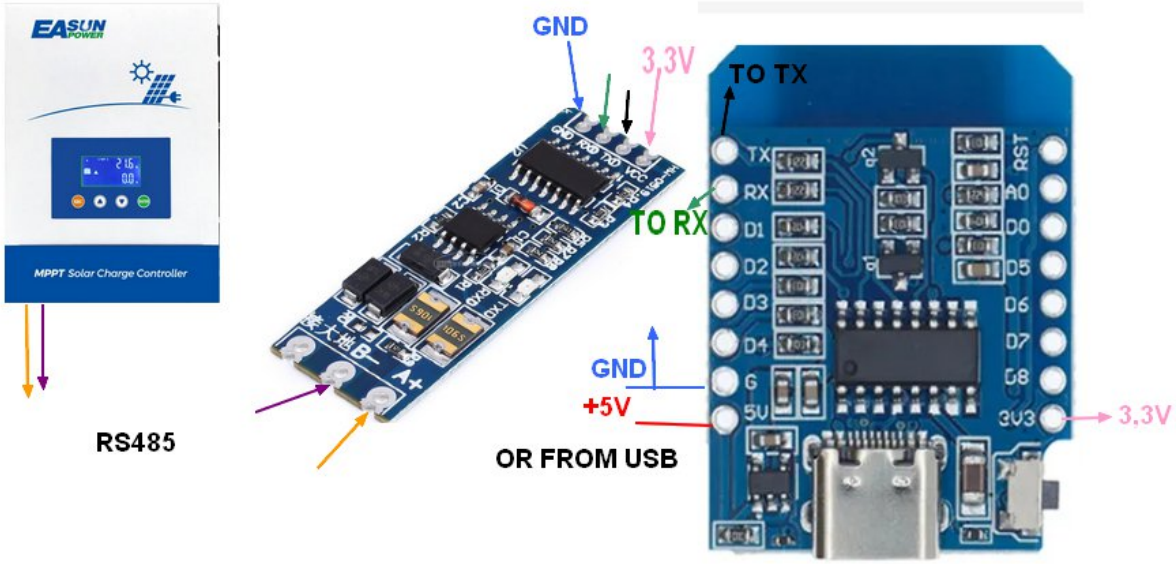
EASUN2 RS485 –RS485 TTL (3,3V) - WEMOS D1 MINI –HA




```
substitutions:
  name: "easun-mppt2"
  device_description: "Monitor EPEVER2"
  updates: 10s

esphome:
  name: esphome-web-ccd0d7
  friendly_name: ESPHome EASUN2

esp8266:
  board: esp01_1m

# Enable logging
logger:

##  level: INFO
  baud_rate: 0
```

```
# Enable Home Assistant API
api:

  reboot_timeout: 0s

ota:
web_server:
  port: 80

wifi:

  networks:
  - ssid: xxxxxx
    password: xxxxxx

  - ssid: xxxx
    password: xxxxx

  -  manual_ip:
      static_ip: X.X.X.X
      gateway: X.X.X.X
      subnet: X.X.X.X

  power_save_mode: none

  # Enable fallback hotspot (captive portal) in case wifi connection fails
  ap:
    ssid: "Esphome-Web-ccd0d7"
    password: "xxxxxx"
```

```
captive_portal:

time:
  - platform: sntp
    id: sntp_time
    servers: "X.X.X.X"

uart:
  - id: modbus_serial_port
    tx_pin: GPIO1
    rx_pin: GPIO3
    baud_rate: 9600
    stop_bits: 1



modbus:
  #flow_control_pin: GPIO2
  #send_wait_time: 200ms
  id: mod_bus_easun2
  uart_id: modbus_serial_port
  send_wait_time:
    milliseconds: 200

modbus_controller:
  - id: easun2
    address: 0x1
    modbus_id: mod_bus_easun2
    command_throttle: 200ms
    setup_priority: -10
```

```
    update_interval: ${updates}



sensor:
  - platform: modbus_controller
    modbus_controller_id: easun2
    id: easun2_battery_voltage
    name: "easun2_Battery voltage"
    custom_command: [0x1,0x3,0x01,0x02,0x00,0x01]
    unit_of_measurement: "V"
    value_type: U_WORD
    accuracy_decimals: 2
    filters:
      - multiply: 0.1

  - platform: modbus_controller
    modbus_controller_id: easun2
    id: easun2_charging_current
    name: "easun2_Charging current"
    custom_command: [0x1,0x3,0x01,0x03,0x00,0x01]
    unit_of_measurement: "A"
    value_type: U_WORD
    accuracy_decimals: 2
    filters:
      - multiply: 0.01

  - platform: modbus_controller
    modbus_controller_id: easun2
    id: easun2_charging_power
    name: "easun2_Charging power"
```

```
  custom_command: [0x1,0x3,0x01,0x04,0x00,0x01]
  unit_of_measurement: "W"
  value_type: U_WORD


- platform: modbus_controller
  modbus_controller_id: easun2
  id: easun2_actual_temperature
  name: "easun2_Actual temperature"
  custom_command: [0x1,0x3,0x01,0x05,0x00,0x01]
  unit_of_measurement: "°C"
  value_type: U_WORD
  bitmask: 0xFF00

- platform: modbus_controller
  modbus_controller_id: easun2
  id: easun2_pv_input_voltage
  name: "easun2_PV array input voltage"
  custom_command: [0x1,0x3,0x01,0x09,0x00,0x01]
  unit_of_measurement: "V"
  value_type: U_WORD
  accuracy_decimals: 2
  filters:
    - multiply: 0.1

- platform: modbus_controller
