linaro@linaro-developer:~/sensor/src/sensor/scripts$ cat sensor.py

#!/usr/bin/env python

import rospy

from std\_msgs.msg import String

from sensor\_msgs.msg import Range

import serial

import sys

def talker():

    port = serial.Serial("/dev/ttyUSB0", baudrate=115200, timeout=0.5)

    range\_msg\_front = Range()

    pub\_front = rospy.Publisher('ultrasound\_front', Range, queue\_size=1)

    range\_msg\_back = Range()

    pub\_back = rospy.Publisher('ultrasound\_back', Range, queue\_size=1)

    range\_msg\_left = Range()

    pub\_left = rospy.Publisher('ultrasound\_left', Range, queue\_size=1)

    range\_msg\_right = Range()

    pub\_right = rospy.Publisher('ultrasound\_right', Range, queue\_size=1)

    rospy.init\_node('sensors', anonymous=True)

    #rate = rospy.Rate(10) # 10hz

    while not rospy.is\_shutdown():

        range\_msg\_front.radiation\_type = 0

        range\_msg\_front.header.frame\_id =  'ultrasound\_front'

        range\_msg\_front.field\_of\_view = 0.10000000149

        range\_msg\_front.min\_range = 0.0

        range\_msg\_front.max\_range = 440

        range\_msg\_front.header.stamp = rospy.Time.now()

        range\_msg\_back.radiation\_type = 0

        range\_msg\_back.header.frame\_id =  'ultrasound\_back'

        range\_msg\_back.field\_of\_view = 0.10000000149

        range\_msg\_back.min\_range = 0.0

        range\_msg\_back.max\_range = 440

        range\_msg\_back.header.stamp = rospy.Time.now()

        range\_msg\_left.radiation\_type = 0

        range\_msg\_left.header.frame\_id =  'ultrasound\_left'

        range\_msg\_left.field\_of\_view = 0.10000000149

        range\_msg\_left.min\_range = 0.0

        range\_msg\_left.max\_range = 440

        range\_msg\_left.header.stamp = rospy.Time.now()

        range\_msg\_right.radiation\_type = 0

        range\_msg\_right.header.frame\_id =  'ultrasound\_right'

        range\_msg\_right.field\_of\_view = 0.10000000149

        range\_msg\_right.min\_range = 0.0

        range\_msg\_right.max\_range = 440

        range\_msg\_right.header.stamp = rospy.Time.now()

        try:

            line = port.readline()

            data = line.split('=')

            print "======="

            print line

            if data[0]=='front' and int(data[1]) < range\_msg\_front.max\_range and int(data[1]) > range\_msg\_front.min\_range:

                range\_msg\_front.range = int(data[1])

            if data[0]=='back' and int(data[1]) < range\_msg\_back.max\_range and int(data[1]) > range\_msg\_back.min\_range:

                range\_msg\_back.range = int(data[1])

            if data[0]=='left'and int(data[1]) < range\_msg\_left.max\_range and int(data[1]) > range\_msg\_left.min\_range:

                range\_msg\_left.range = int(data[1])

            if data[0]=='right'and int(data[1]) < range\_msg\_right.max\_range and int(data[1]) > range\_msg\_right.min\_range:

                range\_msg\_right.range = int(data[1])

            print "======="

        except:

            range\_msg\_front.range = -1

            range\_msg\_back.range = -1

            range\_msg\_left.range = -1

            range\_msg\_left.range = -1

        '''rospy.loginfo(range\_msg\_front)

        rospy.loginfo(range\_msg\_back)

        rospy.loginfo(range\_msg\_left)

        rospy.loginfo(range\_msg\_right)'''

        pub\_front.publish(range\_msg\_front)

        pub\_back.publish(range\_msg\_back)

        pub\_left.publish(range\_msg\_left)

        pub\_right.publish(range\_msg\_right)

        #rate.sleep()

if \_\_name\_\_ == '\_\_main\_\_':

    try:

        talker()

    except rospy.ROSInterruptException:

        pass

linaro@linaro-developer:~/sensor/src/sensor/scripts$

hc-sr04-final​

#include <SoftwareSerial.h>

#include <SoftwareSerial.h>

/\*

 HC-SR04 Ping distance sensor:

 VCC to arduino 5v

 GND to arduino GND

 Echo to Arduino pin 7

 Trig to Arduino pin 8

 This sketch originates from Virtualmix: <http://goo.gl/kJ8Gl>

 Has been modified by Winkle ink here: <http://winkleink.blogspot.com.au/2012/05/arduino-hc-sr04-ultrasonic-distance.html>

 And modified further by ScottC here: <http://arduinobasics.blogspot.com.au/2012/11/arduinobasics-hc-sr04-ultrasonic-sensor.html>

 on 10 Nov 2012.

 \*/

// #define echoPin A1 // Echo Pin

// #define trigPin A0 // Trigger Pin

#define echoPin4 10 // Echo Pin

#define trigPin4 9 // Trigger Pin

#define echoPin3 8 // Echo Pin

#define trigPin3 7 // Trigger Pin

#define echoPin2 6 // Echo Pin

#define trigPin2 5 // Trigger Pin

#define echoPin1 4 // Echo Pin

#define trigPin1 3 // Trigger Pin

#define LEDPin 13 // Onboard LED

int maximumRange = 200; // Maximum range needed

int minimumRange = 0; // Minimum range needed

long duration1, distance1,duration2, distance2,duration3, distance3,duration4, distance4 ; // Duration used to calculate distance

void setup() {

 //Serial.begin (9600);

 Serial.begin(115200);

 pinMode(trigPin1, OUTPUT);

 pinMode(echoPin1, INPUT);

 pinMode(trigPin2, OUTPUT);

 pinMode(echoPin2, INPUT);

 pinMode(trigPin3, OUTPUT);

 pinMode(echoPin3, INPUT);

 pinMode(trigPin4, OUTPUT);

 pinMode(echoPin4, INPUT);

 pinMode(LEDPin, OUTPUT); // Use LED indicator (if required)

}

void loop() {

/\* The following trigPin/echoPin cycle is used to determine the

 distance of the nearest object by bouncing soundwaves off of it. \*/

 digitalWrite(trigPin1, LOW);

 delayMicroseconds(2);

 digitalWrite(trigPin1, HIGH);

 delayMicroseconds(10);

 digitalWrite(trigPin1, LOW);

 duration1 = pulseIn(echoPin1, HIGH);

 //Calculate the distance (in cm) based on the speed of sound.

 distance1 = duration1/58.2;

 // if (distance >= maximumRange || distance <= minimumRange){

 if (0) {

 /\* Send a negative number to computer and Turn LED ON

 to indicate "out of range" \*/

 // Serial.println("-1");

  // digitalWrite(LEDPin, HIGH);

 }

 else {

 /\* Send the distance to the computer using Serial protocol, and

 turn LED OFF to indicate successful reading. \*/

// Serial.println(distance1);

  Serial.println("front="+String(distance1));

 // digitalWrite(LEDPin, LOW);

 }

 //Delay 50ms before next reading.

