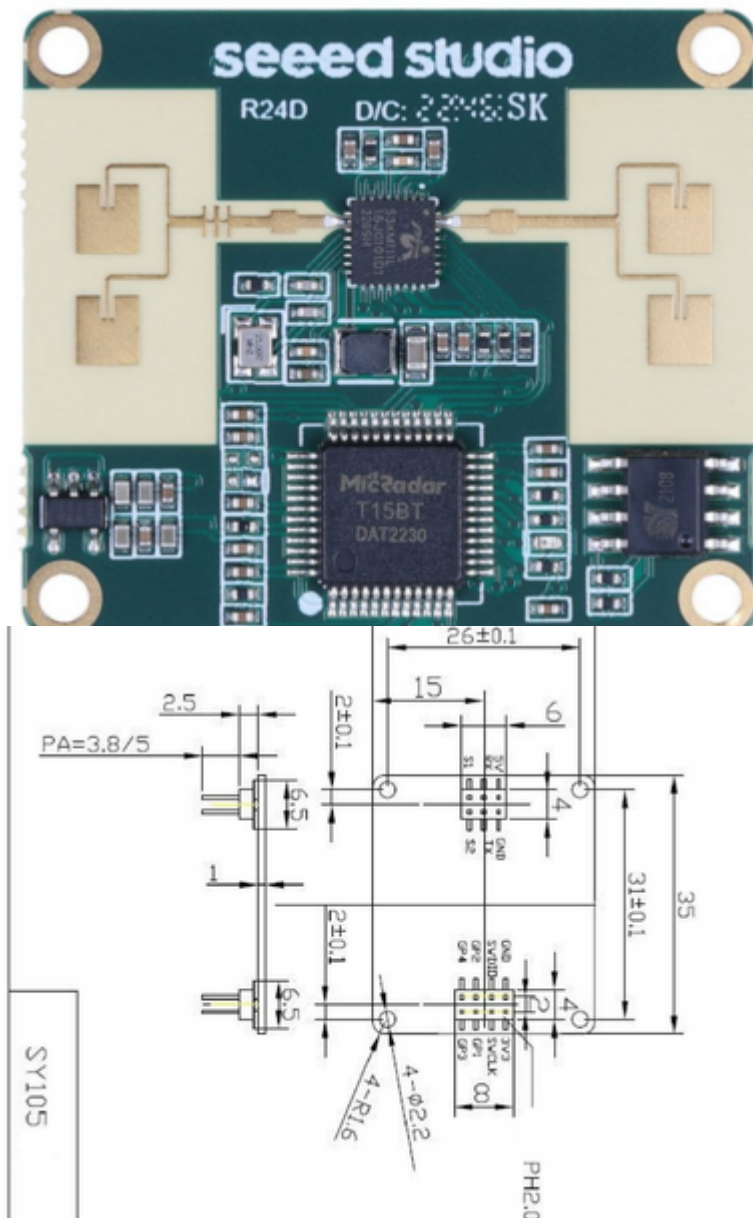
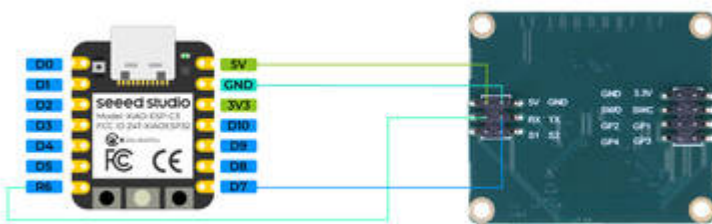
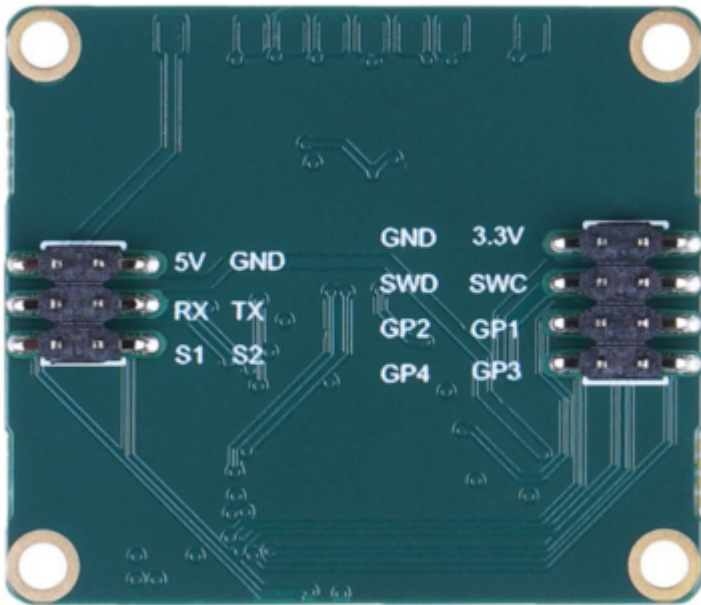


