

# Ultrasound Distance Measurement



# Outline

- How It Works
- KS103 Ultrasound Module
- Module Connection
- Software Development
- Tasks





# How It Works

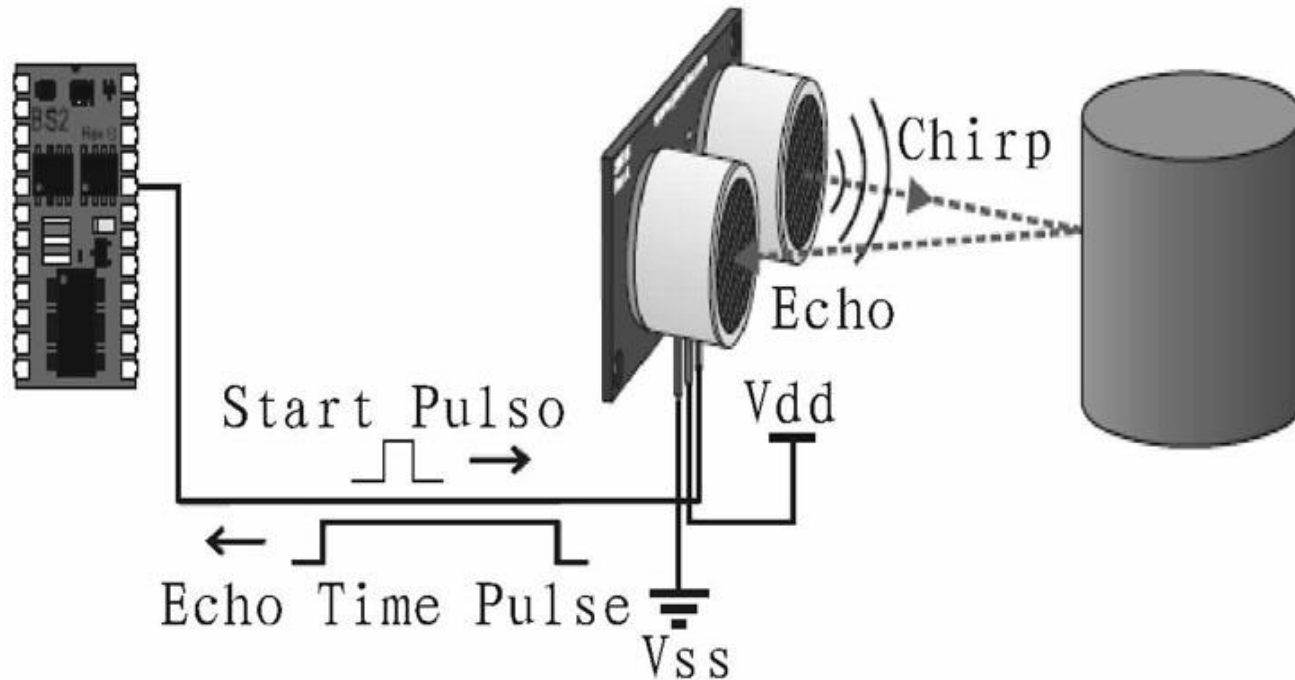


11/7/2016