 //delay(50);

delay(10);

 digitalWrite(trigPin2, LOW);

 delayMicroseconds(2);

 digitalWrite(trigPin2, HIGH);

 delayMicroseconds(10);

 digitalWrite(trigPin2, LOW);

 duration2 = pulseIn(echoPin2, HIGH);

 //Calculate the distance (in cm) based on the speed of sound.

 distance2 = duration2/58.2;

 // if (distance >= maximumRange || distance <= minimumRange){

 if (0) {

 /\* Send a negative number to computer and Turn LED ON

 to indicate "out of range" \*/

  Serial.println("-1");

  // digitalWrite(LEDPin, HIGH);

 }

 else {

 /\* Send the distance to the computer using Serial protocol, and

 turn LED OFF to indicate successful reading. \*/

 //Serial.println(distance2);

 Serial.println("right="+String(distance2));

 // digitalWrite(LEDPin, LOW);

 }

 //Delay 50ms before next reading.

 // delay(50);

delay(10);

digitalWrite(trigPin3, LOW);

 delayMicroseconds(2);

 digitalWrite(trigPin3, HIGH);

 delayMicroseconds(10);

 digitalWrite(trigPin3, LOW);

 duration3 = pulseIn(echoPin3, HIGH);

 //Calculate the distance (in cm) based on the speed of sound.

 distance3 = duration3/58.2;

 // if (distance >= maximumRange || distance <= minimumRange){

 if (0) {

 /\* Send a negative number to computer and Turn LED ON

 to indicate "out of range" \*/

 Serial.println("-1");

 // digitalWrite(LEDPin, HIGH);

 }

 else {

 /\* Send the distance to the computer using Serial protocol, and

 turn LED OFF to indicate successful reading. \*/

// Serial.println(distance3);

   Serial.println("back="+String(distance3));

 // digitalWrite(LEDPin, LOW);

 }

 //Delay 50ms before next reading.

// delay(50);

delay(10);

digitalWrite(trigPin4, LOW);

 delayMicroseconds(2);

 digitalWrite(trigPin4, HIGH);

 delayMicroseconds(10);

 digitalWrite(trigPin4, LOW);

 duration4 = pulseIn(echoPin4, HIGH);

 //Calculate the distance (in cm) based on the speed of sound.

 distance4 = duration4/58.2;

 // if (distance >= maximumRange || distance <= minimumRange){

 if (0) {

 /\* Send a negative number to computer and Turn LED ON

 to indicate "out of range" \*/

 Serial.println("-1");

 // digitalWrite(LEDPin, HIGH);

 }

 else {

 /\* Send the distance to the computer using Serial protocol, and

 turn LED OFF to indicate successful reading. \*/

 //Serial.println(distance4);

 Serial.println("left="+String(distance4));

 // digitalWrite(LEDPin, LOW);

 }

 //Delay 50ms before next reading.

 // delay(50);

delay(10);

}

​

**发件人:** Zhao, Haijun  
**发送时间:** 2016年5月24日 16:27  
**收件人:** Zhao, Haijun  
**主题:** 答复: Eagle checking

root@linaro-developer:/home/linaro/sensor/src/sensor/scripts# cat talker.py  
#!/usr/bin/env python  
  
import rospy  
from std\_msgs.msg import String  
from sensor\_msgs.msg import Range  
import serial  
import sys  
  
  
def talker():  
    port = serial.Serial("/dev/ttyUSB0", baudrate=115200, timeout=0.5)  
    range\_msg = Range()  
    pub = rospy.Publisher('/ultrasound', Range, queue\_size=1)  
    rospy.init\_node('talker', anonymous=True)  
    rate = rospy.Rate(1) # 10hz  
    while not rospy.is\_shutdown():  
        range\_msg.radiation\_type = 0  
        range\_msg.header.frame\_id =  'ultrasound'  
        range\_msg.field\_of\_view = 0.10000000149  
        range\_msg.min\_range = 0.0  
        range\_msg.max\_range = 4.4  
        try:  
            line = port.readline()  
            data = line.split('=')  
            print "======="  
            print line  
            if data[0]=='loop':  
                range\_msg.range = int(data[1])  
            print "======="  
            #range\_msg.range = line  
        except:  
            range\_msg.range = -1  
        range\_msg.header.stamp = rospy.Time.now()  
        rospy.loginfo(range\_msg)  
        pub.publish(range\_msg)  
        rate.sleep()  
  
if \_\_name\_\_ == '\_\_main\_\_':  
    try:  
        talker()  
    except rospy.ROSInterruptException:  
        pass

 /\*  
  Software serial multple serial test  
  
 Receives from the hardware serial, sends to software serial.  
 Receives from software serial, sends to hardware serial.  
  
 The circuit:  
 \* RX is digital pin 10 (connect to TX of other device)  
 \* TX is digital pin 11 (connect to RX of other device)  
  
 Note:  
 Not all pins on the Mega and Mega 2560 support change interrupts,  
 so only the following can be used for RX:  
 10, 11, 12, 13, 50, 51, 52, 53, 62, 63, 64, 65, 66, 67, 68, 69  
  
 Not all pins on the Leonardo support change interrupts,  
 so only the following can be used for RX:  
 8, 9, 10, 11, 14 (MISO), 15 (SCK), 16 (MOSI).  
  
 created back in the mists of time  
 modified 25 May 2012  
 by Tom Igoe  
 based on Mikal Hart's example  
  
 This example code is in the public domain.  
  
 \*/  
#include <SoftwareSerial.h>  
#include <Wire.h>  
  
#define SLAVE\_ADDRESS 0x70  
  
  
SoftwareSerial mySerial(10, 11); // RX, TX  
  
long value=123456;  
char buf[100];  
String incoming;  
void setup() {  
  // Open serial communications and wait for port to open:  
  Serial.begin(115200);  
    
  while (!Serial) {  
    ; // wait for serial port to connect. Needed for native USB port only  
  }  
 Serial.println("Serial ready");  
  // set the data rate for the SoftwareSerial port  
  mySerial.begin(115200);  
  
    while (!mySerial) {  
    ; // wait for serial port to connect. Needed for native USB port only  
  }  
  Serial.println("MySerial ready");  
  Wire.begin(SLAVE\_ADDRESS);  
 Wire.onRequest(sendData);  
    
}  
  
void loop() { // run over and over  
  if (mySerial.available()) {  
    incoming = mySerial.readString();  
    Serial.print("mySerial received: ");  
    Serial.println(incoming);  
  }  
         
       mySerial.print(value);  
       delay(10);  
       Serial.println("loop="+String(value));  
       value=value+1;  
       if (value > 999999)  
       {  
        value=0;  
       }  
}  
  
void sendData(){  
   
//  itoa(value,buf,10);  
   value=value+1;  
  ltoa(value,buf,10);  
  Wire.write(buf);  
    
  Serial.println("===on request sendData()  Wire.write ...itoa[value,buf,10]");  
    
}

**发件人:** Zhao, Haijun  
**发送时间:** 2016年5月24日 10:14  
**收件人:** Zhao, Haijun  
**主题:** 答复: Eagle checking

(UNO 的1，2 PIN   RX/TX)

linaro@linaro-developer:~$ cat read.py  
import serial  
import sys  
port = serial.Serial("/dev/ttyUSB0", baudrate=115200, timeout=5.0)  
  
while True:  
    try:  
        line = port.readline()  
        print line  
        print "==========="  
    except:  
        print "error..."  
    sys.exit(0)

**发件人:** Zhao, Haijun  
**发送时间:** 2016年5月18日 16:48  
**收件人:** Zhao, Haijun  
**主题:** RE: Eagle checking

最后检查ＥＡＧＬＥ没有可用的ＳＰＩ，　　ＮＥＳＴ也可用的SPI)  暂时结束.

