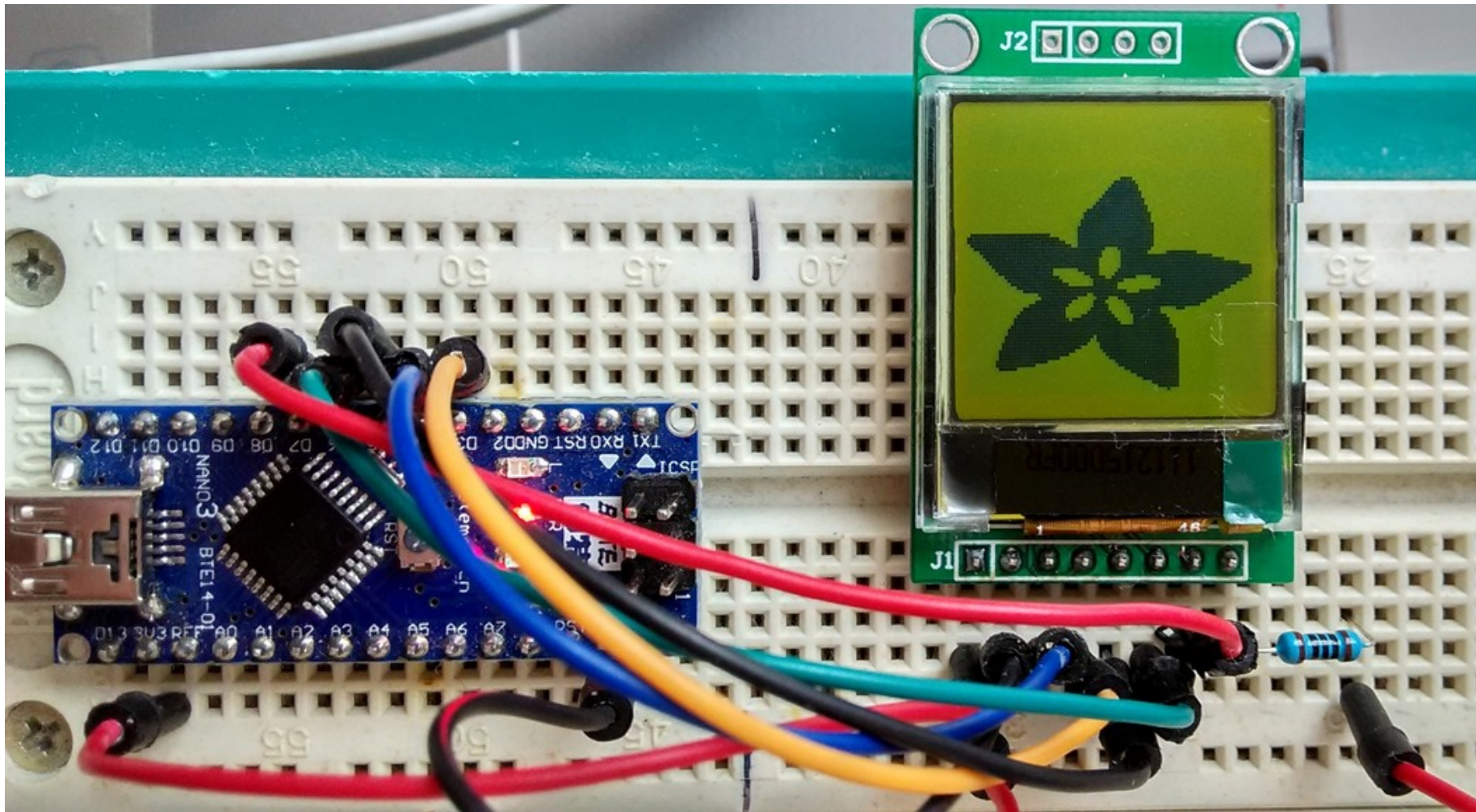


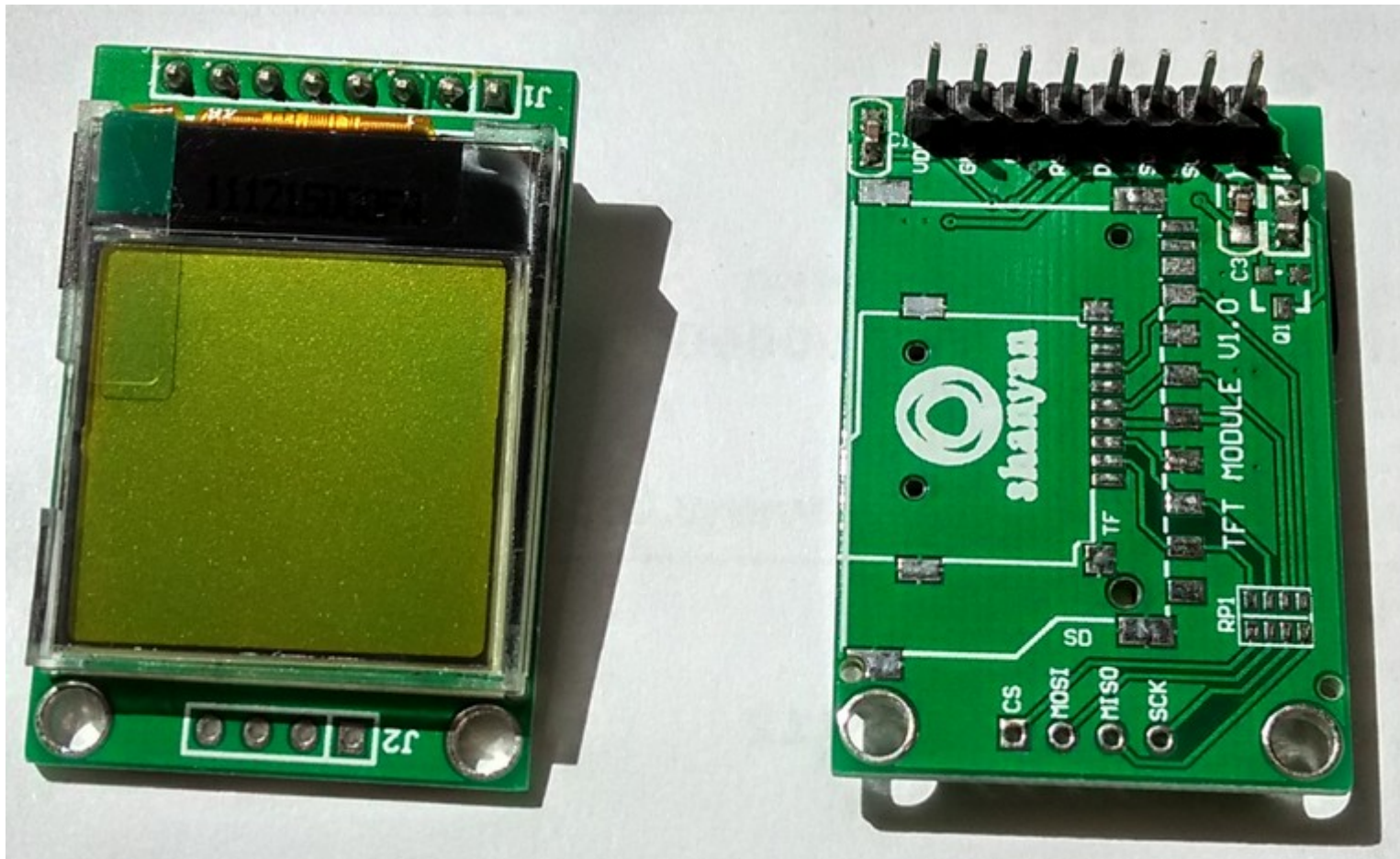
# Vegyes témakörök



## 6. ST7585 monokróm grafikus LCD vezérlése

# Shanyan TFT module 1.0

- Monokróm grafikus LCD 96\*64 képpont + 1 ikonsor
- Sitronix ST7585 vezérlő, SPI vezérléssel, LED megvilágítással
- A modul SD kártyával kiegészíthető