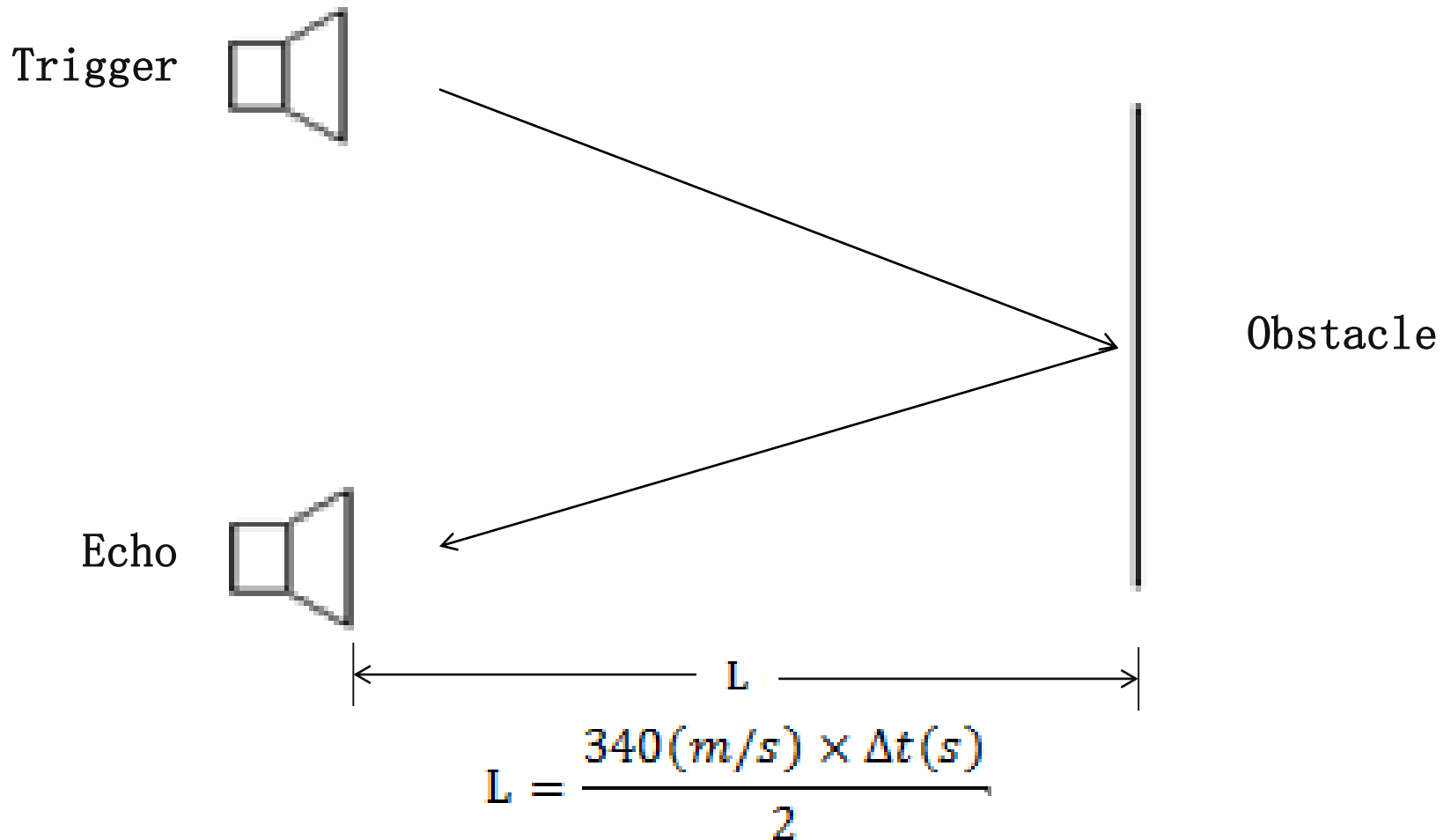
3



# Working Schematic Diagram



# Working Schematic Diagram



# KS 103 Ultrasound Module



11/7/2016

6



# 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



11/7/2016