1.       CONSOLE UART   (EAGLE 或NEST上的都可使用)

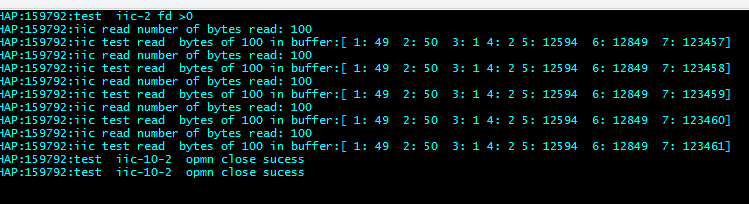
2.       ADSP　tty-3，tty-4

3.       Adsp     nest iic10,   eagle iic-2

4.       ANdriod WIFI

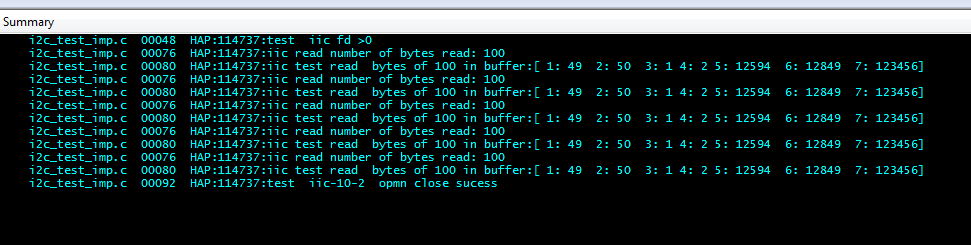
5.       Eagle USB (没解决)．

**From:** Zhao, Haijun   
**Sent:** Tuesday, May 17, 2016 4:05 PM  
**To:** Zhao, Haijun <[haijunz@qti.qualcomm.com](mailto:haijunz@qti.qualcomm.com)>  
**Subject:** RE: Eagle checking



**From:** Zhao, Haijun   
**Sent:** Tuesday, May 17, 2016 3:11 PM  
**To:** Zhao, Haijun <[haijunz@qti.qualcomm.com](mailto:haijunz@qti.qualcomm.com)>  
**Subject:** RE: Eagle checking

/dev/iic-10 also can read



/\*==============================================================================

Copyright (c) 2015 Qualcomm Technologies, Inc.

All rights reserved. Qualcomm Proprietary and Confidential.

==============================================================================\*/

#include <fcntl.h>

#include <unistd.h>

#include <stdint.h>

#include <dev\_fs\_lib\_i2c.h>

#include <dspal\_errno.h>

#define FARF\_MEDIUM 1

#include <HAP\_farf.h>

#include <stdlib.h>

/\*\*

\* @brief Test to see i2c device can be opened and configured.

\*

\* @par

\* Test:

\* 1) Open the i2c device (/dev/iic-9)

\* 2) Configure the i2c device to have (using ioctl):

\*     -Slave address: 0x70

\*     -Bus Frequency in khz: 400

\*     -Transfer timeout in usec: 9000

\* 3) Close the i2c device

\*

\* @return

\* SUCCESS ------ Test Passes

\* ERROR ------ Test Failed

\*/

int dspal\_tester\_i2c\_test(void)

{

   int ret = SUCCESS;

   /\*

    \* Open i2c device

    \*/

   int fd = -1;

  // fd = open("/dev/iic-9", 0);

     fd = open("/dev/iic-10", 0);

   if (fd > 0)

   {

      /\*

       \* Configure I2C device

       \*/

          FARF(MEDIUM, "test  iic fd >0 ");

      struct dspal\_i2c\_ioctl\_slave\_config slave\_config;

      slave\_config.slave\_address = 0x70;

      slave\_config.bus\_frequency\_in\_khz = 400;

      slave\_config.byte\_transer\_timeout\_in\_usecs = 9000;

      if (ioctl(fd, I2C\_IOCTL\_CONFIG, &slave\_config) != 0)

         ret = ERROR;

         int read\_cycle\_count = 0;

         char buf[100];

       // write(fd, buf, sizeof(buf));

         short num\_bytes\_read;

       /\*,num\_bytes\_written;

         memset(buf, 0, sizeof(buf));

         num\_bytes\_written =write(fd, buf,sizeof(buf));

         FARF(MEDIUM, "iic write number of bytes read: %d", num\_bytes\_written);

       \*/

         for (read\_cycle\_count = 0; read\_cycle\_count <5; read\_cycle\_count++)

       {

              memset(buf, 0, sizeof(buf));

              // usleep(100); /\* delay xxx milliseconds \*/

              num\_bytes\_read = read(fd, buf, sizeof(buf));

              FARF(MEDIUM, "iic read number of bytes read: %d", num\_bytes\_read);

              if (num\_bytes\_read > 0)

              {

                     FARF(MEDIUM, "iic test read  bytes of %d in buffer:[ 1: %d  2: %d  3: %c 4: %c 5: %d  6: %d  7: %d]", num\_bytes\_read, (short) buf[0],

                           (short) buf[1], buf[0],buf[1], (short) ((buf[0] << 8) | (buf[1] & 0xff)),(short) ((buf[1] << 8) | (buf[0] & 0xff)),  atoi(buf) );

              }

         }

         /\*

       \* Close the device ID

       \*/

         close(fd);

         FARF(MEDIUM, "test  iic-10-2  opmn close sucess");

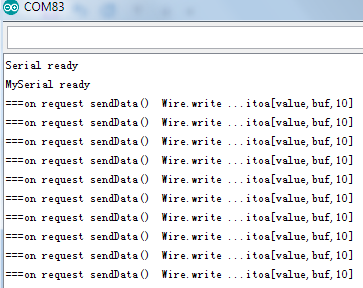
   }

   else

      ret = ERROR;

   return ret;

}



/\*

  Software serial multple serial test

Receives from the hardware serial, sends to software serial.

Receives from software serial, sends to hardware serial.

The circuit:

\* RX is digital pin 10 (connect to TX of other device)

\* TX is digital pin 11 (connect to RX of other device)

Note:

Not all pins on the Mega and Mega 2560 support change interrupts,

so only the following can be used for RX:

10, 11, 12, 13, 50, 51, 52, 53, 62, 63, 64, 65, 66, 67, 68, 69

Not all pins on the Leonardo support change interrupts,

so only the following can be used for RX:

8, 9, 10, 11, 14 (MISO), 15 (SCK), 16 (MOSI).

created back in the mists of time

modified 25 May 2012

by Tom Igoe

based on Mikal Hart's example

This example code is in the public domain.