Hardware, Presence Detection, Sensor

Human Static Presence Module Lite (MR24HPC1)





XIAO ESP32C3	24GHz mmWave Human Static Presence Module Lite
5V	5V
GND	GND
RX	D6
TX	D7

Nützliche Links

- [24GHz mmWave Sensor - Human Static Presence Module Lite - human presence \(Product Details\)](#)
- [24GHz mmWave Sensor - Human Static Presence Module Lite User Manual and User Protocol](#)
- [24GHz mmWave Sensor - Human Static Presence Module Lite Datasheet](#)
- [XIAO ESP32C3 accesses Home Assistant via ESPHome service](#)
- [24GHz mmWave Sensor - Human Static Presence Module Lite \(MR24HPC1\)](#)
- [ULTIMATE IoT project ESP board & NEW mmWave sensor \(YouTube\)](#)
- [Github Link zum Code](#)

Code

xiao-esp32c3.yaml

```
substitutions:
  devicename: xiao-esp32c3
  upper_devicename: XIAO ESP32C3

esphome:
  name: $devicename
  friendly_name: $upper_devicename
  platformio_options:
    board_build.flash_mode: dio
    board_build.mcu: esp32c3
  includes:
    - R24dvd.h

esp32:
  board: esp32-c3-devkitm-1
  variant: esp32c3
  framework:
    type: esp-idf

# Enable logging
logger:
  hardware_uart: USB_SERIAL_JTAG
  level: DEBUG

packages:
  wifi: !include packages/wifi.yaml
  mqtt: !include packages/mqtt.yaml
  ota: !include packages/ota.yaml
  time: !include packages/time.yaml

# Enable Home Assistant API
api:

# captive_portal:

# part 3:
uart:
  id: uart_bus
  baud_rate: 115200
  rx_pin: 20
  tx_pin: 21

select:
  - platform: template
    name: "Standard Scene mode"
```

```
id: scene_mode
icon: mdi:hoop-house
optimistic: true
options:
  - "Living room"
  - "Area detection"
  - "Washroom"
  - "Bedroom"
initial_option: "Living room"
set_action:
  - logger.log:
      format: "set action option: %s"
      args: ["x.c_str()"]
  - uart.write: !lambda
      auto index = id(scene_mode).index_of(x);
      uint8_t value = (uint8_t)index.value() + 1;
      uint8_t crc = value + 0xB9;
      return
{0x53,0x59,0x05,0x07,0x00,0x01,value,crc,0x54,0x43};

- platform: template
  name: "Standard unmanned time"
  id: unmanned_time
  icon: mdi:timeline-clock
  optimistic: true
  options:
    - "None"
    - "10s"
    - "30s"
    - "1min"
    - "2min"
    - "5min"
    - "10min"
    - "30min"
    - "1hour"
  initial_option: "None"
  set_action:
    - logger.log:
        format: "Chosen option: %s"
        args: ["x.c_str()"]
    - uart.write: !lambda
        auto index = id(unmanned_time).index_of(x);
        uint8_t value = (uint8_t)index.value();
        uint8_t crc = value + 0x37;
        return
{0x53,0x59,0x80,0x0a,0x00,0x01,value,crc,0x54,0x43};

- platform: template
  name: "Custom Presence of perception boundary"
  id: custom_presence_of_perception_boundary
```

```
optimistic: true
options:
  - "0.5m"
  - "1.0m"
  - "1.5m"
  - "2.0m"
  - "2.5m"
  - "3.0m"
  - "3.5m"
  - "4.0m"
  - "4.5m"
  - "5.0m"
set_action:
  - logger.log:
      format: "Chosen option: %s"
      args: ["x.c_str()"]
  - uart.write: !lambda
      auto index = id(unmanned_time).index_of(x);
      uint8_t value = (uint8_t)index.value() + 1;
      uint8_t crc = value + 0xBF;
      return
