Software development for TJA102X semi automatic test set-up
Final report

<table>
<thead>
<tr>
<th>Author</th>
<th>Jie Su</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student number</td>
<td>00066761</td>
</tr>
<tr>
<td>school</td>
<td>HZ University of Applied Sciences</td>
</tr>
<tr>
<td>Supervisor teacher</td>
<td>Bert Verhage</td>
</tr>
<tr>
<td>Company</td>
<td>NXP Semiconductor Nijmegen</td>
</tr>
<tr>
<td>In-company mentor</td>
<td>John van Zwam, Brecia Nurastu Sasonko</td>
</tr>
<tr>
<td>Date</td>
<td>1.1.2015</td>
</tr>
</tbody>
</table>
Software development for TJA102X semi automatic test set-up

Author: Jie Su
Student number: 00066761
School: HZ University of Applied Sciences
Supervisor teacher: Bert Verhage
Company: NXP Semiconductor Nijmegen
In-company mentor: John van Zwam, Brecia Nurastu Sasonko
Date: 1.1.2015

Signature:
Summary

This is the report of the test setup for TJA102x family which is used for failure testing LIN Transceiver including the chips TJA1020, TJA1021, TJA1022, TJA1024, TJA1027, TJA1028, TJA1029 and UJA1018.

NXP Nijmegen would like to develop the software to test whether the chip has been made is qualified. If the data achieved is different from the datasheet, how to find out the problem.

The test setup can be divided into two parts, hardware design and software design. This project mainly focuses on software part.

The main functions of the software design are explained as followed:

1. The automatic control:
By the software program, a tester can control the external devices which is SMU, DAC, ADC, Oscilloscope and Power I/O automatically to send the voltage supply and some typical signal which can trigger certain function of the chip and read back the data through the test setup. Both manual measurement and automatic measurement is available.

2. The accurate data transceiver
The hardware of the test setup can accurately receive the signal from the external devices and send back the required data to those devices so that the tester can see the result of the test. The software program can control DAC, SMU and Power I/O to set the expected value.

The main task of the test-setup will be introduced as followed:

- The software program:
The software program can have the basic control of the PCB board, DAC, SMU and Power I/O. In the manual measurement, different operation mode can be measured and changed easily. In the automatic measurement, the characteristic in the datasheet should be measured and checked.

- The hardware design improvement:
During testing the program, some errors can happen and make the result goes wrong. It is required to find out the shortage and design mistakes. Giving advice and solutions for the error so that the PCB board can be improved in next version.

The structure design methodology is applied to do this thesis, it followed the steps: problem analysis – problem definition – functional design – physical design – realization - test.

The software generates all the required tests into one user interface; it is also easy to operate even for someone who don’t have this background knowledge.
Foreword

This thesis is a part of the work required for the fulfillment of the Bachelor degree in Electrical Engineering in HZ University of Applied Sciences.

I would like to show my respect to my company mentors John van Zwam and Brecia Nurastu Sasongko. I also wish to thank my school supervisory teacher Bert Verhage.

The report is open to the public only for the purpose of study in the fields of mechatronics, electrical engineering or Electronics. All of the measurements and results can be used for study and researches on the related topics. However, any other usage for profits should be forbidden.

The report contains seven main sections:

- Introduction - which explains the background of the research carried out and the theme/goal for it
- Theoretical framework - which introduces the knowledge used in the project
- Method and materials - which shows how the author carry out his project
- Results - which shows the measurements and calculations that help illustrate the outcome of designed products
- Discussion - which performs analysis on the results got from the above section
- Conclusion - which gives answer to the main problem of the project
- Recommendations – which gives recommendation for later product improvement & development
Abbreviations

A:
ADC: Analog-to-Digital Converter

B:
BEV: Bird’s Eye View

C:
CAN: Controller Area Network

D:
DAC: Digital-to-Analog Converter

E:
EME: Electromagnetic Emission
EPD: End Product Definition
ESD: Electrostatic Discharge

G:
GUI: Graphical User Interface

L:
LIN: Local Interconnect Network

N:
NXP: NXP Semiconductor N.V.