\*/

#include <SoftwareSerial.h>

#include <Wire.h>

#define SLAVE\_ADDRESS 0x70

SoftwareSerial mySerial(10, 11); // RX, TX

int count=0;

long value=123456;

char buf[100];

void setup() {

  // Open serial communications and wait for port to open:

  Serial.begin(115200);

  while (!Serial) {

    ; // wait for serial port to connect. Needed for native USB port only

  }

Serial.println("Serial ready");

  // set the data rate for the SoftwareSerial port

  mySerial.begin(115200);

    while (!mySerial) {

    ; // wait for serial port to connect. Needed for native USB port only

  }

  Serial.println("MySerial ready");

  count=0;

Wire.begin(SLAVE\_ADDRESS);

Wire.onRequest(sendData);

}

void loop() { // run over and over

  // if (mySerial.available()) {

  // }

   if (count ==0) {

    mySerial.print("1");

    count=count+1;

    };

   if (count ==1) {

    mySerial.write("2");

    count=count+1;

    };

  //   mySerial.print(1);

  //  Serial.println("MySerial in loop() , [count==0 mySerial.print(1)] [cunt=1 mySerial.write(1)  ]");

   // Serial.println("MySerial in loop()");

}

void sendData(){

//  itoa(value,buf,10);

value=value+1;

  ltoa(value,buf,10);

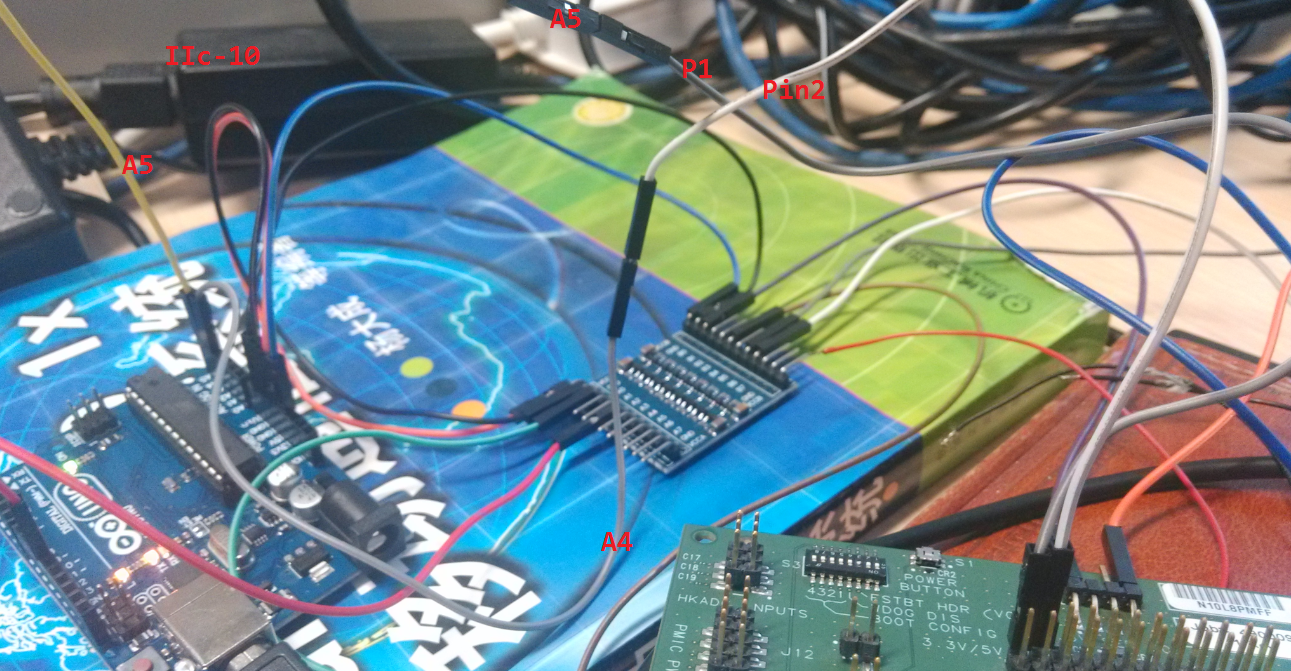
  Wire.write(buf);

  Serial.println("===on request sendData()  Wire.write ...itoa[value,buf,10]");

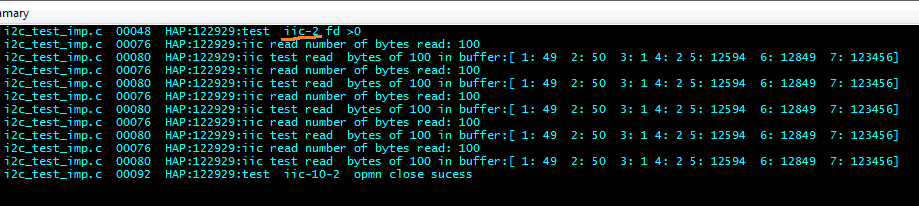
}

Pin1---A5

Pin2 –A4



/dev/iic-2 can read data



/\*==============================================================================

Copyright (c) 2015 Qualcomm Technologies, Inc.

All rights reserved. Qualcomm Proprietary and Confidential.

==============================================================================\*/

#include <fcntl.h>

#include <unistd.h>

#include <stdint.h>

#include <dev\_fs\_lib\_i2c.h>

#include <dspal\_errno.h>

#define FARF\_MEDIUM 1

#include <HAP\_farf.h>

#include <stdlib.h>

/\*\*

\* @brief Test to see i2c device can be opened and configured.

\*

\* @par

\* Test:

\* 1) Open the i2c device (/dev/iic-9)

\* 2) Configure the i2c device to have (using ioctl):

\*     -Slave address: 0x70

\*     -Bus Frequency in khz: 400

\*     -Transfer timeout in usec: 9000

\* 3) Close the i2c device

\*

\* @return

\* SUCCESS ------ Test Passes

\* ERROR ------ Test Failed

\*/

int dspal\_tester\_i2c\_test(void)

{

   int ret = SUCCESS;

   /\*

    \* Open i2c device

    \*/

   int fd = -1;

  // fd = open("/dev/iic-9", 0);

     fd = open("/dev/iic-2", 0);

   if (fd > 0)

   {

      /\*

       \* Configure I2C device

       \*/

          FARF(MEDIUM, "test  iic-2 fd >0 ");

      struct dspal\_i2c\_ioctl\_slave\_config slave\_config;

      slave\_config.slave\_address = 0x70;

      slave\_config.bus\_frequency\_in\_khz = 400;

      slave\_config.byte\_transer\_timeout\_in\_usecs = 9000;

      if (ioctl(fd, I2C\_IOCTL\_CONFIG, &slave\_config) != 0)

         ret = ERROR;

         int read\_cycle\_count = 0;

         char buf[100];

       // write(fd, buf, sizeof(buf));

         short num\_bytes\_read;

       /\*,num\_bytes\_written;

         memset(buf, 0, sizeof(buf));

         num\_bytes\_written =write(fd, buf,sizeof(buf));

         FARF(MEDIUM, "iic write number of bytes read: %d", num\_bytes\_written);

       \*/

         for (read\_cycle\_count = 0; read\_cycle\_count <5; read\_cycle\_count++)

       {

              memset(buf, 0, sizeof(buf));

              // usleep(100); /\* delay xxx milliseconds \*/

              num\_bytes\_read = read(fd, buf, sizeof(buf));

              FARF(MEDIUM, "iic read number of bytes read: %d", num\_bytes\_read);

              if (num\_bytes\_read > 0)

              {

                     FARF(MEDIUM, "iic test read  bytes of %d in buffer:[ 1: %d  2: %d  3: %c 4: %c 5: %d  6: %d  7: %d]", num\_bytes\_read, (short) buf[0],

                           (short) buf[1], buf[0],buf[1], (short) ((buf[0] << 8) | (buf[1] & 0xff)),(short) ((buf[1] << 8) | (buf[0] & 0xff)),  atoi(buf) );

              }

         }

         /\*

       \* Close the device ID

       \*/

         close(fd);

         FARF(MEDIUM, "test  iic-10-2  opmn close sucess");

   }

   else

      ret = ERROR;

   return ret;

}

/\*

  Software serial multple serial test

Receives from the hardware serial, sends to software serial.

Receives from software serial, sends to hardware serial.