# Shanyan TFT module 1.0 kivezetések

---

- VDD (Voltage Drain Drain) tápfeszültség (+3,3 V)
- GND (Ground) a tápegység közös pontja
- CS (Chip Select) modul aktiválásához alacsony szintre kell húzni
- RST (Reset) a modult alaphelyzetbe állítja
- D/C (Data/Command) adat vagy parancsküldés választó vonal
- SDI (Serial Data Input) SPI soros adatbemenet
- SCK (Serial Clock) SPI órajel
- BL (Backlight) LED megvilágítás ( $I_f = 20 \text{ mA}$ ,  $U_f = 2,1 \text{ V typ.}$ )



# ST7585 vezérlő

- LCD 66 x 102 pontmátrix vezérlő/meghajtó (esetünkben 64x96)
- 8-bit, SPI, I2C kommunikáció (esetünkben SPI)
- **Esetünkben a 8 pixeles sávok Y címe fordítva működik!**  
A képernyő bal felső sarkában  $Y = 7$ ,  $X = 0$ , s az adatbájt LSB bitje szabja meg a képpont láthatóságát.

## Data Structure

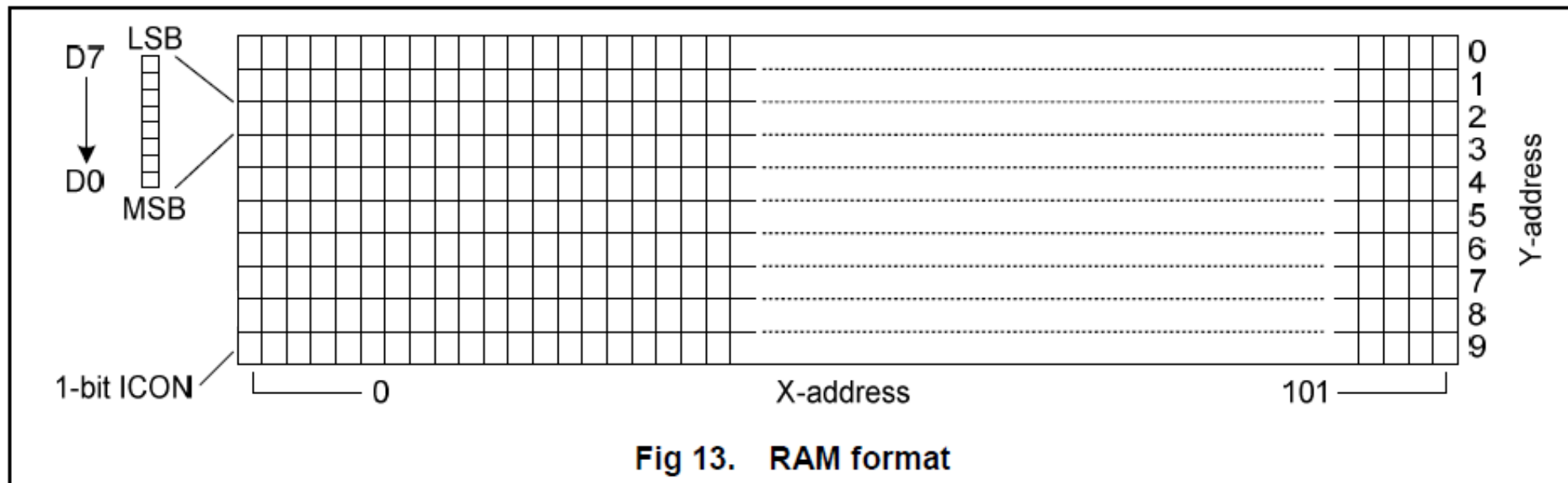
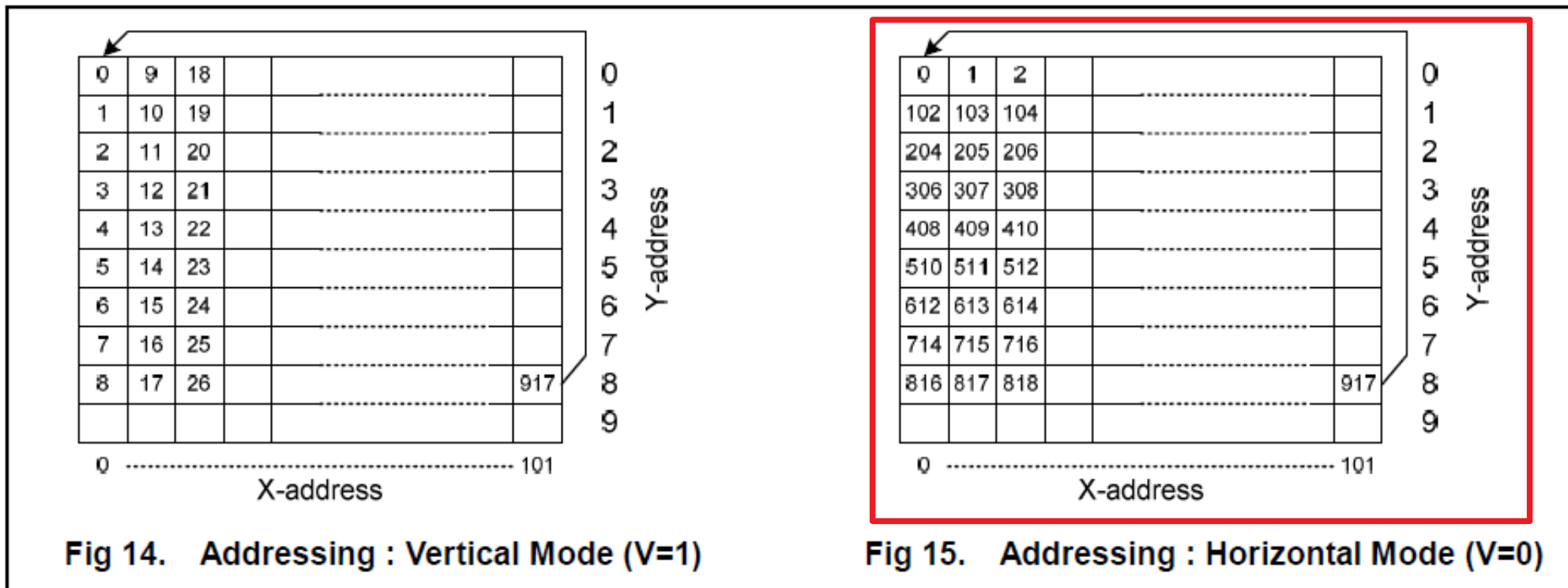


