=1; /*Chip select inactive*/  
}  
  
//-----  
void write_data(unsigned char datum)  
{  
    A0=1; /*DDRAM data register*/  
    E=1;  
    P1=datum;  
    CS1=0;  
    RW=0;  
    RW=1;  
    CS1=1;  
}  
  
//-----  
void lcd_init(void){  
    write_command(0xA2); //ICON OFF;  
    write_command(0xAE); //Display OFF  
  
    write_command(0x48); //Set Duty ratio  
    write_command(0x80); //No operation  
    write_command(0xa1); //Set scan direction //changed from 0 to 1  
    write_command(0xc8); //SHL select  
    write_command(0x40); //Set START LINE  
    write_command(0x00);  
    write_command(0xab); //OSC on  
  
    write_command(0x64); //3x  
    delay(2000);  
    write_command(0x65); //4x  
    delay(2000);  
    write_command(0x66); //5x  
    delay(2000);  
    write_command(0x67); //6x  
    delay(2000);  
  
    write_command(Ra_Rb); //RESISTER SET  
    write_command(0x81); //Set electronic volume register  
    write_command(vopcode); //n=0~3f  
  
    write_command(0x57); //1/12bias  
    write_command(0x92); //FRC and pwm  
  
    write_command(0x2C);  
    delay(20000); //200ms
```



```
write_command(0x2E);
delay(20000);//200ms
write_command(0x2F);
delay(20000);//200ms
```

```
write_command(0x92); //frc and pwm
write_command(0x38); //external mode
write_command(0x75);
//start settings for 16-level grayscale
write_command(0x97); //3frc,45pwm
```

```
write_command(0x80);
write_command(0x00);
write_command(0x81);
write_command(0x00);
write_command(0x82);
write_command(0x00);
write_command(0x83);
write_command(0x00);
```

```
write_command(0x84);
write_command(0x06);
write_command(0x85);
write_command(0x06);
write_command(0x86);
write_command(0x06);
write_command(0x87);
write_command(0x06);
```

```
write_command(0x88);
write_command(0x0b);
write_command(0x89);
write_command(0x0b);
write_command(0x8a);
write_command(0x0b);
write_command(0x8b);
write_command(0x0b);
```

```
write_command(0x8c);
write_command(0x10);
write_command(0x8d);
write_command(0x10);
write_command(0x8e);
write_command(0x10);
write_command(0x8f);
write_command(0x10);
```

```
write_command(0x90);
write_command(0x15);
write_command(0x91);
write_command(0x15);
write_command(0x92);
write_command(0x15);
write_command(0x93);
write_command(0x15);
```

```
write_command(0x94);
write_command(0x1a);
write_command(0x95);
write_command(0x1a);
write_command(0x96);
write_command(0x1a);
```

```
write_command(0x97);  
write_command(0x1a);
```

```
write_command(0x98);  
write_command(0x1e);  
write_command(0x99);  
write_command(0x1e);  
write_command(0x9a);  
write_command(0x1e);  
write_command(0x9b);  
write_command(0x1e);
```

```
write_command(0x9c);  
write_command(0x23);  
write_command(0x9d);  
write_command(0x23);  
write_command(0x9e);  
write_command(0x23);  
write_command(0x9f);  
write_command(0x23);
```

```
write_command(0xa0);  
write_command(0x27);  
write_command(0xa1);  
write_command(0x27);  
write_command(0xa2);  
write_command(0x27);  
write_command(0xa3);  
write_command(0x27);
```

```
write_command(0xa4);  
write_command(0x2b);  
write_command(0xa5);  
write_command(0x2b);  
write_command(0xa6);  
write_command(0x2b);  
write_command(0xa7);  
write_command(0x2b);
```

```
write_command(0xa8);  
write_command(0x2f);  
write_command(0xa9);  
write_command(0x2f);  
write_command(0xaa);  
write_command(0x2f);  
write_command(0xab);  
write_command(0x2f);
```

```
write_command(0xac);  
write_command(0x32);  
write_command(0xad);  
write_command(0x32);  
write_command(0xae);  
write_command(0x32);  
write_command(0xaf);  
write_command(0x32);
```

```
write_command(0xb0);  
write_command(0x35);  
write_command(0xb1);  
write_command(0x35);  
write_command(0xb2);  
write_command(0x35);
```

```
write_command(0xb3);
write_command(0x35);

write_command(0xb4);
write_command(0x38);
write_command(0xb5);
write_command(0x38);
write_command(0xb6);
write_command(0x38);
write_command(0xb7);
write_command(0x38);

write_command(0xb8);
write_command(0x3a);
write_command(0xb9);
write_command(0x3a);
write_command(0xba);
write_command(0x3a);
write_command(0xbb);
write_command(0x3a);

write_command(0xbc);
write_command(0x3c);
write_command(0xbd);
write_command(0x3c);
write_command(0xbe);
write_command(0x3c);
write_command(0xbf);
write_command(0x3c);
    //end settings for 16-level grayscale
write_command(0x38);
write_command(0x74);
write_command(0xaf); //Display ON
}

//-----
```

## Quality Information

Test Item	Content of Test	Test Condition	Note
High Temperature storage	Endurance test applying the high storage temperature for a long time.	+70°C , 48hrs	2
Low Temperature storage	Endurance test applying the low storage temperature for a long time.	-20°C , 48hrs	1,2
High Temperature Operation	Endurance test applying the electric stress (voltage & current) and the high thermal stress for a long time.	+60°C 48hrs	2
Low Temperature Operation	Endurance test applying the electric stress (voltage & current) and the low thermal stress for a long time.	-10°C , 48hrs	1,2
High Temperature / Humidity Operation	Endurance test applying the electric stress (voltage & current) and the high thermal with high humidity stress for a long time.	+40°C , 90% RH , 48hrs	1,2
Thermal Shock resistance	Endurance test applying the electric stress (voltage & current) during a cycle of low and high thermal stress.	-0°C,30min -> 25°C,5min -> 50°C,30min = 1 cycle 10 cycles	
Vibration test	Endurance test applying vibration to simulate transportation and use.	10-55Hz , 15mm amplitude. 60 sec in each of 3 directions X,Y,Z For 15 minutes	3
Static electricity test	Endurance test applying electric static discharge.	VS=800V, RS=1.5kΩ, CS=100pF One time	

**Note 1:** No condensation to be observed.

**Note 2:** Conducted after 4 hours of storage at 25°C, 0%RH.

**Note 3:** Test performed on product itself, not inside a container.

## Precautions for using LCDs/LCMs

See Precautions at [www.newhavendisplay.com/specs/precautions.pdf](http://www.newhavendisplay.com/specs/precautions.pdf)

## Warranty Information and Terms & Conditions

[http://www.newhavendisplay.com/index.php?main\\_page=terms](http://www.newhavendisplay.com/index.php?main_page=terms)