The circuit:

\* RX is digital pin 10 (connect to TX of other device)

\* TX is digital pin 11 (connect to RX of other device)

Note:

Not all pins on the Mega and Mega 2560 support change interrupts,

so only the following can be used for RX:

10, 11, 12, 13, 50, 51, 52, 53, 62, 63, 64, 65, 66, 67, 68, 69

Not all pins on the Leonardo support change interrupts,

so only the following can be used for RX:

8, 9, 10, 11, 14 (MISO), 15 (SCK), 16 (MOSI).

created back in the mists of time

modified 25 May 2012

by Tom Igoe

based on Mikal Hart's example

This example code is in the public domain.

\*/

#include <SoftwareSerial.h>

#include <Wire.h>

#define SLAVE\_ADDRESS 0x70

SoftwareSerial mySerial(10, 11); // RX, TX

int count=0;

long value=123456;

char buf[100];

void setup() {

  // Open serial communications and wait for port to open:

  Serial.begin(115200);

  while (!Serial) {

    ; // wait for serial port to connect. Needed for native USB port only

  }

Serial.println("Serial ready");

  // set the data rate for the SoftwareSerial port

  mySerial.begin(115200);

    while (!mySerial) {

    ; // wait for serial port to connect. Needed for native USB port only

  }

  Serial.println("MySerial ready");

  count=0;

Wire.begin(SLAVE\_ADDRESS);

Wire.onRequest(sendData);

}

void loop() { // run over and over

  // if (mySerial.available()) {

  // }

   if (count ==0) {

    mySerial.print("1");

    count=count+1;

    };

   if (count ==1) {

    mySerial.write("2");

    count=count+1;

    };

  //   mySerial.print(1);

  //  Serial.println("MySerial in loop() , [count==0 mySerial.print(1)] [cunt=1 mySerial.write(1)  ]");

   // Serial.println("MySerial in loop()");

}

void sendData(){

//  itoa(value,buf,10);

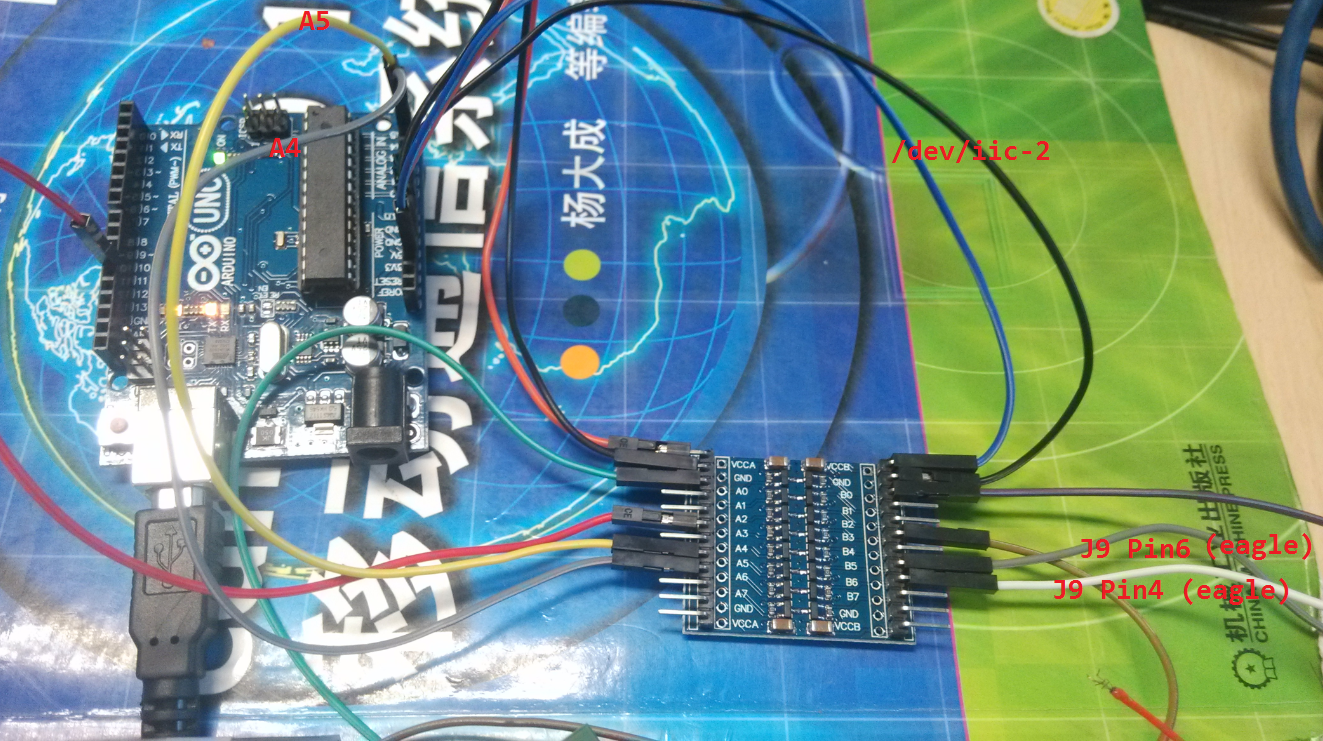
value=value+1;

  ltoa(value,buf,10);

  Wire.write(buf);

  Serial.println("===on request sendData()  Wire.write ...itoa[value,buf,10]");

}



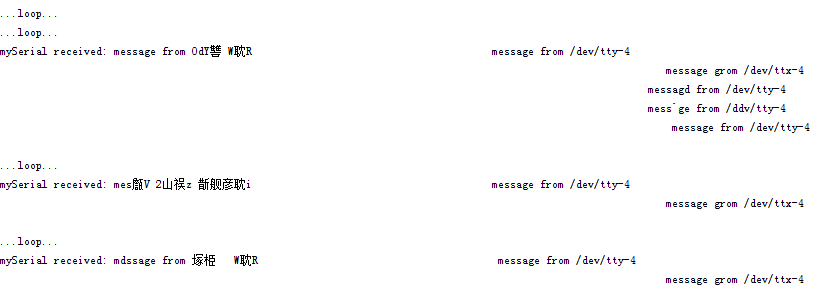
**From:** Zhao, Haijun   
**Sent:** Tuesday, May 10, 2016 6:15 PM  
**To:** Zhao, Haijun <[haijunz@qti.qualcomm.com](mailto:haijunz@qti.qualcomm.com)>  
**Subject:** RE: Eagle checking

delay(1000);

Result  tty-1 , tty-2 cannot open from adsp.

tty-4 can be used from adsp  :  one pin  connect  RX(pin3) —TX[ardunion]   测试时将J9的两个3V UART　同　Arduino 直接连接，　（**通过逻辑电平反而不通，　经过检查线连到了逻辑电平最下两排也是ＧＮＤ，ＶＣＣ，　后将线连接到逻辑电平Ａ１－Ｂ１，　Ａ２－Ｂ１后成功**）

adsp 侧不稳定，　多数收不到，　原因是read 不是block call　　（adsp采用readloop 后可以收到，　　但速度慢，　检查原因　　Arduino侧delay(1000)　**改为 delay(100)后速度加快;）**



/\*

  Software serial multple serial test

Receives from the hardware serial, sends to software serial.

Receives from software serial, sends to hardware serial.

The circuit:

\* RX is digital pin 10 (connect to TX of other device)

\* TX is digital pin 11 (connect to RX of other device)

Note:

Not all pins on the Mega and Mega 2560 support change interrupts,

so only the following can be used for RX:

10, 11, 12, 13, 50, 51, 52, 53, 62, 63, 64, 65, 66, 67, 68, 69

Not all pins on the Leonardo support change interrupts,

so only the following can be used for RX:

8, 9, 10, 11, 14 (MISO), 15 (SCK), 16 (MOSI).

created back in the mists of time

modified 25 May 2012

by Tom Igoe

based on Mikal Hart's example

This example code is in the public domain.