Fig 13. RAM format

# ST7585 vezérlő

- Mi a *Horizontal* (vízszintes) *címzési módot* használjuk
- Ebben az egymás után kiküldött adatbájtok az Y címmel kiválasztott sáv X irányú szomszédos képpontjait adják meg. Tehát minden bájttal 8 db egymás alatt felkvő pontot ír le, s minden adatbájttal 1 képpontot lép X irányba.



# ST7585 parancsok

- Az inicializálás és a vezérlés az alábbi parancsokkal történik

H=0 or 1 (H-Flag Independent)											
INSTRUCTION	A0	R/W (RWR)	COMMAND BYTE								DESCRIPTION
			D7	D6	D5	D4	D3	D2	D1	D0	
NOP	0	0	0	0	0	0	0	0	0	0	No operation
Function Set	0	0	0	0	1	0	0	PD	V	H	Power down; entry mode; Select instruction table
Write Data	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data to RAM

H=0 (Basic Instruction)												
INSTRUCTION	A0	R/W (RWR)	COMMAND BYTE								DESCRIPTION	
			D7	D6	D5	D4	D3	D2	D1	D0		
Display Control	0	0	0	0	0	0	0	1	D	0	E	Sets display configuration
Set Y Address of RAM	0	0	0	1	0	0	Y3	Y2	Y1	Y0	Sets Y address of RAM $0 \leq Y \leq 9$	
Set X Address of RAM	0	0	1	X6	X5	X4	X3	X2	X1	X0	Sets X address of RAM $0 \leq X \leq 101$	

