Resources

main.py

```
from machine import UART, Pin import time import my_oled

uart = UART(1, baudrate=9600, tx=43, rx=44)

led = Pin(23, Pin.OUT)

START_BYTE = 0x41 SENDER_ID = 0x03 # HMI ID END_BYTE = 0x42

my_oled.print_text("Waiting...", 0, 0)

def flash_led(duration=0.1): led.on() time.sleep(duration) led.off()

def forward_message(msg): time.sleep(1) # ~1s delay to give MQTT board time to recover uart.write(msg) print("Forwarded:", [hex(b) for b in msg]) flash_led()

def read_message(): """ Read a 5-byte UART message if available and valid. """ if uart.any() >= 5: msg = uart.read(5) if msg and msg[0] == START_BYTE and msg[-1] == END_BYTE: print("Received:", [hex(b) for b in msg]) flash_led() return msg else: print("Invalid message.") return None

while True: # Listen and forward messages incoming = read_message() if incoming: sender = incoming[1] data = incoming[3] my_oled.print_text(f"From {hex(sender)}: {hex(data)}", 0, 0) forward_message(incoming)

time.sleep(0.1)
```

## **my_oled.py**

```
from machine import Pin, SoftI2C import ssd1306 import gfx

i2c = SoftI2C(scl=Pin(4), sda=Pin(5))

oled_width = 128 oled_height = 64 oled = ssd1306.SSD1306_I2C(oled_width, oled_height, i2c)

graphics = gfx.GFX(oled_width, oled_height, oled.pixel)

def print_text(msg, x=0, y=0): oled.fill(0) oled.text(msg, x, y) oled.show()

def clear(): oled.fill(0) oled.show()

def draw_line(x1, y1, x2, y2): graphics.line(x1, y1, x2, y2) oled.show()

def draw_filled_rect(x, y, w, h): graphics.fill_rect(x, y, w, h, 1) oled.show()
```

Final CAD Files

- *Download HMI_3D_CAD_Files*