\*/

#include <SoftwareSerial.h>

#include <Wire.h>

#define SLAVE\_ADDRESS 0x70

SoftwareSerial mySerial(10, 11); // RX, TX

long value=123456;

char buf[100];

String incoming;

void setup() {

  // Open serial communications and wait for port to open:

  Serial.begin(115200);

  while (!Serial) {

    ; // wait for serial port to connect. Needed for native USB port only

  }

Serial.println("Serial ready");

  // set the data rate for the SoftwareSerial port

  mySerial.begin(115200);

    while (!mySerial) {

    ; // wait for serial port to connect. Needed for native USB port only

  }

  Serial.println("MySerial ready");

  Wire.begin(SLAVE\_ADDRESS);

Wire.onRequest(sendData);

}

void loop() { // run over and over

  if (mySerial.available()) {

    incoming = mySerial.readString();

    Serial.print("mySerial received: ");

    Serial.println(incoming);

  }

       mySerial.print(value);

      delay(1000);

        Serial.println("...loop...");

     value=value+1;

}

void sendData(){

//  itoa(value,buf,10);

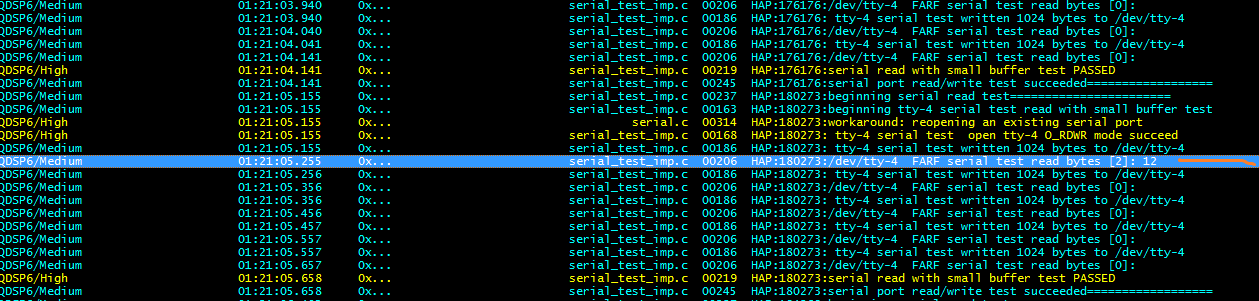
   value=value+1;

  ltoa(value,buf,10);

  Wire.write(buf);

  Serial.println("===on request sendData()  Wire.write ...itoa[value,buf,10]");

}



/\*==============================================================================

Copyright (c) 2015 Qualcomm Technologies, Inc.

All rights reserved. Qualcomm Proprietary and Confidential.

==============================================================================\*/

#include <stdint.h>

#include <errno.h>

#include <fcntl.h>

#include <unistd.h>

#include <dev\_fs\_lib\_serial.h>

#include <stdlib.h>

/\* Enable medium level debugging. \*/

#define FARF\_MEDIUM 1  /\* 0 turns me off \*/

#include <HAP\_farf.h>

#include "dspal\_errno.h"

#include "test\_utils.h"

//#include "dspal\_tester.h"

#include <stdlib.h>

#define SERIAL\_READ\_CYCLES 5

#define SERIAL\_SIZE\_OF\_DATA\_BUFFER\_B 100

int read\_cycle\_count = 0;

int dspal\_tester\_serial\_read\_with\_small\_buffer(void)

{

       int result = SUCCESS;

       int num\_bytes\_written = 0;

       int num\_bytes\_read = 0;

       char tx\_buffer[SERIAL\_SIZE\_OF\_DATA\_BUFFER\_B];

       char rx\_buffer[SERIAL\_SIZE\_OF\_DATA\_BUFFER\_B];

       int fd;

       int devid = 4;

       FARF(MEDIUM, "beginning tty-4 serial test read with small buffer test");

       fd = open("/dev/tty-4", O\_RDWR);

       FARF(HIGH, " tty-4 serial test  open %s O\_RDWR mode %s", "tty-4",

              (fd < SUCCESS) ? "fail":"succeed");

       if (fd < SUCCESS)

       {

              result = ERROR;

              goto exit;

       }

       for (read\_cycle\_count = 0; read\_cycle\_count <5; read\_cycle\_count++)

       {

              memset(tx\_buffer, 0, SERIAL\_SIZE\_OF\_DATA\_BUFFER\_B);

              sprintf(tx\_buffer, "message from /dev/tty-%d\n", devid);

              num\_bytes\_written = write(fd,(const char \*)tx\_buffer,SERIAL\_SIZE\_OF\_DATA\_BUFFER\_B);

              if (num\_bytes\_written == SERIAL\_SIZE\_OF\_DATA\_BUFFER\_B)

              {

                     FARF(MEDIUM, " tty-4 serial test written %d bytes to %s", num\_bytes\_written,

                           "/dev/tty-4");

              }

              else

              {

                     FARF(MEDIUM, "tty-4 serial test failed to write to %s","/dev/tty-4");

                     }

readloop:

              memset(rx\_buffer, 0, SERIAL\_SIZE\_OF\_DATA\_BUFFER\_B);

              num\_bytes\_read = read(fd, rx\_buffer, SERIAL\_SIZE\_OF\_DATA\_BUFFER\_B);

              if (num\_bytes\_read < 1)

              {

                     FARF(MEDIUM, "tty-4 serial test  read()  \*\*\*\*\*\* return expected  code[%d]",num\_bytes\_read);

                     goto readloop;

              } else

              {

                     FARF(MEDIUM, "%s  FARF serial test read bytes [%d]: %s","/dev/tty-4", num\_bytes\_read, rx\_buffer);

                     FARF(MEDIUM, "FARF serial test read int  [%d]:" , atoi(rx\_buffer) );

              }

       }

exit:

       if (fd >= SUCCESS)

       {

              close(fd);

       }

       FARF(HIGH, "serial read with small buffer test %s",

              result == SUCCESS ? "PASSED":"FAILED");

       return result;

}

/\*\*

\* @brief Runs all the serial tests and returns 1 aggregated result.

\*

\* @return

\* SUCCESS ------ All tests pass

\* ERROR -------- One or more tests failed

\*/

int dspal\_tester\_serial\_test(void)

{

       int result;

       FARF(MEDIUM, "beginning serial read test=======================");

       // if ((result = flight\_tester\_test\_serial\_read()) < SUCCESS)

       if ((result =dspal\_tester\_serial\_read\_with\_small\_buffer()) < SUCCESS)

       {

              FARF(HIGH, "error: serial port read test failed: %d", result);

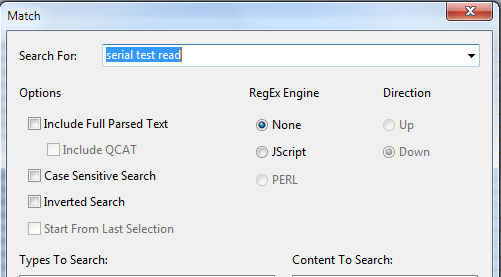
              return result;

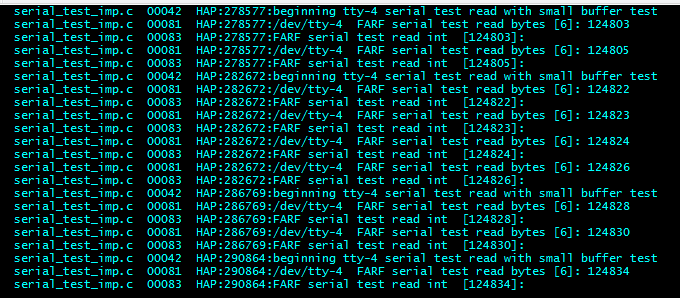
       }

       FARF(MEDIUM, "serial port read/write test succeeded==================");

       return result;

}





tty-3 can be used from adsp  :  one pin  connect  RX(pin3) —TX[ardunion]   测试时将J9的两个3V UART　同　Arduino 直接连接，　（**通过逻辑电平反而不通**）

adsp 侧不稳定，　多数收不到，　原因是read 不是block call

/\*==============================================================================

Copyright (c) 2015 Qualcomm Technologies, Inc.

