

## E-Ink display 2,9"

### 1. POPIS

E-Ink displej se vyznačuje dobrou čitelností na přímém slunci, širokými pozorovacími úhly a nízkou energetickou náročností. Tento typ displejů najde své využití zejména v elektronických čtečkách knih, jelikož neunavuje oči jako klasické LCD, LED nebo OLED displeje. Tento konkrétní E-Ink komunikuje prostřednictvím komunikačního rozhraní SPI. Displej je kompatibilní s platformou Arduino, Raspberry a ST(ARM).



Základní charakteristika:

- rozlišení 296 x 128
- úhlopříčka 2,9"
- černobílý displej
- rozhraní SPI

### 2. SPECIFIKACE

<b>Napájení</b>	3,3 V	<b>Počet barev</b>	2
<b>Logická napěťová úroveň</b>	3,3 V	<b>Pozorovací úhel</b>	cca 170°
<b>Pracovní proud</b>	až 10 mA	<b>Komunikační rozhraní</b>	SPI
<b>Spotřeba při vykreslení</b>	cca 27 mW	<b>Operační teplota</b>	0 až 50 °C
<b>Rozlišení displeje</b>	296 x 128 pixelů	<b>Rozteč mont. otvorů (mm)</b>	84 x 33
<b>Úhlopříčka</b>	2,9"	<b>Rozměry displeje (mm)</b>	67 x 29
<b>Úplná odezva (při 25 °C)</b>	cca 680 ms	<b>Rozměry zařízení (mm)</b>	90 x 38



### 3. ZAPOJENÍ

Pro správnou funkčnost je nutné snížit logickou napěťovou úroveň datových pinů Arduino, a to na úroveň 3,3 V. Pokud tak nebude učiněno, displej může být poškozen a nemusí pracovat správně. Doporučujeme využít level-shifter 1489315030 nebo napěťový dělič (vizte <https://navody.arduino-shop.cz/technikuv-blog/zmena-logicky-ch-napetovych-urovni.html>).

Pin displeje	Pin Arduino Uno
BUSY	D7
RST	D8
DC	D9
CS	D10
CLK	D13
DIN	D11
GND	GND
VCC	3,3 V



### 4. UKÁZKA PROGRAMU

Pro správnou funkci si uživatel stáhne knihovnu [epd2in9.h](#) (společně s příkladem vykreslování bitmapy).

```
#include <SPI.h>
#include <epd2in9.h>
#include <epdpaint.h>

#define COLORED 0
#define UNCOLORED 1

unsigned char image[1024];
Paint paint(image, 0, 0); // width should be the multiple of 8
Epd epd;
unsigned long time_start_ms;
unsigned long time_now_s;
```

```

void setup() {
  Serial.begin(9600);
  if (epd.Init(lut_full_update) != 0) {
    Serial.print("e-Paper init failed");
    return;
  }

  if (epd.Init(lut_partial_update) != 0) {
    Serial.print("e-Paper init failed");
    return;
  }

  epd.ClearFrameMemory(0xFF);
  epd.DisplayFrame();
  time_start_ms = millis();
}

void loop() {
  epd.ClearFrameMemory(0xFF);
  // put your main code here, to run repeatedly:
  time_now_s = (millis() - time_start_ms) / 1000;
  char time_string[] = {'0', '0', ':', '0', '0', '\0'};
  time_string[0] = time_now_s / 60 / 10 + '0';
  time_string[1] = time_now_s / 60 % 10 + '0';
  time_string[3] = time_now_s % 60 / 10 + '0';
  time_string[4] = time_now_s % 60 % 10 + '0';

  paint.SetWidth(32);
  paint.SetHeight(250);
  paint.SetRotate(ROTATE_90);

  paint.Clear(UNCOLORED);
  paint.DrawStringAt(0, 4, "Arduino-shop.cz", &Font24, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 70, 20, paint.GetWidth(), paint.GetHeight());

  paint.SetWidth(32);
  paint.SetHeight(150);
  paint.SetRotate(ROTATE_90);

  paint.Clear(UNCOLORED);
  paint.DrawStringAt(0, 4, "Spusteno: ", &Font24, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 30, 20, paint.GetWidth(), paint.GetHeight());

  paint.SetWidth(32);
  paint.SetHeight(96);
  paint.SetRotate(ROTATE_90);

  paint.Clear(UNCOLORED);
  paint.DrawStringAt(0, 4, time_string, &Font24, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 30, 180, paint.GetWidth(), paint.GetHeight());
  epd.DisplayFrame();
}

```