9



# 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

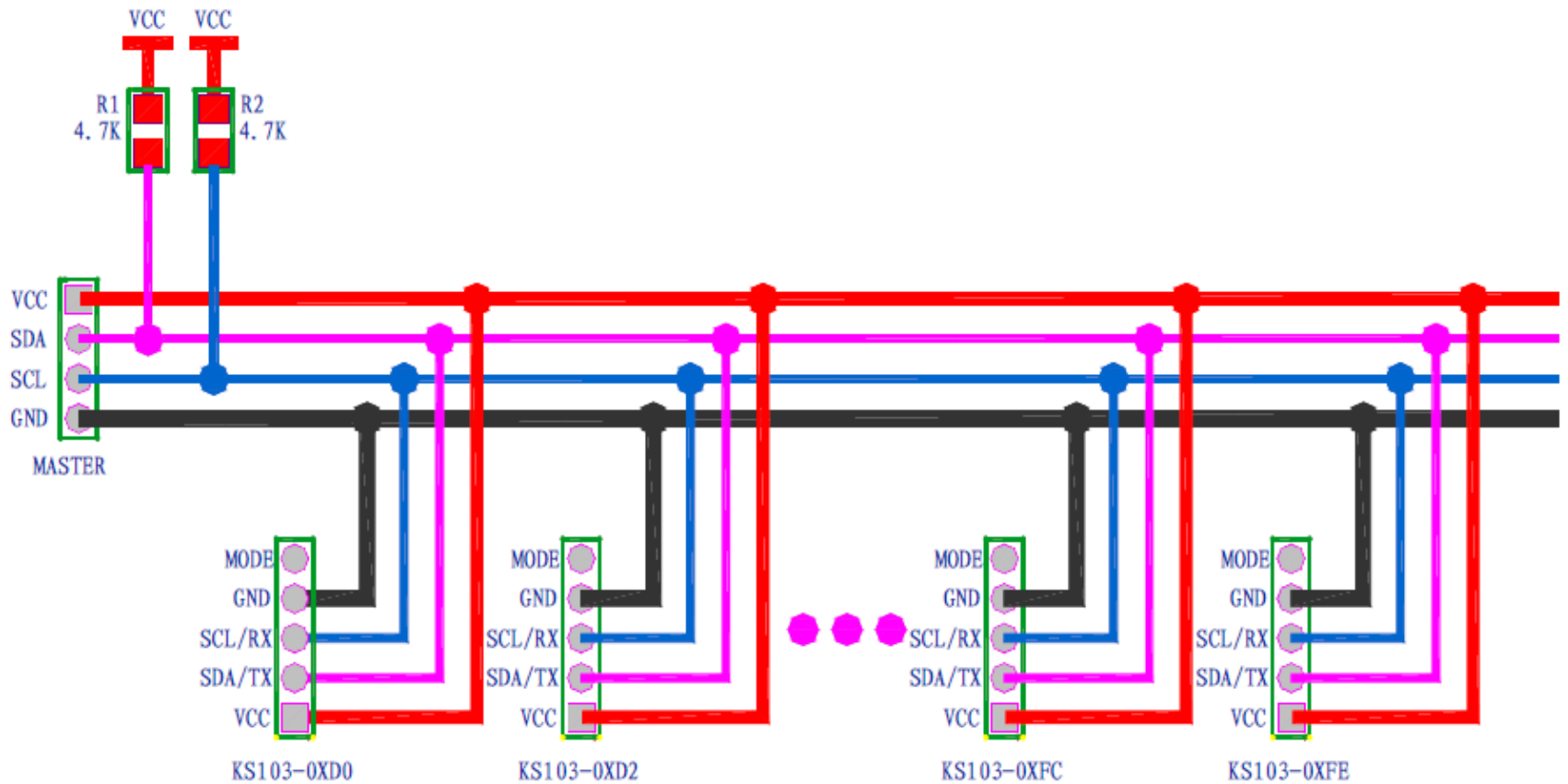


# I2C connection

- Under I2C bus, 20 I2C address can be changed
- 20 modules can be connected when using I2C module

