# Measure distance using Ultrasonic sensor US-100

US-100 is one of popular distance sensor used in many project. It has ranging distance upto 450 cm with 1mm resolution in less then 15 degree of view angle. It also has temperature sensor to compensate the result. It provides 2 interface, one-pulse and UART.

#sensor #distance #pulse #uart

---

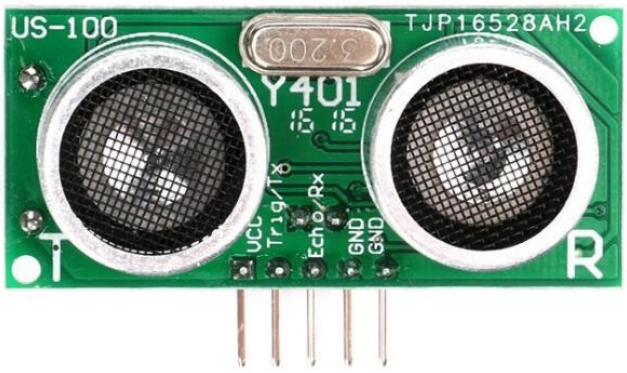
Last update: May 3, 2021

## Table of Content

## 1. US-100 Ultrasonic sensor

The US-100 Ultrasonic sensor is very similar to the popular HC-SR04, and even looks the same, but has a few extra tricks:

- Can run from 3V to 5V, so don't need any logic level shifters or dividers.
- Can use in "Pulse" mode (like on HC-SR04) or in "Serial UART" mode.
- Range is about 2cm to 450cm away, but 10cm-250cm will get the best results



*Ultrasonic sensor US-100*

- When the jumper is in place, use an 9600 baud UART to communicate with the sensor:
- send `0x55` and read back two bytes (16 bit value) that is mm distance
- send `0x50` to read the temperature in degrees C, in offset of -45
- When the jumper on the back is removed, it acts like an HC-SR04 with a trigger and echo pin
- The width of echo pulse is the time it takes for the ultrasonic sound to travel from the sensor to the object and back.

**Debug on UART1 using Redirection**

For more convenient, this project use UART Redirection technique to use the UART1 as the debug terminal, and use standard `printf()` function to output messages.

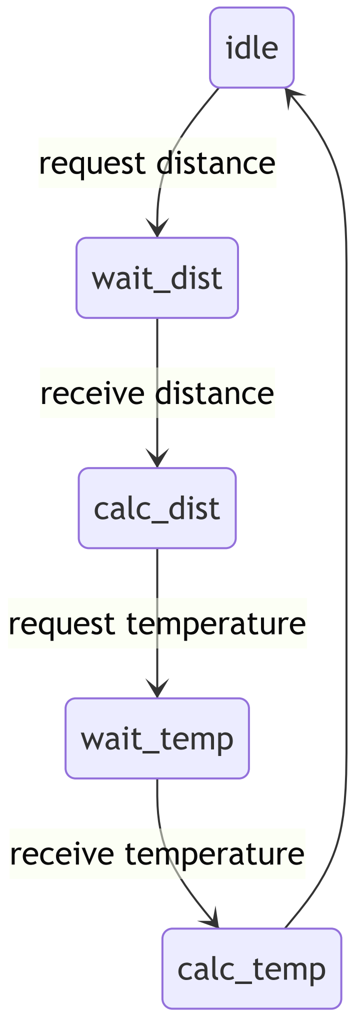
## 2. UART mode

Use UART2 to communicate with US-100.

| MCU Pin | US-100 Pin |
|---|---|
| PA2 (UART2TX) | Trigger/TX |
| PA3 (UART2RX) | Echo/RX |

### State

The system will go around 5 states:



### Code

Create variable to hold the state, trial counter, commands, and returned value:

```
enum {
  IDLE,
  WAIT_DIST,
  CALC_DIST,
  WAIT_TEMP,
  CALC_TEMP
};
char state = IDLE;
char try = 0;
uint16_t value = 0;
uint8_t cmd_dist[] = {0x55};
uint8_t cmd_temp[] = {0x50};
uint8_t buffer[2] = {0};
```

Then, in the main loop, process each state, note to use interrupt mode to receive data:

