1x10 1” display

USER MANUAL

DRAFT

Attention: Please read the manual carefully.
In case of any doubts or questions pls contact
Alfa - Zeta at info@alfazeta.pl or +48 42 6891200,
Please note that Poland is GMT+1

Rev. 0.1 / 28 Nov 2019

Alfa-Zeta Co. Ltd.
ul. Starorudzka 6a
93-418 Łódź
POLAND

tel. +48 42 689 12 00
tel. +48 42 689 12 01
tel. +48 42 689 12 02
fax +48 42 689 12 03

http://www.alfazeta.pl
http://www.flipdots.com
http://www.swiatlowody.com
http://led.alfazeta.pl

info@alfazeta.pl
info@flipdots.com
info@swiatlowody.com
linfo@alfazeta.pl

corporate website
visual information displays
fiber optic lighting
LED lighting

SAFETY PRECAUTIONS

- assemble process must be performed by qualified personnel only,
- always check connection before installing components,
- any additional installations must be made according to local safety regulations,

- anti-shock protection system must be made according to local regulations.

Working conditions

Operating temperature: -40°C to 80 °C
Relative Humidity: 5 to 95% non-condensing (at 40°C)

For most of applications these displays has to be covered in a housing to protect against dust and humidity. The housing needs to be ventilated in order to reduce temperature resulting from sun load.

It is possible to use the displays without a housing however cleaning process would be necessary for proper operation.

Connections, sockets and buttons

For proper operation you need to connect the following:

- 24V DC (+ and -)
- RS485 (+ and -) (by means of RJ11 socket)

See below description of connections:

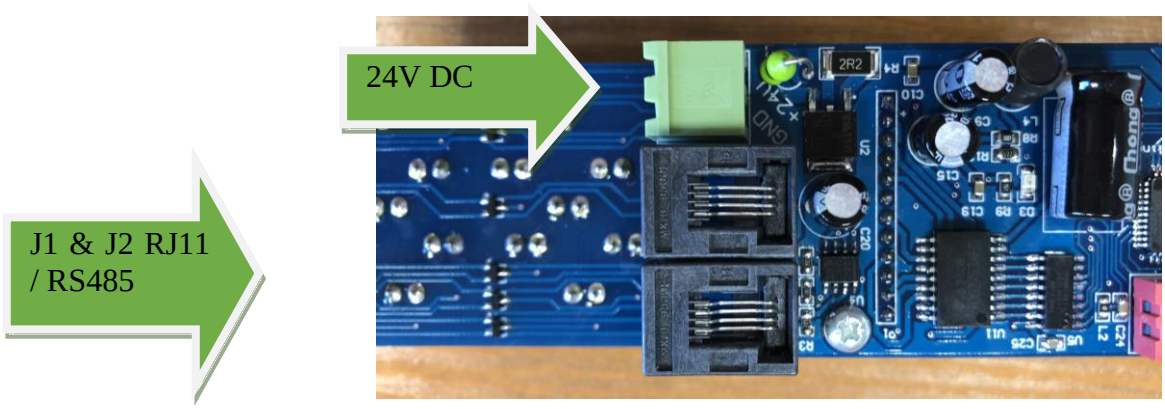
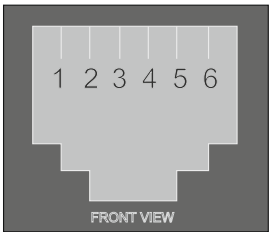


Image 1. Description of inputs / outputs

Name	Description
24V DC	24V DC Power supply. Please observe polarity and see “Power consumption considerations’
J1	R11 socket for data connection. In series with J2 and J3. See below pinout description.
J2	R11 socket for data connection. In series with J1 and J3. See below pinout description.
DS#1	See below for explanation
DS#2	See below for explanation

J1 & J2 description (RJ11 standard)



Pin	*Function*
1	N/C
2	RS485+
3	RS485+
4	RS485-
5	RS485-
6	N/C

Dip Switch DS#2 : speed of transmission settings.