P:
PC: Personal Computer

S:
SDM: Sturcture Development Methodology
SMU: Source Measurement Unit

V:
VCL: Visual Component Library
Contents

1. Introduction .......................................................................................................................... 1
   1.1 Background information ................................................................................................. 1
       1.1.1 Background information about NXP Semiconductor ............................................. 1
       1.1.2 Background information of the thesis ................................................................. 2
   1.2 Theme and goal of the thesis ......................................................................................... 4
2. Theoretical Framework ....................................................................................................... 6
   2.1 Source ........................................................................................................................... 6
   2.2 Context .......................................................................................................................... 7
   2.3 Presupposition ............................................................................................................... 8
   2.4 Concepts ....................................................................................................................... 9
3. Method and materials ......................................................................................................... 11
   3.1 Introduction of Structure Design Methodology (SDM) .................................................. 11
   3.2 Justification .................................................................................................................. 12
4. Results .............................................................................................................................. 13
   4.1 Problem analysis phase ............................................................................................... 13
       4.1.1 Bird’s Eye View .................................................................................................... 14
       4.1.2 End Product Definition ..................................................................................... 15
   4.2 External Overview ........................................................................................................ 18
   4.3 Internal overview ......................................................................................................... 19
   4.4 Functional design ......................................................................................................... 20
       4.4.1 Manual measurement .......................................................................................... 20
       4.4.2 Automatic measurement .................................................................................... 21
   4.5 Physical Design ............................................................................................................ 23
       4.5.1 Manual measurement .......................................................................................... 23
       4.5.2 Automatic measurement .................................................................................... 26
   4.6 Realization phase .......................................................................................................... 27
       4.6.1 Test mode ............................................................................................................ 29
       4.6.2 Curvetrace ........................................................................................................... 33
5 Discussion .......................................................................................................................... 35
   5.1 Answer of sub questions ............................................................................................. 35
   5.2 Answer of main question ............................................................................................ 37
6 Conclusion ........................................................................................................................................... 38
7 Recommendation .................................................................................................................................. 39
8 Bibliography .......................................................................................................................................... 41
Appendices ................................................................................................................................................ 42
   A. The relation between the schematic design and the software ............................................................. 42
   B. Functional design manual measurement ............................................................................................. 44
   C. Functional design automatic measurement ......................................................................................... 47
   D. Physical design manual measurement ................................................................................................ 54
   E. Program code ......................................................................................................................................... 56
   F. Physical design automatic measurement ............................................................................................. 61
   G. Program Code ....................................................................................................................................... 63
   H. Test mode ............................................................................................................................................... 70
   I. Curvetrace ........................................................................................................................................... 73
   J. Schematic design in Altium board ......................................................................................................... 75

Figure

Figure 1 NXP Semiconductor Nijmegen ......................................................................................................... 1
Figure 2 TJA1020 ........................................................................................................................................... 2
Figure 3 Test setup hardware mother board ............................................................................................... 3
Figure 4 Test setup hardware daughter board ............................................................................................ 3
Figure 5 State diagram .................................................................................................................................. 7
Figure 6 Pin select circuit ............................................................................................................................. 8
Figure 7 theoretical framework .................................................................................................................... 9
Figure 8 Structure development methodologies ......................................................................................... 11
Figure 9 Bird’s Eye View ............................................................................................................................ 14
Figure 10 main menu ..................................................................................................................................... 16
Figure 11 TJA1020 manual measurement .................................................................................................. 16
Figure 12 Schematic design ......................................................................................................................... 17
Figure 13 External Overview ....................................................................................................................... 18
Figure 14 Internal overview .......................................................................................................................... 19
Figure 15 State diagram flow chart ........................................................................................................... 21
Figure 16 TJA1020 automatic measurement ............................................................................................... 26
Figure 17 TJA1020 automatic measurement results .................................................................................... 28
Table 1 TJA1020 automatic measurement results ........................................................................29
1. Introduction

Chapter 1 gave the introduction of the background information. The background information about the company told about the history of NXP and the department where this project worked out. The information about this project would give information about the test which would be performed and the devices which would be used for measurement in the test.

The main question and sub questions would be mentioned in this chapter to give a clear idea about what should be done and how to finish the thesis.

1.1 Background information

1.1.1 Background information about NXP Semiconductor

NXP Semiconductors is a semiconductor manufacturer. It is one of the worldwide top 20 semiconductor sales leaders and was founded in 1953, when the Philips Board started a semiconductor operation with manufacturing and development in Nijmegen, Netherlands. The name, NXP, stood for the consumer’s "next experience", according to then-CEO Frans van Houten.

NXP Semiconductors provides mixed signal and standard product solutions based on its RF, analog, power management, interface, security and digital processing expertise. These semiconductors are used in a wide range of "smart" automotive, identification, wireless infrastructure, lighting, industrial, mobile, consumer and computing applications. (NXP, 2006)

This project is developed in NXP Nijmegen in BU Automotive department. This department, The Product Line in Vehicle Networking develops, produces and sells integrated circuits for uses in automotive applications. The design support is responsible for a.o. the creation of electrical test programs. In the lab, there is a common test station where all basic functions are available. On top of that NXP has a product dedicated PCB, the complete datasheet can be measured with a test program.
1.1.2 Background information of the thesis

This assignment is designing a semi automatic test set up for TJA102X family, mainly focus on the software. The products to be tested are TJA102X family. They are a series of Local Interconnect Network (LIN) transceiver including TJA1020/21/22/24/27/28/29 and UJA1018.