{0x53,0x59,0x08,0x0a,0x00,0x01,value,crc,0x54,0x43};

- platform: template
  name: "Custom Motion trigger boundary"
  id: custom_motion_trigger_boundary
  optimistic: true
  options:
    - "0.5m"
    - "1.0m"
    - "1.5m"
    - "2.0m"
    - "2.5m"
    - "3.0m"
    - "3.5m"
    - "4.0m"
    - "4.5m"
    - "5.0m"
  set_action:
    - logger.log:
        format: "Chosen option: %s"
        args: ["x.c_str()"]
    - uart.write: !lambda
        auto index = id(unmanned_time).index_of(x);
        uint8_t value = (uint8_t)index.value() + 1;
        uint8_t crc = value + 0xC0;
        return
{0x53,0x59,0x08,0x0b,0x00,0x01,value,crc,0x54,0x43};

number:
  - platform: template
```

```
id: sensitivity
name: "Standard sensitivity"
icon: mdi:archive-check-outline
min_value: 0
max_value: 3
optimistic: false
step: 1
update_interval: 2s
set_action:
  - uart.write: !lambda
      uint8_t crc = x + 0xBA;
      return
{0x53,0x59,0x05,0x08,0x00,0x01,(uint8_t)x,crc,0x54,0x43};

- platform: template
name: "Standard Maximum detectable range of moving target"
id: moving_target_detection_max_distance
icon: mdi:map-marker-path
unit_of_measurement: "cm"
min_value: 0
max_value: 65536
step: 500
set_action:
  - uart.write: !lambda
      int h_num = (int)x>> 8;
      int l_num = (int)x & 0xff;
      int crc = 0xB6 + h_num + l_num;
      return
{0x53,0x59,0x07,0x01,0x00,0x02,(uint8_t)(h_num),(uint8_t)(l_num),(uint8_t)crc,0x54,0x43};

- platform: template
name: "Standard Maximum detectable range of stationary target"
id: static_target_detection_max_distance
icon: mdi:map-marker-path
unit_of_measurement: cm
min_value: 0
max_value: 65536
step: 500
set_action:
  - uart.write: !lambda
      int h_num = (int)x>> 8;
      int l_num = (int)x & 0xff;
      int crc = 0xB9 + h_num + l_num;
      return
{0x53,0x59,0x07,0x04,0x00,0x02,(uint8_t)(h_num),(uint8_t)(l_num),(uint8_t)crc,0x54,0x43};

- platform: template
name: "Custom Judgment threshold exists"
```

```
id: custom_judgment_threshold_exists
min_value: 0
max_value: 250
step: 1
set_action:
  - uart.write: !lambda
      int crc = 0xBD + (int)x;
      return
{0x53,0x59,0x08,0x08,0x00,0x01,(uint8_t)x,(uint8_t)crc,0x54,0x43};

- platform: template
  name: "Custom Motion amplitude trigger threshold"
  id: custom_motion_amplitude_trigger_threshold
  min_value: 0
  max_value: 250
  step: 1
  set_action:
    - uart.write: !lambda
        int crc = 0xBE + (int)x;
        return
{0x53,0x59,0x08,0x09,0x00,0x01,(uint8_t)x,(uint8_t)crc,0x54,0x43};

- platform: template
  name: "Custom Mode Settings"
  id: custom_mode_settings
  icon: mdi:cog
  min_value: 0
  max_value: 250
  step: 1
  set_action:
    - uart.write: !lambda
        int crc = 0xBB + (int)x;
        return
{0x53,0x59,0x05,0x09,0x00,0x01,(uint8_t)x,(uint8_t)crc,0x54,0x43};

- platform: template
  name: "Custom Mode Settings End"
  id: custom_mode_setting_completed
  icon: mdi:cog
  min_value: 0
  max_value: 250
  step: 1
  set_action:
    - uart.write: !lambda
        int crc = 0xBC + (int)x;
        return
{0x53,0x59,0x05,0x0a,0x00,0x01,(uint8_t)x,(uint8_t)crc,0x54,0x43};

- platform: template
  name: "Custom Custom Mode Query"
