

Customized ECU and Software Interface



Outline

- Customized Nios processor for lego Car
- Software Interface for Sensors
- Communications between Imx6 and Nano



Customized Nios Processor for Lego Car



11/2/2015

3



Hardware Features

- Nios 32-bit processor (50Mhz)
- 32MByte SDRAM
- EPCS flash
- 32-bit timer (used for RTOS)
- Jtag-uart (for Debug)
- 8-Channel ADC
- 6 old version ultrasound IP (**Not used**)
- 8 PWM (16 channels)
- 4 new version ultrasound IP (**used for this course**)
- 2 encoder IP
- LED, key, GPIO



How to use this hardware

- Download New hardware image file from the course webpage
- .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)



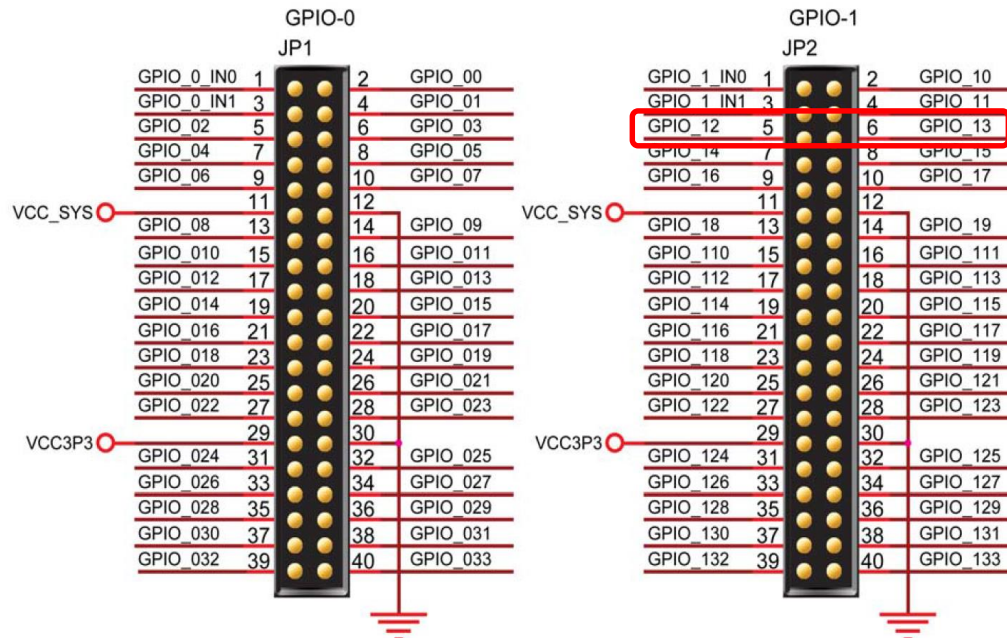
Find correct Pins (important)

```

#-----
#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



Software Interface



11/2/2015

7



Sensors and actuator Interface

- Ultrasound sensor interface
uart0.c, uart0.h,
new_ultrasound.c,new_ultrasound.h
- Communication interface between Imx6 and Nano
uartcom.c, uartcom.h
(115200 baud rate, 8 data bits, 1 stop bit, Non-parity)
- Encoder
please refer to the previous slides
- H-bridge interface
please refer to the previous slides



APIs for Ultrasound

- `void Ultrasound_init(long Address)`

Augments:

Address: used to indicate different ultrasound interface

Find in `system.h`

- `unsigned int Ultrasound_read(long Address, unsigned char settings)`

Argument:

Address: used to indicate different ultrasound interface

Settings : set the maximum range of measurement (5m or 11m)



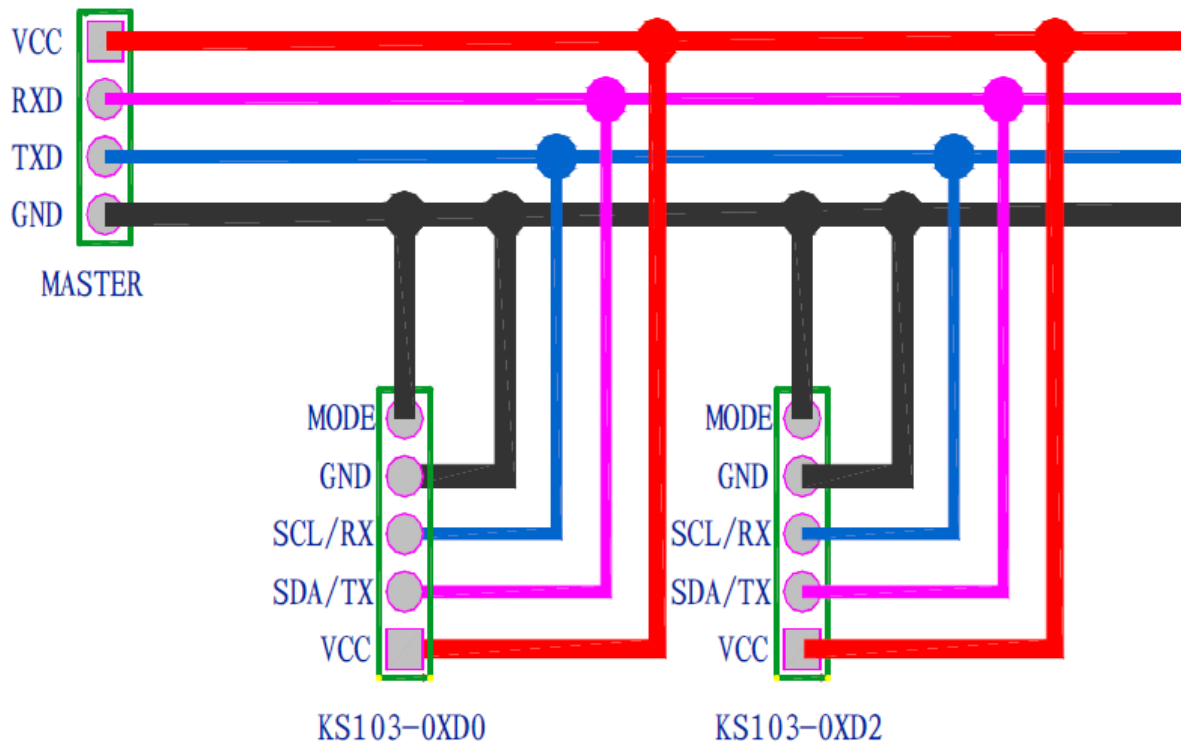
Examples Code -Ultrasound