  modbus_controller_id: easun2
  id: easun2_max_day_power
  name: "easun2_Maximum day power"
  custom_command: [0x1,0x3,0x01,0x0A,0x00,0x01]
```

```
  unit_of_measurement: "W"
  value_type: U_WORD

- platform: modbus_controller
  modbus_controller_id: easun2
  id: easun2_power_day_generation
  name: "easun2_Power day generation"
  custom_command: [0x1,0x3,0x01,0x0B,0x00,0x01]
  unit_of_measurement: "Wh"
  state_class: "total_increasing"
  device_class: "energy"
  value_type: U_WORD
  accuracy_decimals: 2

- platform: modbus_controller
  modbus_controller_id: easun2
  id: easun2_power_day_consumption
  name: "easun2_Power day consumption"
  custom_command: [0x1,0x3,0x01,0x0C,0x00,0x01]
  unit_of_measurement: "Wh"
  state_class: "total_increasing"
  device_class: "energy"
  value_type: U_WORD
  accuracy_decimals: 2

- platform: modbus_controller
  modbus_controller_id: easun2
  id: easun2_charging_status_number
  name: "easun2_Charging status number"
  custom_command: [0x1,0x3,0x01,0x0D,0x00,0x01]
```

```
value_type: U_WORD
bitmask: 0x00FF
on_value:
  - if:
      condition:
        lambda: 'return id(easun2_charging_status_number).state == 0;'
      then:
        - text_sensor.template.publish:
            id: easun2_charging_status_text
            state: "No charge"
  - if:
      condition:
        lambda: 'return id(easun2_charging_status_number).state == 1;'
      then:
        - text_sensor.template.publish:
            id: easun2_charging_status_text
            state: "Open charge mode"
  - if:
      condition:
        lambda: 'return id(easun2_charging_status_number).state == 2;'
      then:
        - text_sensor.template.publish:
            id: easun2_charging_status_text
            state: "MPPT charge mode"
  - if:
      condition:
        lambda: 'return id(easun2_charging_status_number).state == 3;'
      then:
        - text_sensor.template.publish:
            id: easun2_charging_status_text
```

```
            state: "Equalizing charge mode"
    - if:
        condition:
          lambda: 'return id(easun2_charging_status_number).state == 4;'
        then:
          - text_sensor.template.publish:
              id: easun2_charging_status_text
              state: "Boost charge mode"
    - if:
        condition:
          lambda: 'return id(easun2_charging_status_number).state == 5;'
        then:
          - text_sensor.template.publish:
              id: easun2_charging_status_text
              state: "Float charge mode"
    - if:
        condition:
          lambda: 'return id(easun2_charging_status_number).state == 6;'
        then:
          - text_sensor.template.publish:
              id: easun2_charging_status_text
              state: "Current limit!!!"

