|  |
| --- |
| # Plik st7796\_horizontal.py  import machine import framebuf import time  WIDTH = const(480) # 1 HEIGHT = const(320)  RED = const(0b000\_00000\_11111\_000) # 2 YELLOW = const(0b111\_00000\_11111\_111) GREEN = const(0b111\_00000\_00000\_111) CYAN = const(0b111\_11111\_00000\_111) BLUE = const(0b000\_11111\_00000\_000) MAGENTA = const(0b000\_11111\_11111\_000) WHITE = const(0b111\_11111\_11111\_111) BLACK = const(0b000\_00000\_00000\_000)  class ST7796(framebuf.FrameBuffer): # 3    def \_\_init\_\_(self, spi, cs, dc, rst): # 4  self.spi = spi  self.cs = cs  self.dc = dc  self.rst = rst  self.cs.init(mode=machine.Pin.OUT, value=1) # 5  self.dc.init(mode=machine.Pin.OUT, value=1)  self.rst.init(mode=machine.Pin.OUT, value=1)    self.array = bytearray(WIDTH \* HEIGHT \* 2) # 6  super().\_\_init\_\_(self.array, WIDTH, HEIGHT, framebuf.RGB565) # 7    self.rst(0) # 8  time.sleep\_ms(15)  self.rst(1)  time.sleep\_ms(15)    self.write\_cmd(0x3A) # COLMOD: Pixel Format Set  self.write\_data(0x05) # 16-bit pixel format    self.write\_cmd(0x36) # Memory Access Control  self.write\_data(0b11101100); # MY=1 MX=1 MV=1 ML=0 BGR=1 MH=1    self.write\_cmd(0x2B) # Row range 0..319  self.write\_data(0x00)  self.write\_data(0x00)  self.write\_data(0x01)  self.write\_data(0x3F)    self.write\_cmd(0x2A) # Col range 0..479  self.write\_data(0x00)  self.write\_data(0x00)  self.write\_data(0x01)  self.write\_data(0xDF)    self.write\_cmd(0x11) # Sleep Out  self.write\_cmd(0x29) # Display ON    def write\_data(self, data): # 8  self.dc(1)  self.cs(0)  self.spi.write(bytes([data]))  self.cs(1)    def write\_cmd(self, cmd): # 9  self.dc(0)  self.cs(0)  self.spi.write(bytes([cmd]))  self.cs(1)    def refresh(self): # 10  self.cs(0)  self.dc(0)  self.spi.write(bytes([0x2C]))  self.dc(1)  self.spi.write(self.array)  self.cs(1)    def color(self, red, green, blue): # 11  red = int(red)  green = int(green)  blue = int(blue)    if red > 255: # 12  red = 255  if green > 255:  green = 255  if blue > 255:  blue = 255    red = red & 0xF8 # 13  green1 = (green & 0xE0) >> 5  green2 = (green & 0x1C) << 11  blue = (blue & 0xF8) << 5  color = red | green1 | green2 | blue  return color    def print\_char(self, font, char, x, y, color): # 14  try:  bitmap = font[ord(char)]  except:  bitmap = font[0]  print(f"Char {char} doesn't exist in font")    width = bitmap[0]  height = bitmap[1]  space = bitmap[2]  offset\_x = 0  offset\_y = 0    for byte in bitmap[3:]:  for bit in range(8):  if byte & (1<<bit):  self.pixel(offset\_x+x, offset\_y+y+bit, color)    offset\_x += 1  if offset\_x == width:  offset\_x = 0  offset\_y += 8    return width + space    def print\_text(self, font, text, x, y, align="L", color=1): # 15  width = self.get\_text\_width(font, text)    if align == "R":  x = WIDTH - width  elif align == "C":  x = WIDTH//2 - width//2  elif align == "r":  x = x - width + 1  elif align == "c":  x = x - width//2    for char in text:  x += self.print\_char(font, char, x, y, color)    def get\_text\_width(self, font, text): # 16  total = 0  last\_char\_space = 0  for char in text:  bitmap = font.get(ord(char), font[0])  total += bitmap[0]  total += bitmap[2]  last\_char\_space = bitmap[2]    return total - last\_char\_space |

Listing 1. Kod pliku st7796\_horizontal.py