Ultrasound Distance Measurement
Outline

- How It Works
- KS103 Ultrasound Module
- Module Connection
- Software Development
- Tasks
How It Works
Working Schematic Diagram
Working Schematic Diagram

Trigger ➔ Obstacle

Echo ➔ Obstacle

Distance $L = \frac{340 \text{ m/s} \times \Delta t \text{ s}}{2}$
KS 103 Ultrasound Module
KS 103 Ultrasound Module

Operate Voltage: 3.0-5.5V;
Operate Current: 1.6-2.7 mA typical , 10.6mA max.
Standby Current: 500uA, max (I2C mode)
Main features I

- Resolution of 1mm, high precision, the accuracy of ranging is 1mm - 10mm (within 4.5m)
- Detecting frequency up to 500Hz or more
- Unique filtering noise reduction technology, can still work under noisy power supply
- Temperature Compensation
- Two digital communication: I2C/Uart
Module Connection
Pin Description

- VCC: Power Pin
- SDA/TX: data pin in I2C bus/TX pin in uart bus
- SCL/RX: clock pin in I2C bus/RX pin in uart bus
- GND: power ground pin
- Mode: Select the communication mode.
  - High Level (VCC): I2C
  - Low Level (GND): Uart
I2C connection

- Under I2C bus, 20 I2C address can be changed
- 20 modules can be connected when using I2C module
I2C connection
Uart connects (used in this course)

- 2 modules can be connected under uart mode
- We use uart mode in this course. For simplicity, each uart only connect 1 KS103.
Software Development
Control KS103 module

Send Address 0xe8

Dealy 20-100us

Send register address 0x2

Dealy 20-100us

Send command

Receive high 8 bits of result

Receive low 8 bits of result
Uart API

- The hardware image has 4 uart IPs
- Software interface is provided in uart0.c and uart0.h

```c
void Uart_init(long Address): Initialize the uart
    Address: uart IP address which can be found in system.h

void Uart_send_byte(long Address, unsigned char data): send data via uart
    Address: uart IP address which can be found in system.h
    data: the date you want to send

unsigned char Uart_receive_byte(long Address);
    Address: uart IP address which can be found in system.h
    return : the date you received
```
## Command List

<table>
<thead>
<tr>
<th>Register</th>
<th>Command (8 bits)</th>
<th>Range of Return value (16 bits)</th>
<th>Return</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x2</td>
<td>0xa0</td>
<td>0-0x3ff</td>
<td>Return light intensity</td>
</tr>
<tr>
<td>0x2</td>
<td>0xb0</td>
<td>0x0a-0x1450</td>
<td>Return distance in mm (range: 0-5m)</td>
</tr>
<tr>
<td>0x2</td>
<td>0xb8</td>
<td>0x14-0x2c10</td>
<td>Return distance in mm (range: 0-11m)</td>
</tr>
<tr>
<td>0x2</td>
<td>Number X from 0x01 to 0x2f</td>
<td></td>
<td>Return transmission time in us (range: 0-X*100 mm)</td>
</tr>
</tbody>
</table>
Example: measure the distance

- Send Address 0xe8
  - Uart_send_byte(Address, 0xe8)
- Dealy 20-100us
  - For loop for delay
- Send register address 0x2
  - Uart_send_byte(Address, 0x02)
- Dealy 20-100us
  - For loop for delay
- Send command
  - Uart_send_byte(Address, 0xb8)
  - (0-11m range settings)
- Receive high 8 bits of result
  - high=Uart_receive_byte(Address)
- Receive low 8 bits of result
  - low=Uart_receive_byte(Address)
Hardware

- .sof is hardware image file. Download it by using Quartus
- .sopcinfo is used for BSP set-up in Nios-eclipse
- .QSF indicates the pin-map information (tell you how you how to connect pins to outside component, e.g., ultrasound sensors and H-bridge)
- Connect mode pin in ultrasound module to GND to select UART communication mode.
Find correct Pins (important)

#reuse above pins for New Ultrasound 1

#set_location_assignment PIN_T14 -to GPIO_1[2]
#set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to GPIO_1[2]
#set_location_assignment PIN_T13 -to GPIO_1[3]
#set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to GPIO_1[3]

set_location_assignment PIN_T14 -to New_Ultrasound1_rx
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to New_Ultrasound1_rx

set_location_assignment PIN_T13 -to New_Ultrasound1_tx
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to New_Ultrasound1_tx

Physical Pins On Nano

GPIO_1[2] -> GPIO12
GPIO_1[3] -> GPIO13
Questions
Tasks and evaluation scheme

- Task 1: Write code to read light intensity and print it out
- Task 2: use two key to trigger measurement under different range settings. One key is used to trigger measurement under 5m range setting and another is for 11m range setting. Print the results out and tell the difference.
- Task 3: Write code to measure the distances of one obstacle under 3m range setting.
- (Task1: 30%, Task2: 40%, Task3 :40%)