  - platform: modbus_controller
    modbus_controller_id: easun2
    name: "easun2_Alarm information"
    id: easun2_alarm_information
    custom_command: [0x1,0x3,0x01,0x0E,0x00,0x01]
    value_type: U_WORD
```

```
  - platform: modbus_controller
    modbus_controller_id: easun2
    name: "easun2_Total running days"
    id: easun2_total_running_days
    custom_command: [0x1,0x3,0x01,0x0F,0x00,0x01]
    value_type: U_WORD

  - platform: modbus_controller
    modbus_controller_id: easun2
    id: easun2_cumulative_power_generation
    name: "easun2_Cumulative power generation"
    custom_command: [0x1,0x3,0x01,0x10,0x00,0x02]
    unit_of_measurement: "kWh"
    state_class: "total_increasing"
    device_class: "energy"
    value_type: U_DWORD

  - platform: modbus_controller
    modbus_controller_id: easun2
    id: easun2_cumulative_power_consumption
    name: "easun2_Cumulative power consumption"
    custom_command: [0x1,0x3,0x01,0x12,0x00,0x02]
    unit_of_measurement: "Wh"
    state_class: "total_increasing"
    device_class: "energy"
    value_type: U_DWORD

binary_sensor:
  - platform: template
    id: easun2_alarm_batt_h_odrv
```

```
  name: "easun2_Alarm Batt Vol higher than ODRV"
  entity_category: diagnostic
  lambda: |-
    uint16_t alarm = id(easun2_alarm_information).state;
    if (bitRead(alarm,0)) {
      return true;
    } else {
      return false;
    }
- platform: template
  id: easun2_alarm_batt_s_odrv
  name: "easun2_Alarm Batt Vol smaller than ODRV"
  entity_category: diagnostic
  lambda: |-
    uint16_t alarm = id(easun2_alarm_information).state;
    if (bitRead(alarm,1)) {
      return true;
    } else {
      return false;
    }
- platform: template
  id: easun2_alarm_batt_h_arv
  name: "easun2_Alarm Batt higher than ARV"
  entity_category: diagnostic
  lambda: |-
    uint16_t alarm = id(easun2_alarm_information).state;
    if (bitRead(alarm,2)) {
      return true;
    } else {
      return false;
```

```
    }
  - platform: template
    id: easun2_alarm_long_press_load_btn
    name: "easun2_Alarm Long press load btn"
    entity_category: diagnostic
    lambda: |-
      uint16_t alarm = id(easun2_alarm_information).state;
      if (bitRead(alarm,3)) {
        return true;
      } else {
        return false;
      }
  - platform: template
    id: easun2_alarm_lower_load_pwr
    name: "easun2_Alarm Lover load power"
    entity_category: diagnostic
    lambda: |-
      uint16_t alarm = id(easun2_alarm_information).state;
      if (bitRead(alarm,4)) {
        return true;
      } else {
        return false;
      }
  - platform: template
    id: easun2_alarm_ctrl_tmp_lo_to_ot_ret_val
    name: "easun2_Alarm Controller temp lowed to OT ret val"
    entity_category: diagnostic
    lambda: |-
      uint16_t alarm = id(easun2_alarm_information).state;
      if (bitRead(alarm,5)) {
```

```
      return true;
    } else {
      return false;
    }
  - platform: template
    id: easun2_alarm_amb_tmp_lo_to_ot_ret_val
    name: "easun2_Alarm Ambient temp lowed to OT ret val"
    entity_category: diagnostic
    lambda: |-
      uint16_t alarm = id(easun2_alarm_information).state;
      if (bitRead(alarm,6)) {
        return true;
      } else {
        return false;
      }
  - platform: template
    id: easun2_alarm_pwr_lo_to_ret_val
    name: "easun2_Alarm Power lowed to return value"
    entity_category: diagnostic
    lambda: |-
      uint16_t alarm = id(easun2_alarm_information).state;
      if (bitRead(alarm,7)) {
        return true;
      } else {
        return false;
      }
  - platform: template
    id: easun2_alarm_timing
    name: "easun2_Alarm Timing"
    entity_category: diagnostic
```

```
  lambda: |-
    uint16_t alarm = id(easun2_alarm_information).state;
    if (bitRead(alarm,8)) {
      return true;
    } else {
      return false;
    }
- platform: template
  id: easun2_alarm_batt_lo_to_ret_val
  name: "easun2_Alarm Batt lowed to OV return value"
  entity_category: diagnostic
  lambda: |-
    uint16_t alarm = id(easun2_alarm_information).state;
    if (bitRead(alarm,9)) {
      return true;
    } else {
      return false;
    }
- platform: template
  id: easun2_alarm_pv_rev_pol
  name: "easun2_Alarm PV reversed polarity"
  entity_category: diagnostic
  lambda: |-
    uint16_t alarm = id(easun2_alarm_information).state;
    if (bitRead(alarm,12)) {
      return true;
    } else {
      return false;
    }
- platform: template
```

```
    id: easun2_alarm_batt_rev_pol
    name: "easun2_Alarm Batt reversed polarity"
    entity_category: diagnostic
    lambda: |-
      uint16_t alarm = id(easun2_alarm_information).state;
      if (bitRead(alarm,13)) {
        return true;
      } else {
        return false;
      }

text_sensor:
  - platform: template
    name: "easun2_Charging status text"
    id: easun2_charging_status_text

button:
  - platform: restart
    name: "easun2_ESP_Restart"
```