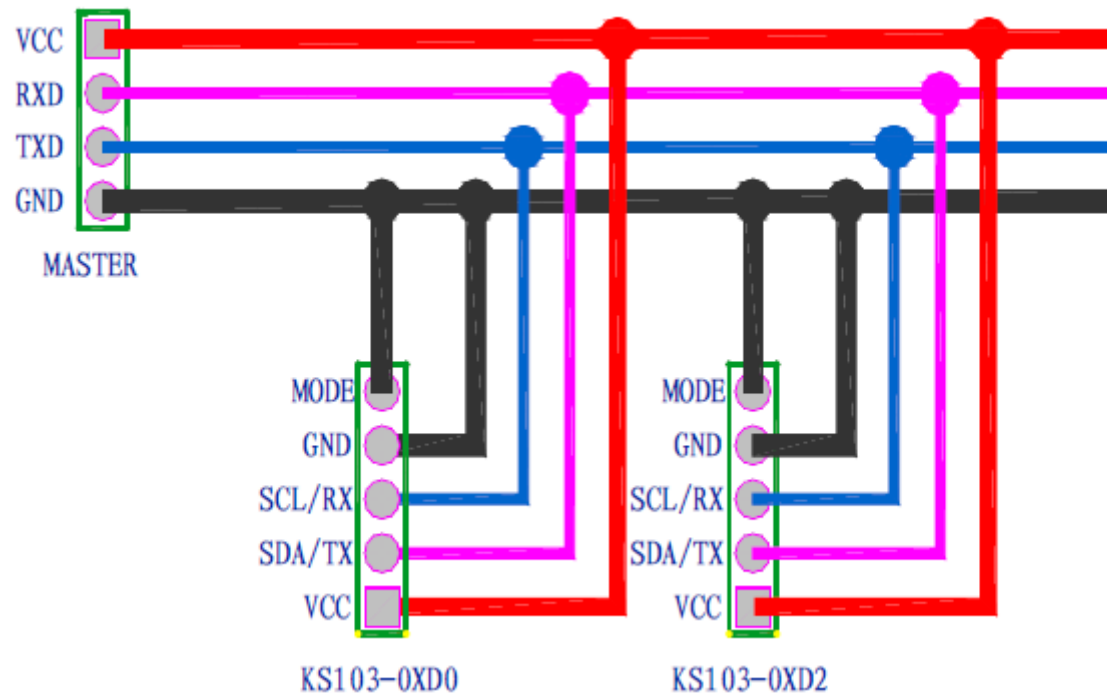


# I2C connection



# 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

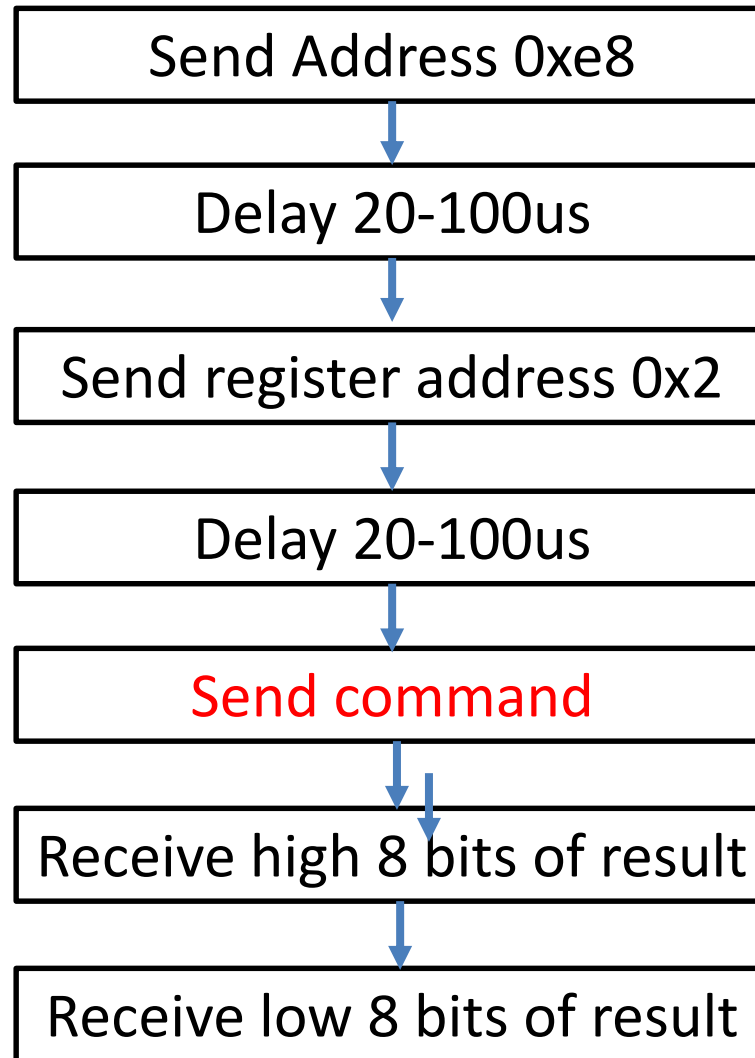


11/7/2016

14



# Control KS103 module





# Uart API

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

`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 data you want to send

`unsigned char Uart_receive_byte(long Address);`