All rights reserved. Qualcomm Proprietary and Confidential.

==============================================================================\*/

#include <stdint.h>

#include <errno.h>

#include <fcntl.h>

#include <unistd.h>

#include <dev\_fs\_lib\_serial.h>

#include <stdlib.h>

/\* Enable medium level debugging. \*/

#define FARF\_MEDIUM 1  /\* 0 turns me off \*/

#include <HAP\_farf.h>

#include "dspal\_errno.h"

#include "test\_utils.h"

//#include "dspal\_tester.h"

#include <stdlib.h>

#define SERIAL\_READ\_CYCLES 5

#define SERIAL\_SIZE\_OF\_DATA\_BUFFER\_B 100

int read\_cycle\_count = 0;

int dspal\_tester\_serial\_read\_with\_small\_buffer(void)

{

       int result = SUCCESS;

       int num\_bytes\_read = 0;

       char rx\_buffer[SERIAL\_SIZE\_OF\_DATA\_BUFFER\_B];

       int fd;

       FARF(MEDIUM, "beginning tty-3 serial test read with small buffer test");

       fd = open("/dev/tty-3", O\_RDWR);

       FARF(HIGH, " tty-3 serial test  open %s O\_RDWR mode %s", "tty-3",

              (fd < SUCCESS) ? "fail":"succeed");

       if (fd < SUCCESS)

       {

              result = ERROR;

              goto exit;

       }

       for (read\_cycle\_count = 0; read\_cycle\_count <5; read\_cycle\_count++)

       {

readloop:

              memset(rx\_buffer, 0, SERIAL\_SIZE\_OF\_DATA\_BUFFER\_B);

              num\_bytes\_read = read(fd, rx\_buffer, SERIAL\_SIZE\_OF\_DATA\_BUFFER\_B);

              if (num\_bytes\_read < 1)

              {

                     FARF(MEDIUM, "tty-3 serial test  read()  \*\*\*\*\*\* return expected  code[%d]",num\_bytes\_read);

                     goto readloop;

              } else

              {

                     FARF(MEDIUM, "%s  FARF serial test read bytes [%d]: %s","/dev/tty-3", num\_bytes\_read, rx\_buffer);

                     FARF(MEDIUM, "FARF serial test read int  [%d]:" , atoi(rx\_buffer) );

              }

       }

exit:

       if (fd >= SUCCESS)

       {

              close(fd);

       }

       FARF(HIGH, "serial read with small buffer test %s",

              result == SUCCESS ? "PASSED":"FAILED");

       return result;

}

/\*\*

\* @brief Runs all the serial tests and returns 1 aggregated result.

\*

\* @return

\* SUCCESS ------ All tests pass

\* ERROR -------- One or more tests failed

\*/

int dspal\_tester\_serial\_test(void)

{

       int result;

       FARF(MEDIUM, "beginning serial read test=======================");

       if ((result =dspal\_tester\_serial\_read\_with\_small\_buffer()) < SUCCESS)

       {

              FARF(HIGH, "error: serial port read test failed: %d", result);

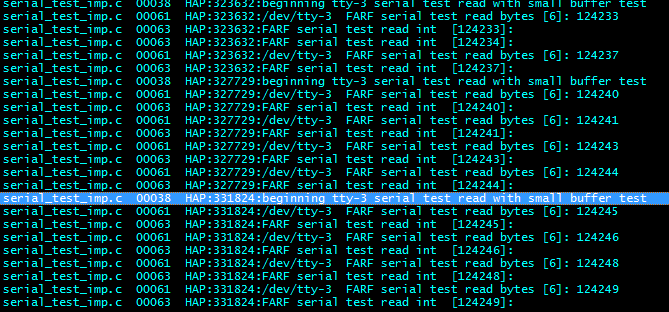
              return result;

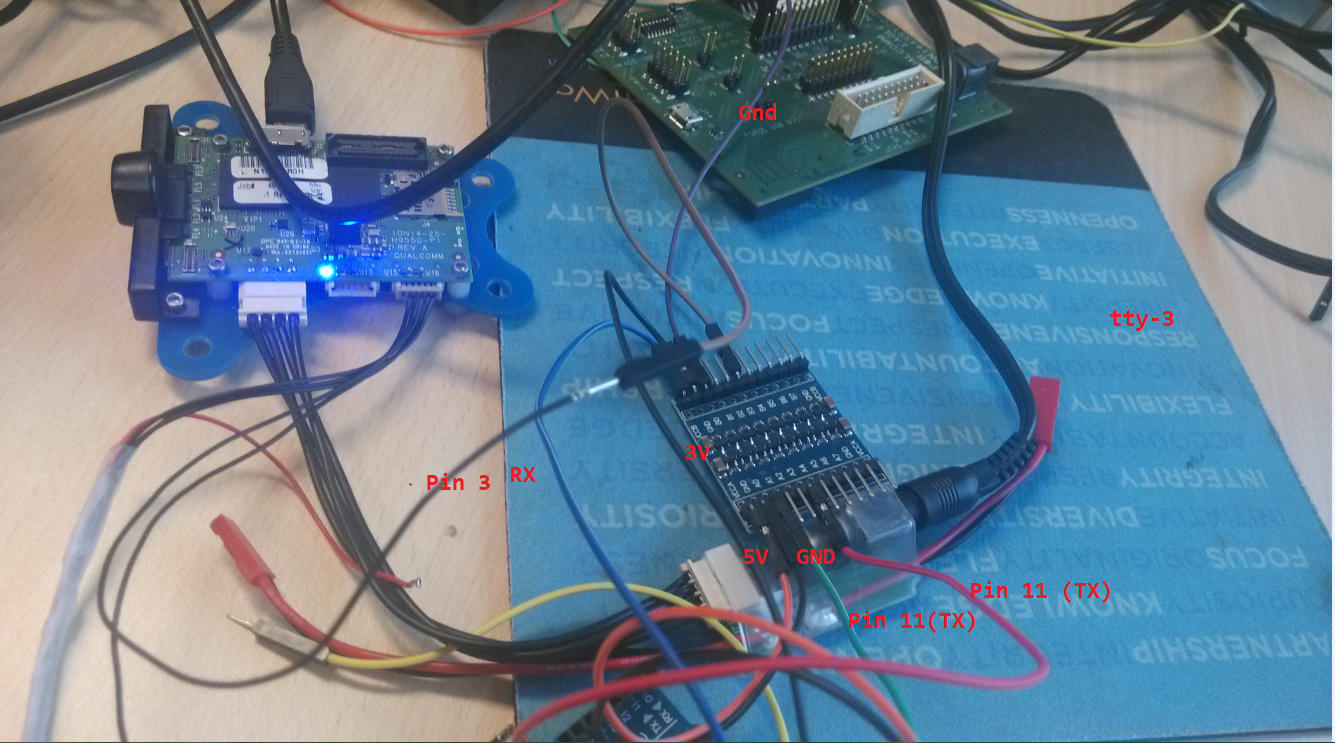
       }

       FARF(MEDIUM, "serial port read/write test succeeded==================");

       return result;