  icon: mdi:cog
```

```
id: custom_mode_query
min_value: 0
max_value: 250
step: 1
set_action:
  - uart.write: !lambda
      int crc = 0x3B + (int)x;
      return
{0x53,0x59,0x05,0x89,0x00,0x01,(uint8_t)x,(uint8_t)crc,0x54,0x43};

- platform: template
  name: "Custom Motion trigger time"
  id: custom_motion_trigger_time
  icon: mdi:camera-timer
  unit_of_measurement: "ms"
  min_value: 0
  max_value: 4294967295
  step: 5000
  set_action:
    - uart.write: !lambda
        int crc = 0xC4 + (int)x;
        int h24_num = ((int)x>> 24) & 0xff;
        int h16_num = ((int)x>> 16) & 0xff;
        int h8_num = ((int)x>> 8) & 0xff;
        int l8_num = (int)x & 0xff;
        return
{0x53,0x59,0x08,0x0c,0x00,0x04,(uint8_t)h24_num,(uint8_t)h16_num,(uint8_t)h8_num,(uint8_t)l8_num,(uint8_t)crc,0x54,0x43};

- platform: template
  name: "Custom Movement to rest time"
  id: custom_movement_to_rest_time
  icon: mdi:camera-timer
  unit_of_measurement: "ms"
  min_value: 0
  max_value: 4294967295
  step: 5000
  set_action:
    - uart.write: !lambda
        int crc = 0xC5 + (int)x;
        int h24_num = ((int)x>> 24) & 0xff;
        int h16_num = ((int)x>> 16) & 0xff;
        int h8_num = ((int)x>> 8) & 0xff;
        int l8_num = (int)x & 0xff;
        return
{0x53,0x59,0x08,0x0d,0x00,0x04,(uint8_t)h24_num,(uint8_t)h16_num,(uint8_t)h8_num,(uint8_t)l8_num,(uint8_t)crc,0x54,0x43};

- platform: template
  name: "Custom Time of entering unmanned state"
```

```
id: custom_time_of_enter_unmanned
icon: mdi:camera-timer
unit_of_measurement: "ms"
min_value: 0
max_value: 4294967295
step: 5000
set_action:
  - uart.write: !lambda
      int crc = 0xC6 + (int)x;
      int h24_num = ((int)x>> 24) & 0xff;
      int h16_num = ((int)x>> 16) & 0xff;
      int h8_num = ((int)x>> 8) & 0xff;
      int l8_num = (int)x & 0xff;
      return
{0x53,0x59,0x08,0x0e,0x00,0x04,(uint8_t)h24_num,(uint8_t)h16_num,(uint8_t)h8_num,(uint8_t)l8_num,(uint8_t)crc,0x54,0x43};

text_sensor:
- platform: custom
  lambda: |-
    auto my_custom_sensor = new MyCustomTextSensor();
    App.register_component(my_custom_sensor);
    return {my_custom_sensor->Heartbeat};

text_sensors:
  - name: "Standard Heartbeat"
    icon: mdi:connection

- platform: template
  name: "Standard Product model"
  id: product_mode
  icon: mdi:information-outline
  on_raw_value:
    then:
      - logger.log: text_sensor on_raw_value

- platform: template
  name: "Standard Product ID"
  id: product_id
  icon: mdi:information-outline

- platform: template
  name: "Standard Hardware model"
  id: hardware_model
  icon: mdi:information-outline

- platform: template
  name: "Standard Firmware version"
  id: firmware_version
  icon: mdi:information-outline
```

```
- platform: template
  name: "Standard protocol type"
  id: protocol_type
  icon: mdi:information-outline

- platform: template
  name: "Standard moving direction"
  id: keep_away
  icon: mdi:walk

- platform: template
  name: "Standard Sports information"
  id: motion_status
  icon: mdi:human-greeting

- platform: template
  name: "Standard Presence information"
  id: someoneExists