**Address:** uart IP address which can be found in `system.h`

**return :** the data you received

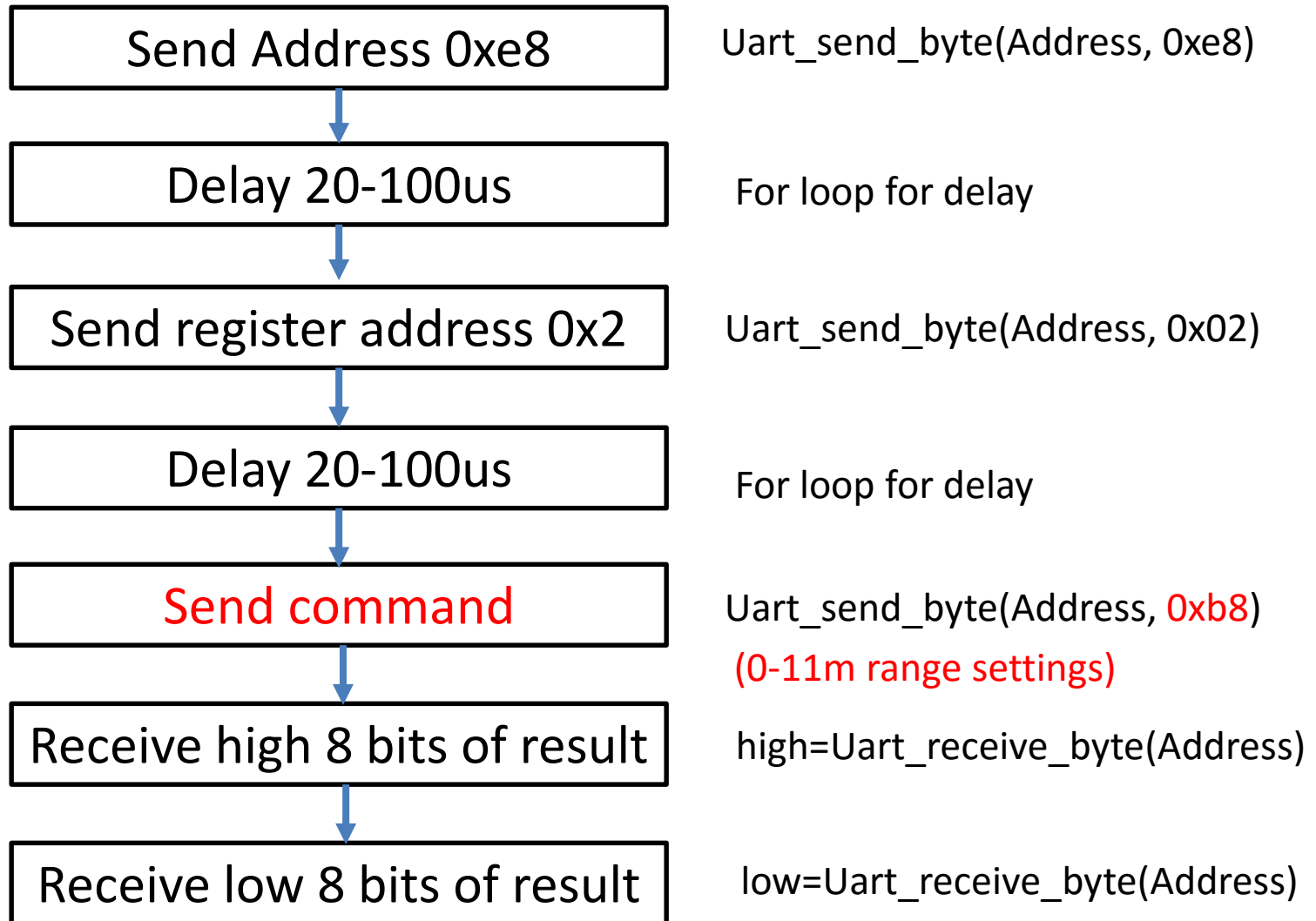


# Command List

Register	Command ( 8 bits)	Range of Return value (16 bits)	Return
0x2	0xa0	0-0x3ff	Return light intensity
0x2	0xb0	0x0a-0x1450	Return distance in mm (range: 0-5m)
0x2	0xb8	0x14-0x2c10	Return distance in mm (range: 0-11m)
0x2	Number X from 0x01 to 0x2f		Return transmission time in us (range: 0- X*100 mm)