Value	ON			Speed
	1	2	3	
0	↓	↓	↓	N/A
1	↑	↓	↓	N/A
2	↓	↑	↓	N/A
3	↑	↑	↓	9600
4	↓	↓	↑	19200
5	↑	↓	↑	38400
6	↓	↑	↑	57600
7	↑	↑	↑	9600
	OFF			

Dip Switch DS#1 : address and test mode .

Position	*Meaning*
0 – 5	Address in binary code (natural)
6	Not used
7	Test mode: ON – test mode ON, OFF – test mode OFF – normal operation

Image 2 A controller

PLEASE DISREGARD ALL OTHER CONNECTIONS.

Connection to Arduino

Arduino does not have built in rs485 interface. In order to drive a panel a converter is needed. Google for 'RS485 breakout' or ask for advice.

Connection to PC/MAC or RAspi

Most of PC/MAC does not have RS485 port. In order to drive a panel from a PC or MAC, USB to RS485 converter or Ethernet – to – RS485 converter is needed.

Driving protocol

Protocols are two:

1) The simple one

   0XFE
   Address [0x00 to 0x1F or 0xFF for all connected displays]
   Data1 [0x01 -> 1, 0x02 -> 2, 0x0A – blank]
   …
   Data10
   0x8F

2) The more sophisticated one:
   0x80
   Command [0x8E – load data and do not show them, 0x8D – load data and show them, 0x82 – show loaded data]
   Address [0x00 to 0x1F or 0xFF for all connected displays]
   Data1 [each bit is responsible for one flag settings: bit0 is flag A… bit6 is flag G, bit7 is not used]
   …
   Data10
   0x8F

Cleaning and maintenance

It is strongly advised to protect displays with a housing in order to prevent dust build up.

Usage of vacuum cleaner.
Vacuum cleaner is a good device because it allows to remove dust from the sign. **Damages caused by improper cleaning are not covered by a guarantee.**

DO NOT use any kind of lubricants or grease. This may help for a while but later block the flags completely.

Code example:

It is showing Fibonacci sequence, code is for processing.org

```
// Protocol

// 0xFE
// adress
// data 1... data 10 (0x01 - 1, 0x02-2... 0x0A space)
//0x8F


//0x80
// 0x8E - no refresh / 0x8D - with refresh / 0x82 - refresh command
// addres - 0xFF- ALL
// data1...data10
//0x8F


import processing.serial.*;


// The serial port:


Serial myPort;

int digit[]={0, 1, 2, 4, 8, 16, 32, 64, 128, 0, 0, 0, 0, };
```

```
  long last_digit=0;
  long before_last_digit=0;
  long this_digit=1;



// void ShowError(int adres);


void SendDigits(long number_to_send, int leading_zeros, int adres)
{

  int u;
  long digit;
  int wasdigit=0;
  int send_buffer[]={ 0xFE, 0xFF, 0x02 , 0x02,  0x02, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08,
0x8F, 0x8F};

  if (number_to_send>9999999999L) return;


for (u=9;u>-1;u--) { digit=(int)(number_to_send/pow(10,u));
                    // if (digit>0) wasdigit=1;
                    // if ((digit==0)&&(wasdigit==0)&&(leading_zeros==0)) digit=0x0A;

                     send_buffer[11 -u]=(int) digit;
                     number_to_send=number_to_send-(digit*(int)pow(10,u));
}

send_buffer[1]=adres;

            for (u=0;u<14; u++) { myPort.write(send_buffer[u]); // send transmission for one panel
//                                  println(send_buffer[u]);
            }

}




void setup() {


  size(300, 300, JAVA2D);

  frameRate(15);
```

```
  printArray(Serial.list());

// Open the port you are using at the rate you want:
myPort = new Serial(this, Serial.list()[4], 57600);

// if you want to show a static image, you can put it here.


}


void draw() {



  // COUNTING DEMO
  /*
  for (t=0;t<9999999999L;t++)  { SendDigits(t,0);
                                 delay(200);
                                 }
  */

  // Fibonacci sequence



  SendDigits (this_digit,0, 0xFF);
  before_last_digit=last_digit;

  last_digit=this_digit;

  this_digit=last_digit+before_last_digit;

  if (this_digit>9999999999L)  {
                               last_digit=0;
                               before_last_digit=0;
                               this_digit=1;
                               };

  delay(1000);



}
```