```
#include "system.h"
#include "io.h"
#include "stdio.h"
#include "uart0.h"
#include "new_ultrasound.h"
int main()
{
    unsigned int res;
    unsigned int i;
    Ultrasound_init(NEW_ULTRASOUND1_BASE);
    while(1)
    {
        res=Ultrasound_read(NEW_ULTRASOUND1_BASE,0);
        printf("The distance is %f m.\n",res*340/1000000.0);
        for(i=0;i<100000;i++);
    }
    return 0;
}
```



Connections

- Mode Pin is connected to GND to select



Communication between imx6 and nano



Simple Communication Protocol (open)

- Master-Slave Communication

imx6 (master) send request to nano (slave), while nano (slave) response the request and return data to imx6 (master) .

- Synchronization Principle

Use shake hands message to Synchronize master and slave

- Master Message Format

Message header (1st byte): 0xaa

Sensor type (2nd byte): read or write (1bit), sensor type(3 bits), sensor ID(4 bits)

write data (3rd and 4th bytes): data for write command (for H-bridge)

Message Ender: (5th byte): 0x55

- Slave Message Format (shake hands message)

Message header: Two byte Data: Message Ender



APIs

- `unsigned char message_frame[InFrameByteNum];`
Received message frame from imx6 (master)
- `void Uartcom_init();`
Configuration the interrupt and uart
- `Uartcom_send_frame(unsigned char *data, unsigned char size)`
send the shake hand message



Connections

- Imx6 RS232 port <-> **Max232 converters** <-> Nano
- **Note: there are two 232 port in Imx6 board. Do not use the debug port.**



MAX232 converter (missed)



Questions