# Example: measure the distance



# 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.

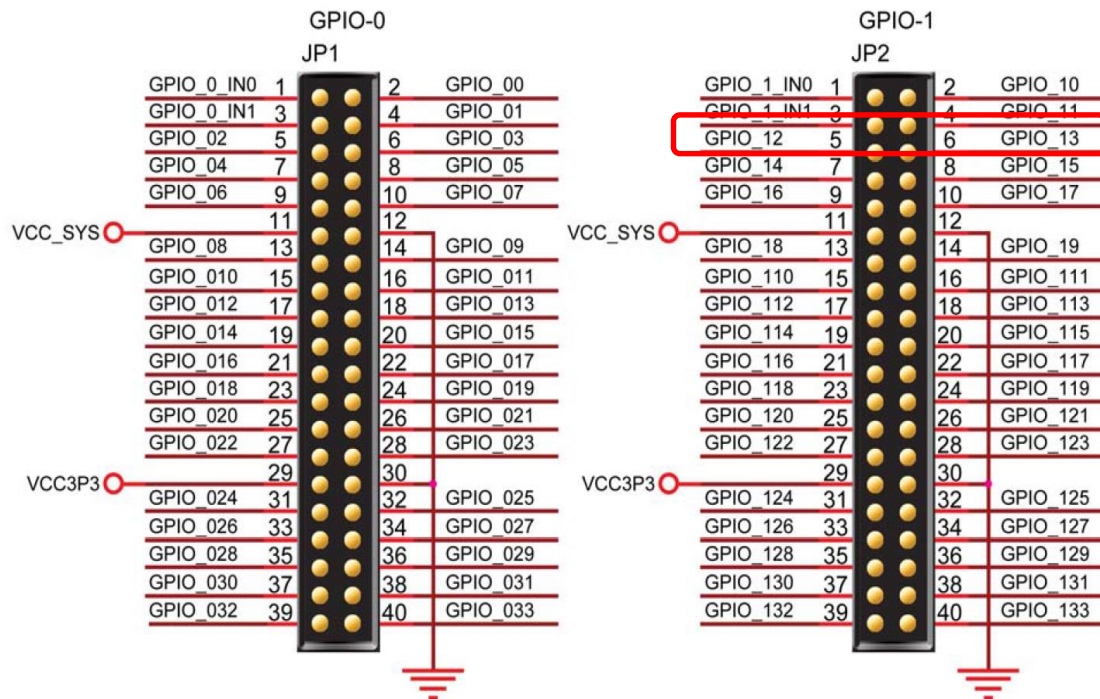


# Find correct Pins (important)

```
#-----  
#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
```

**GPIO\_1[2] -> GPIO12**  
**GPIO\_1[3] -> GPIO13**

Physical Pins  
On Nano



# 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.