H=1 (Extended Instruction)											
INSTRUCTION	A0	R/W (RWR)	COMMAND BYTE								DESCRIPTION
			D7	D6	D5	D4	D3	D2	D1	D0	
Set V0	0	0	1	V04	V03	V02	V01	V00	0	0	Set $V_{OP}$ parameter to register
Set Test Mode	0	0	0	0	1	1	0	T1	T0	TEN	Select test mode

# ST7585 parancsok – Function Set

## ■ A Function Set paranccsal

- ❖ Választhatunk az alap és a kibővített parancskészlet között ( H bit)
- ❖ Power Down módba állíthatjuk a vezérlőt (PD = 1 esetén)
- ❖ Választhatunk a Horizontal (V=0) és a Vertical (V=1) mód között

### Function Set

A0	R/W	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	1	0	0	PD	V	H

Flag	Description
PD	PD=0: chip is active PD=1: chip is in power down mode All LCD outputs at VSS (display off), bias generator and V0 generator off, VOUT can be disconnected, oscillator off (external clock possible), RAM contents not cleared; RAM data can be written.
V	Select addressing mode: V=0 for Horizontal Addressing; V=1 for Vertical Addressing.
H	H=0: Basic Instruction set; H=1: Extended instruction set. Data access can be used in both instruction blocks. Refer to the instruction table.

# ST7585 parancsok - adatküldés

- Adatküldésnél a D/C bemenetet (itt A0) magas szintre kell húzni!
- **Az adatküldés** (Write Data) parancs H = 0 és H = 1 állapotban egyaránt kiadható (független a H bittől)!
- A kiküldött adatok az adat RAM soron következő címére kerülnek.
- Az adott 8 pixeles sávon belül LSB bit (itt D0) a legfelső, az MSB bit (itt D7) pedig a legalsó képpontot vezérli
- A bit 0 értéke törli (papír szín), az 1 érték pedig megjeleníti (tintasín) az adott képpontot

A0	R/W	D7	D6	D5	D4	D3	D2	D1	D0
1	0	Write Data							



# ST7585 parancsok- Display Control

- A *Display vezérlés* parancs  $H = 0$  esetén használható
- A **D** bit 0 értéke letiltja, az 1 érték pedig engedélyezi a RAM-ban tárolt értékek megjelenítését
- Az **E** bit 0 értéke a normál, 1 értéke pedig az inverz megjelenítés kiválasztására szolgál.

## H=0 (Basic Instruction)

### Display Control

This bits D and E selects the display mode.

A0	R/W	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	1	D	0	E

Flag	Description		
D,E	D	E	The bits D and E select the display mode.
	0	0	Display OFF
	0	1	All display segments on
	1	0	Normal mode
	1	1	Inverse video mode

# ST7585 parancsok- X/Y címbeállítás

- Az **Y címbeírásnál** 0 – 9 közötti számot adhatunk meg egy-egy pixelsáv, illetve az ikonsor megcímzéséhez
- $Y = 0$  a legalsó, 7 pedig a legfelső pixelsávot címzi meg
- $Y = 8$ , illetve  $Y = 9$  esetén adatbeíráskor csak a D7 bit értéke számít!
- **X beíráskor** 0 – 101 közötti (esetünkben 0 – 95 közötti) számot adhatunk meg

H=0 (Basic Instruction)											
INSTRUCTION	A0	R/W (RWR)	COMMAND BYTE								DESCRIPTION
			D7	D6	D5	D4	D3	D2	D1	D0	
Display Control	0	0	0	0	0	0	1	D	0	E	Sets display configuration
Set Y Address of RAM	0	0	0	1	0	0	Y3	Y2	Y1	Y0	Sets Y address of RAM $0 \leq Y \leq 9$
Set X Address of RAM	0	0	1	X6	X5	X4	X3	X2	X1	X0	Sets X address of RAM $0 \leq X \leq 101$

# ST7585 parancsok- kiterjesztett parancsok

- A V0 beállítással az LCD-t működtető feszültséget szabályozhatjuk, ami a kontrasztot befolyásolja. A nagyobb érték sötétebb kijelzöt eredményez.
- Alapértéknek  $V[4:0] = 0$ -val próbálkozzunk! Gyenge kontraszt esetén növeljük 0x20-ra az értéket! Ennél nagyobb értéknél besötétülhet a képernyő.