  icon: "mdi:motion-sensor"

- platform: template
  name: "Custom Presence of detection"
  id: custom_presence_of_detection
  icon: mdi:signal-distance-variant

# - platform: template
#   name: "Custom Motion distance"
#   id: custom_motion_distance

# - platform: template
#   name: "Custom Static distance"
#   id: custom_static_distance

# - platform: template
#   name: "Custom Spatial static value"
#   id: custom_spatial_static_value

# - platform: template
#   name: "Custom Spatial motion value"
#   id: custom_spatial_motion_value

# - platform: template
#   name: "Custom Motion speed"
#   id: custom_motion_speed

button:
- platform: template
  name: "Standard reset"
  id: "reset"
  icon: mdi:reload
```

```

    on_press:
      then:
        - logger.log: Button Pressed
        - uart.write:
[0x53,0x59,0x01,0x02,0x00,0x01,0x0F,0xBF,0x54,0x43]

switch:
  - platform: template
    id: output_info_switch
    name: "Custom Infor output switch"
    icon: mdi:electric-switch
    assumed_state: true
    turn_on_action:
      - uart.write: [0x53,0x59,0x08,0x00,0x00,0x01,0x01,0xB6,0x54,0x43]
      - delay: 1s
      - lambda: !lambda |-
          id(product_mode).publish_state("");
          id(product_id).publish_state("");
          id(hardware_model).publish_state("");
          id(firmware_version).publish_state("");
          id(protocol_type).publish_state("");
    turn_off_action:
      - uart.write: [0x53,0x59,0x08,0x00,0x00,0x01,0x00,0xB5,0x54,0x43]

sensor:
  - platform: custom
    lambda: |-
      auto my_custom_sensor = new UartReadLineSensor(id(uart_bus));
      App.register_component(my_custom_sensor);
      return {
        my_custom_sensor->movementSigns,
        my_custom_sensor->initied,
      };
  sensors:
    - name: "Standard body movement"
      id: movementSigns
      icon: "mdi:human-greeting-variant"
      device_class: "temperature"
      state_class: "measurement"

    - name: "Standard initied"
      id: initied
      icon: mdi:all-inclusive

  - platform: template
    name: "Custom Motion distance"
    id: custom_motion_distance
    icon: mdi:signal-distance-variant
    on_value:
      then:
        # - logger.log: Custom Motion distance on_value

```

```
- logger.log:
    format: "Custom Motion distance on_value : %d"
    args: ["x"]
on_raw_value:
    then:
        - logger.log:
            format: "Custom Motion distance on_raw_value : %d"
            args: ["x"]

- platform: template
  name: "Custom Static distance"
  id: custom_static_distance
  icon: mdi:signal-distance-variant

- platform: template
  name: "Custom Spatial static value"
  id: custom_spatial_static_value
  icon: mdi:counter

- platform: template
  name: "Custom Spatial motion value"
  id: custom_spatial_motion_value
  icon: mdi:counter

- platform: template
  name: "Custom Motion speed"
  id: custom_motion_speed
  icon: mdi:run-fast
```

From:
<https://gatonero.duckdns.org/digitaless/> - **Digitaless**

Permanent link:
https://gatonero.duckdns.org/digitaless/digitaless:hardware:sensoren:human_static_presence_module_lite_mr24hpc1

Last update: **25.05.2025**