LIN is a bus system which is aimed mainly at the automotive industry. It has been designed to provide a low-cost networking solution for low-speed, low-risk applications as an alternative to the high speed, high integrity, higher cost Control Area Network (CAN) bus which is required for engine management, braking and safety features. (LIN solution teacher notes, 2007)

TJA1020 is NXP’s first LIN transceiver. Normally it transmits data for rain sensor, wiper motor, switch panel, steering wheel module and so on.

The transmit data stream of the protocol controller at the TXD input is converted by the LIN transceiver into a bus signal with controlled slew rate and wave shaping to minimize Electromagnetic Emission (EME). The receiver detects the data stream at the LIN bus input pin and transfers it via pin RXD to the microcontroller.

There are several test to be worked out in this thesis. The first one is manual measurement. In the manual measurement, an interface should be made to test the curve and the value for every pin of the chip to check whether it met the requirement.

The second part is called automatic measurement. It is required to test all the characteristics in the datasheet for the chip. Different measurement method and devices will be used here.

The third part named test mode, it is used to test some specifications such as the frequency of the oscillator inside the IC. For some characteristics which is not able to test in the automatic measurement.

The forth part, which is curvetracer, is applied for checking whether the pin has Electrostatic Discharge (ESD) problems or not.

Finally, some improvement in the both software and hardware part could be discussed and tested in the future.
Figure 3 and 4 shows the hardware board of the test setup which is designed by the previous designer. The board is consist of a mother board and a daughter board. The mother board is used to connect the measurement tools. The daughter board is plugged in the mother board. There are 4
ports on the daughter board which can connect the oscilloscope waveform generator. The TJA102x can be plugged in the daughter board.

There are a lot of LEDs on the daughter board. When devices connected to the pin are used, the LED related to that pin will on which is easy to check whether the programming is right or not. The schematic design circuit can be found in Altium board (Appendices J).

The hardware schematic can be divided into 4 parts. The first part is low power pin schematic design. The schematic design was used for pin1 to pin6, pin10 to pin12. The current of the pin should be lower than 200mA, the voltage of the pin should be lower than 60V.

The second part is high power pin schematic design. it is the schematic design for high power pin 15. For TJA1028 and UJA1018, the current from BAT pin supply was higher than 200mA which was larger than the optocoupler could handle. The circuit is designed to handle maximum 3A current. The third part is LIN termination schematic design. The circuit is used to change the load of LIN. If the master chip went wrong, then 1k is connected to LIN. If the slave chip went wrong, then the customer would mention the load whether it was 500Ω or 650Ω. The last part is dataline, clockline and voltage controller schematic design which is used in the test mode. Dataline and clockline is used to shift the bits mentioned in the datasheet for test mode. Voltage controller is used to supply the negative voltage to the pin.

1.2 Theme and goal of the thesis

The aim of the project is to make the software setup for TJA102X family to test whether the chips meet the requirement in the datasheet. The test can be done both manually and automatically. Tasks:
* Good command of knowledge about LIN transceiver operation.
* A user friendly GUI (Graphical User Interface) for manual measurement.
* A user friendly GUI (Graphical User Interface) for automatic measurement.

The aim of this assignment is about making a user friendly GUI (Graphical User Interface) for both manual measurement and automatic measurement. Therefore, the main question will be: How to develop the software for the semi-automatic test set up to verify the TJA102X family specifications, and implement it using Delphi 5?
The sub questions of the thesis will be:

- What is TJA102X family and test setup?
  - What is LIN Transceiver (TJA102X family)?
  - What is the datasheet of TJA102X family?
  - What is automatic test set-up?
  - What should be measured in the test?
  - How to use the measurement tools?
  
  This will be answered both in chapter 1.1.2 Background information of TJA102X Family and test setup and chapter 2 theoretical framework.

- What is the specification for each pin of the test setup?
  - What is the specification for each pin of the test set-up?
  - What is the relationship between each pin?
  
  This will be answered in chapter 4.1 problem analysis phase: Bird’s Eye View (BEV) and End Product Definition (EPD).

- What is the functional block of the program?
  - How to use the VCL designed by NXP to set values on measurement tool?
  - What is the functional block mentioned in Internal Problem Definition phase (S)?
  - What is the function for pins on the chip?
  - What does every pin refer to the in the Altium board?
  
  This will be answered in chapter 4.2 external problem definition and 4.3 internal problem definition.