H=1 (Extended Instruction)											
INSTRUCTION	A0	R/W (RWR)	COMMAND BYTE								DESCRIPTION
			D7	D6	D5	D4	D3	D2	D1	D0	
Set V0	0	0	1	V04	V03	V02	V01	V00	0	0	Set V <sub>OP</sub> parameter to register
Set Test Mode	0	0	0	0	1	1	0	T1	T0	TEN	Select test mode

- A teszt mód beállítása nem derült ki, hogy mire jó....

# lcd\_IO.ino – egyszerű próba

- Szoftveres SPI vezérléssel
- Bekötési vázlat:

Display	Arduino
VDD	3,3 V
GND	GND
CS	D4
RST	D3
D/C	D5
SDI	D6
SCK	D7
BL	100 Ω-mal VDD-re

```
#define SDI    6
#define RSX    5
#define CSX    4
#define SCLX   7
#define RESX   3
#define Function_Set      0x20
#define Display_Control   0x08
#define Set_Y_Address     0x40
#define Set_X_Address     0x80
#define Set_V0            0x80
#define Set_Test_Mode     0x30
```

```
void send_cmd(unsigned char cmd,
unsigned char dat) {
    digitalWrite(RSX,LOW);
    digitalWrite(CSX,LOW);
    LCDShiftWrite(cmd|dat);
    digitalWrite(CSX,HIGH);
    digitalWrite(RSX,HIGH);
}
```

```
void send_dat(unsigned char dat) {
    digitalWrite(CSX,LOW);
    LCDShiftWrite(dat);
    digitalWrite(CSX,HIGH);
}
```



# lcd\_IO.ino – egyszerű próba

```
void LCDInit() {
    digitalWrite(RESEX,LOW);          // HW reset
    delay(10);
    digitalWrite(RESEX,HIGH);
    delay(10);
    send_cmd(Function_Set,0x01);     // Extended command set selection
    send_cmd(Set_V0,0x20);           // Set contrast
    send_cmd(Set_Test_Mode,0x02);
    send_cmd(Function_Set,0x00);     // Basic Command set selection
    send_cmd(Display_Control,0x04); // Normal Display mode selection
}

void LCD_set_XY(unsigned char x,unsigned char y) {
    send_cmd(Set_X_Address,x);      // X cím beállítás
    send_cmd(Set_Y_Address,y);      // Y (sáv) cím beállítás
}

void cls(void) {                    // Képernyő törlése
    send_cmd(Set_X_Address,0);
    send_cmd(Set_Y_Address,0);
    for(int i=0;i<960;i++) send_dat(0x00); // RAM terület nullázása
    send_cmd(Set_X_Address,0);
    send_cmd(Set_Y_Address,0);
}
```

# lcd\_IO.ino – egyszerű próba

```
void putch(unsigned char x, unsigned char y, unsigned int ch) {  
    unsigned char i;  
    send_cmd(Set_X_Address,x);  
    send_cmd(Set_Y_Address,y);  
    for(i=0;i<5;i++)  
        send_dat(FONT[(ch-0x20)*5+i]);  
}
```

A 8 pixeles sávokba 5x8 pixeles  
**karakereket írunk**  
x pozicionálás pixelenként,  
y pozicionálás sávonként

**FONT[]** előre definiált karakterkép a flash memóriában eltárolva  
(const unsigned char típusú)

```
void putstr(unsigned char x, unsigned char y, char *str) {  
    while(*str!=0) {  
        putch(x,y,*str++); x=x+6;  
    }  
}
```

```
void SHOW_BMP() {  
    unsigned int i,j,n=0;  
    for(i=0;i<8;i++) {  
        for(j=0;j<96;j++) {  
            send_cmd(Set_Y_Address,7-i);  
            send_cmd(Set_X_Address,j);  
            send_dat(BMP[n++]);  
        }  
    }  
}
```