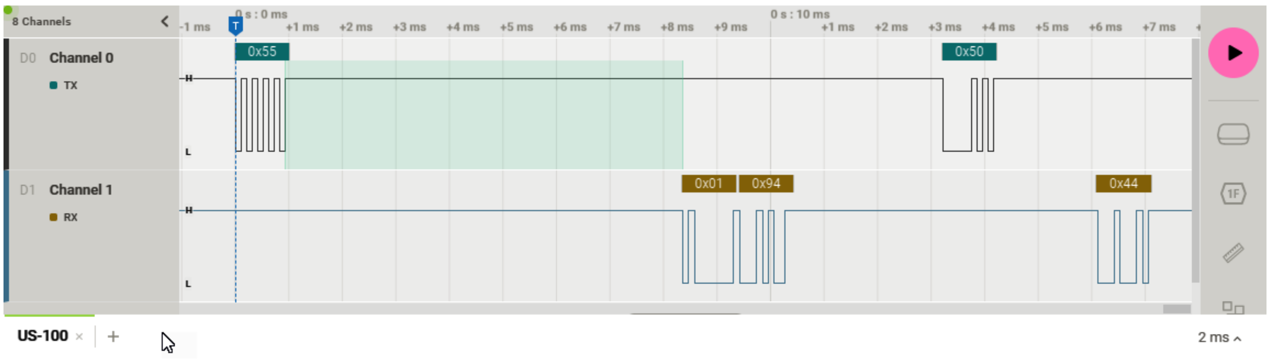
```
int main(void) {
    Set_Redirect_UART_Port(TERMINAL);

    ... other setup ...

    while(1) {
        if (state == IDLE) {
            // send request to measure distance
            printf("D?\n\r");
            HAL_UART_Transmit(US_100, cmd_dist, 1, HAL_MAX_DELAY);
            HAL_UART_Receive_IT(US_100, buffer, 2);
            // change state
            state = WAIT_DIST;
            try = 0;
        } else if (state == CALC_DIST) {
            // calculate distance
            value = (buffer[0] << 8) + buffer[1];
            printf("D = %d mm\n\r", value);
            // send request to get temperature
            printf("T?\n\r");
            HAL_UART_Transmit(US_100, cmd_temp, 1, HAL_MAX_DELAY);
            HAL_UART_Receive_IT(US_100, buffer, 1);
            // change state
            state = WAIT_TEMP;
            try = 0;
        } else if (state == CALC_TEMP){
            // calculate temperature
            value = buffer[0] - 45;
            printf("T = %d\n\r", value);
            // change state
            state = IDLE;
            try = 0;
        }

        HAL_Delay(500);

        // retry after 5 seconds
        if(++try >= 10) {
            printf("Re-try\n\r");
            state = IDLE;
        }
    }
}
```

Finally, handle the interrupt callback by checking the state and set new state for the main loop:

```
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
  if (huart == US_100) {
    if (state == WAIT_DIST) {
      state = CALC_DIST;
    } else if (state == WAIT_TEMP) {
      state = CALC_TEMP;
    }
  }
}
```

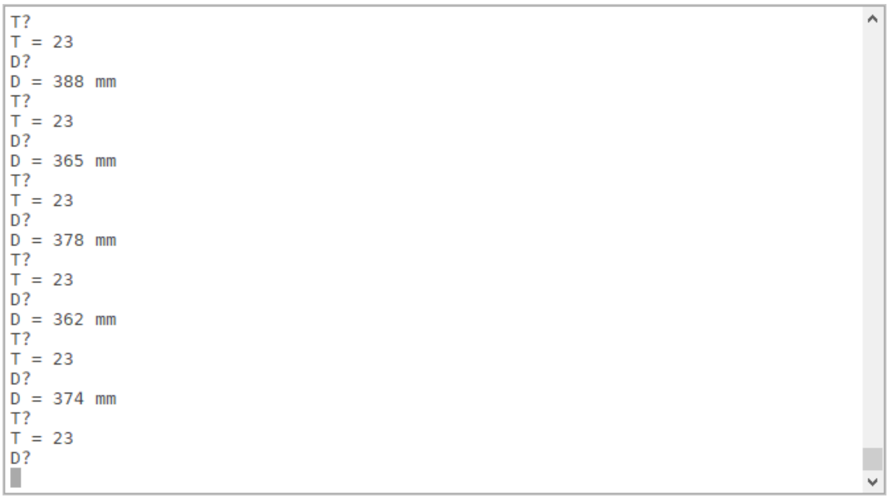
Compile and run on the board, use an logic analyser to check how fast the US-100 can response for each command:

- Distance response time: < 10 ms
- Temperature response time: < 5 ms



*Output of US-100*

And on the debug terminal, the distance and temperature are printed in decimal value:



```
T?
T = 23
D?
D = 388 mm
T?
T = 23
D?
D = 365 mm
T?
T = 23
D?
D = 378 mm
T?
T = 23
D?
D = 362 mm
T?
T = 23
D?
D = 374 mm
T?
T = 23
D?
```

*Print output on terminal*

## 3. Pulse mode

*update soon*