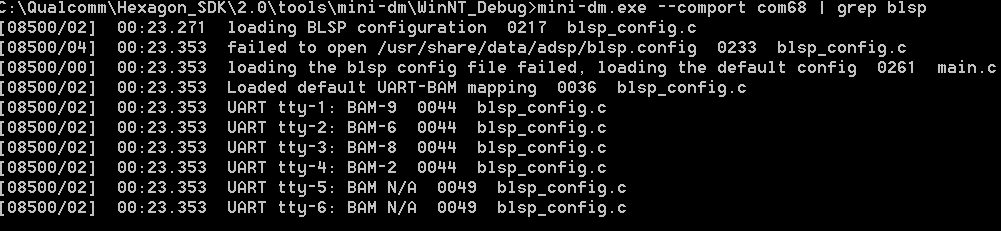
}

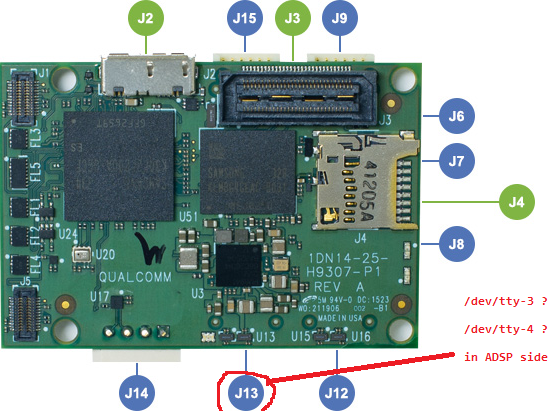


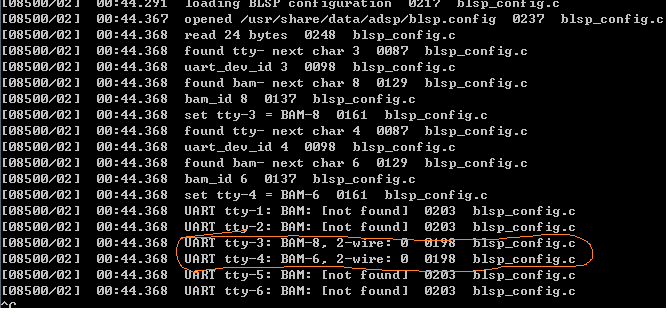


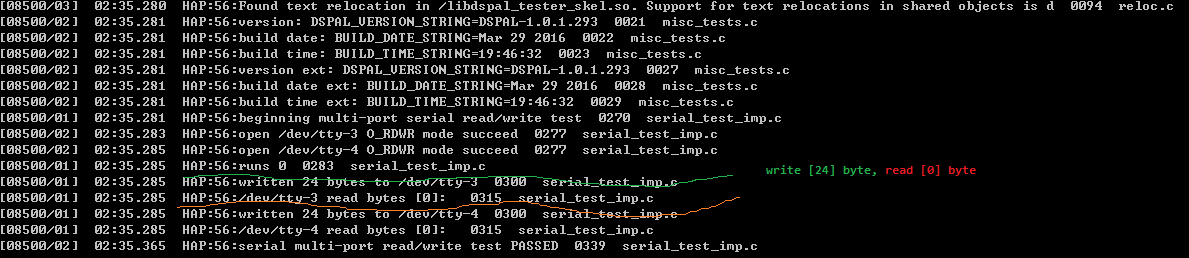


**From:** Zhao, Haijun   
**Sent:** Tuesday, May 10, 2016 5:55 PM  
**To:** Zhao, Haijun <[haijunz@qti.qualcomm.com](mailto:haijunz@qti.qualcomm.com)>  
**Subject:** Eagle checking

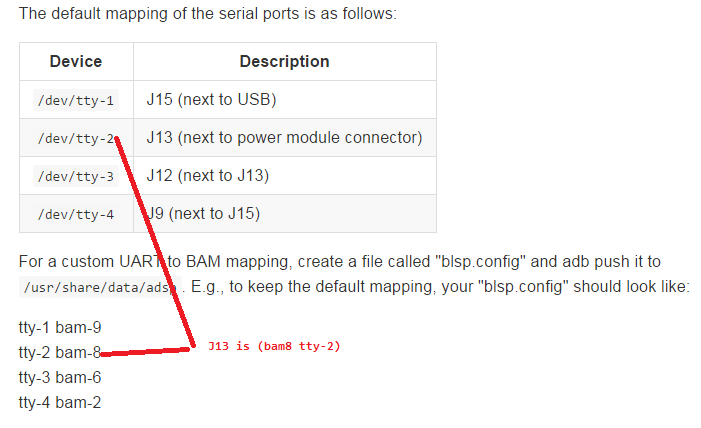








<http://dev.px4.io/hardware-snapdragon.html>    (I’m not sure this is correct)



|  | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **J9** | BLSP2 off-board (external IMU or GPS) | Hirose | DF13-6P-1.25H(50) | Pin 1: 3.3V Pin 2: UART2\_TXO Pin 3: UART2\_RXI Pin 4: ii Pin 5: GND Pin 6: I2C2\_SCL | DF13-6S-1.25C cable assemblies | UART and I2C are at 3.3V CMOS levels. I2C has 10K pull-ups. Connect TXO to target device’s RXI, and vice-versa. |
| **J12** | BLSP8 off-board (external gimbal or sensor) | Hirose | DF13-6P-1.25H(50) | Pin 1: 3.3V Pin 2: UART8\_TXO Pin 3: UART8\_RXI Pin 4: APQ\_GPIO\_47 Pin 5: GND Pin 6: APQ\_GPIO\_48 | DF13-6S-1.25C cable assemblies | UART and GPIOs are at 3.3V CMOS levels. Connect TXO to target device’s RXI, and vice-versa. GPIO’s are not set for I2C (no pull-ups, push-pull driver). |
| **J13** | BLSP6 off-board (ESC UART) | Hirose | DF13-6P-1.25H(50) | Pin 1: 5V Pin 2: UART6\_TXO Pin 3: UART6\_RXI Pin 4: APQ\_GPIO\_29 Pin 5: GND Pin 6: APQ\_GPIO\_30 | DF13-6S-1.25C cable assemblies | UART and GPIOs are at 5V CMOS levels. Connect TXO to target device’s RXI, and vice-versa. GPIO’s are not set for I2C (no pull-ups, push-pull driver |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **J15** | BLSP9 off-board (external magnetometer or radio receiver or sensors) | Hirose | DF13-6P-1.25H(50) | Pin 1: 3.3V Pin 2: UART9\_TXO Pin 3: UART9\_RXI Pin 4: I2C9\_SDA Pin 5: GND Pin 6: I2C9\_SCL | DF13-6S-1.25C cable assemblies | UART and I2C are at 3.3V CMOS levels. 3.3V power and I/O’s can be disabled (see schematic). Connect TXO to target device’s RXI, and vice versa. I2C has 10K pull-ups. |