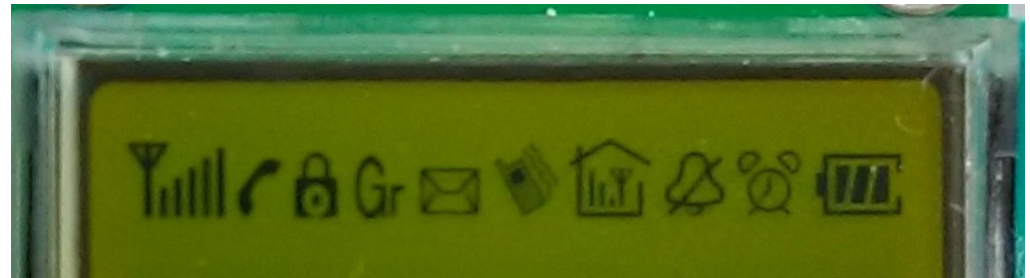
**Bitkép kirajzolása**  
Y transzformáció a fordított  
számozás miatt

Itt **BMP[]** egy előre definiált bitkép a flash  
memóriában eltárolva (const unsigned char típusú)

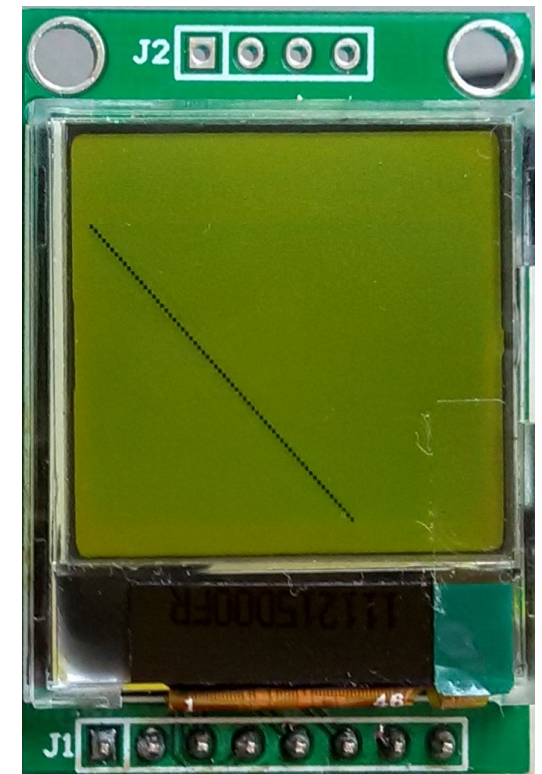


# lcd\_IO.ino – egyszerű próba

```
// Az ikonsor megjelenítése
void SHOW_ICO() {
  unsigned char i;
  for(i=0; i<96; i++){
    send_cmd(Set_Y_Address,8);
    send_cmd(Set_X_Address,i);
    send_dat(0xFF);
  }
}
```

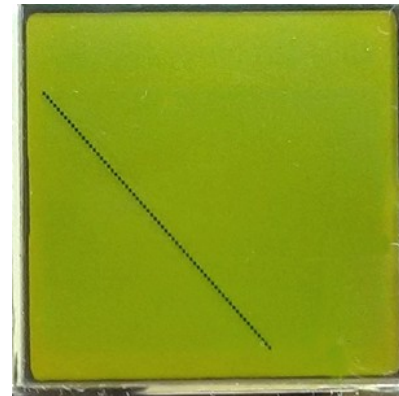


```
// Egy ferde vonal megrajzolása
void SHOW_LINE() {
  unsigned int i,j;
  const unsigned char line[8] = {
    0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80
  };
  for(i=0; i<8; i++) {
    send_cmd(Set_Y_Address,7-i);
    for(j=0;j<8; j++) {
      send_cmd(Set_X_Address,j+8*i);
      send_dat(line[j]);
    }
  }
}
```



# lcd\_IO.ino – egyszerű próba

```
void setup() {  
  pinMode(SDI,OUTPUT);  
  pinMode(RSX,OUTPUT);  
  pinMode(CSX,OUTPUT);  
  pinMode(SCLX,OUTPUT);  
  pinMode(RESX,OUTPUT);  
  LCDInit();  
}  
  
void loop() {  
  SHOW_ICO();  
  putstr(30,5,"ST7585");  
  putstr(15,3,"96*64 GLCD");  
  putstr(30,1,"Arduino");  
  delay(5000);  
  
  cls();  
  SHOW_LINE();  
  delay(5000);  
  cls();  
  
  SHOW_BMP();  
  delay(5000);  
  cls();  
}
```





# Adafruit\_ST7585 library

- Készítsünk könyvtárat az `Adafruit_PCD8544` mintájára!  
Az ikonokkal egyelőre nem foglalkozunk!
- `Adafruit_ST7585.h` módosítások releváns részletei

```
#define LCDWIDTH 96
#define LCDHEIGHT 64
#define ST7585_POWERDOWN 0x04
#define ST7585_ENTRYMODE 0x02
#define ST7585_EXTENDEDINSTRUCTION 0x01
```

```
#define ST7585_DISPLAYBLANK 0x0
#define ST7585_DISPLAYNORMAL 0x4
#define ST7585_DISPLAYALLON 0x1
#define ST7585_DISPLAYINVERTED 0x5
#define ST7585_FUNCTIONSET 0x20
#define ST7585_DISPLAYCONTROL 0x08
#define ST7585_SETYADDR 0x40
#define ST7585_SETXADDR 0x80
#define ST7585_SETTESTMODE 0x03
#define ST7585_SETVOP 0x80
```

```
void begin(uint8_t contrast = 0);
void setContrast(uint8_t val);
```

**(nincs bias beállítás!)**

# Adafruit\_ST7585.cpp módosítások

```
//--- Inicializálás releváns részletei -----
command(ST7585_FUNCTIONSET | ST7585_EXTENDEDINSTRUCTION );
command( ST7585_SETVOP | contrast);
command(ST7585_SETTESTMODE | 0x02);
command(ST7585_FUNCTIONSET);
command(ST7585_DISPLAYCONTROL | ST7585_DISPLAYNORMAL);
updateBoundingBox(0, 0, LCDWIDTH-1, LCDHEIGHT-1);
display(); // Push out buffer to LCD (will show the AFI logó)

//--- A megjelenítés releváns részletei -----
void Adafruit_ST7585::display(void) {
  uint8_t col, maxcol, p;
  for(p = 0; p < LCDHEIGHT/8; p++) {
    command(ST7585_SETYADDR | 7-p); // different from the Nokia 5110!!!
    command(ST7585_SETXADDR | col);
    digitalWrite(_dc, HIGH);
    if (_cs > 0) digitalWrite(_cs, LOW);
    for(col = 0; col <= maxcol; col++) {
      spiWrite(ST7585_buffer[(LCDWIDTH*p)+col]);
    }
    if (_cs > 0) digitalWrite(_cs, HIGH);
  }
  command(ST7585_SETYADDR );
}
```

# library\_test.ino

- A félig kész programkönyvtárat az alábbi programmal teszteljük:

```
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Adafruit_ST7585.h>
Adafruit_ST7585 display = Adafruit_ST7585(7,6,5,4,3); //SCLK,DIN,DC,CE,RST
void setup() {
  Serial.begin(9600);
  display.begin();
  display.setContrast(0x20); // adjust contrast value for best viewing
  display.display(); // show splashscreen
  delay(2000);
  display.clearDisplay(); // clears the screen and buffer

  // draw lines from pixel to pixel
  for(int i=0; i<95; i++) {
    for (int j=0; j<65; j+=4) {
      display.drawPixel(i, j, BLACK);
    }
  }
  display.display();
  delay(5000);
  display.clearDisplay();
  ...
}
```

**Folytatás a következő oldalon!**

# library\_test.ino

---

```
// draw many lines
testdrawline();
display.display();
delay(2000);
display.clearDisplay();

// draw rectangles
testdrawrect();
display.display();
delay(2000);
display.clearDisplay();

}

void loop() {
  // do nothing
}
```

- A `testdrawline()` és `testdrawrect()` függvények megegyeznek az `Adafruit_PCD8544` programkönyvtár mintaprogramjában találhatóakkal.



# Az Arduino Nano kártya kivezetései



## NANO PINOUT

The power sum for each pin's group should not exceed 100mA