  - Which function should be enabled or disabled during the measurement?
  - Which pin or switch should be on at the beginning of the test? What should be reset?
  - What kind of waveform in (LIN/TXD/RXD) pin is expected on the oscilloscope?
  
  This will be answered in chapter 4.4 functional design.

- What kind of code can send data to the test setup?
  - How to sweep the voltage or current in the automatic measurement?
  - If the measured data is different from the datasheet, how to solve the problem?
  - What can be improved in the code writing?
  
  This will be answered in chapter 4.5 physical design.
2. Theoretical Framework

Chapter 2 talked about the theoretical framework of this thesis. It can be separated into four parts. The source part is consisted of the source where the background knowledge and useful information for the project can be found. The context illustrated the area of knowledge learning from the source. The presupposition gave idea about what should be acquired before doing the thesis. The last part, concept, explained how to set a theoretical framework for this thesis.

2.1 Source

- Delphi language guide. Scotts Valley, California: Borland Software Corporation. (Borland, 2004 OCT)
  This DELPHI guide illustrated how to use the DELPHI tools to write the program. The function for each VCL can be found in the book. Since this project is based on DELPHI programming, this book is useful and helpful for the student who uses DELPHI for the first time.

- LIN Bus and its potential for use in Distributed. (Johna V.DeNuto S. E., 2011)

- Copyright Matrix Multimedia. (LIN solution teacher notes, 2007)
  These two PDF describes how the LIN bus works. TJA102x Family are LIN transceivers, it is important to understand how the transceivers send and receive data.

- AN00093 TJA1020LIN transceiver Applcaiton note. (Philips, 2005 Sep 16)
- TJA1020 LIN transceiver datasheet. (Semiconductor, 2004 Jan 13)
- PHILIPS. TJA1020 LIN transceiver. (PHILIPS, 2005)
  These three datasheet explain the knowledge for TJA1020. TJA1020 is the first chip to be tested, the measurement method and programming structure can be applied in other types of chips. From the datasheet, information such as block diagram, state diagram and characteristics, all general description can be found.

- MOSFET modeling & BSIM3 user's guide. China. (Yuhua Cheng, 1999)
  This book tells the basic knowledge of electric circuit. The mosfet and optocouplers are widely used in the hardware schematic. Besides, the board still has some mistakes in the design which have to be improved, this book helps the student to learn how to connect circuit and what kind of components can be used.

These documents are the datasheets for other type of chips in TJA102X family. The GUIs for manual measurement and automatic measurement are based on the information described in the datasheets.

### 2.2 Context

In the datasheet, it shows clearly what the block diagram is. How does the chip transform from one mode to others. It also illustrates what happens during different modes. Figure 5 shows the state diagram of TJA1020. It describes how to change different operation mode.

In the manual measurement, the design is based on this diagram, set the pin high or low and check whether the waveform and voltage is correct.

Last but not least, the characteristics for each pin which should be measured in this assignment. The MIN/TYP/MAX value is available, the condition is also accessible. For some measurements which the student does not familiar with, the company mentor will describe the measurement knowledge and gives advice.

The DELPHI language guide tells how to use and program in DELPHI5. Some videos also help understand different VCL in DELPHI. Company mentor give some help as well. The VCL
Hardwareform/ADC/DAC/Power I/O is made by NXP, How to use them in the programming is also a challenge. The subject for LIN gives an overview of LIN transceiver and how LIN bus works. The company mentor also gives some courses to describe the LIN transceiver and how every pin works according to the block diagram.

There are many components used in the schematic design, the knowledge about analog circuit is required. LED has been mentioned in chapter1.1 used to indicate users. MOSFET is widely used in the assignment, the background knowledge should be command. Figure 6 is a basic circuit for pin select. Both Pina and Pinb used the same schematic. It is controlled by the optocoupler. If Pina is used, pin 3 and 4 for the optocoupler should conduct, it means that PMOS should conduct while NMOS should not. DAC is used to set Pin Ha Lb out. In this function, for PMOS, Vsource should be higher than Vgate, so DAC will give 0V which made PMOS conducted and NMOS not conducted.

2.3 Presupposition

Before setting up the theoretical framework, some knowledge needed to be acquired. The first one is the basic knowledge about the electric circuit. Despite some knowledge which is new for the student, the schematic design should be understood. The second one is programming, Delphi 5 is a new programming tools, but with the knowledge of C language, it is convenient to learn this new code fast. The third one is measurement devices. Student should understand how to use every measurement devices to measure and set values.
The last one is to understand how to set a semi-automatic test setup. Before doing the thesis, student should have a global idea about what need to be tested.

2.4 Concepts

The theoretical framework has been separated into several parts. Since the TJA102X family have 8 chips, it is efficient to complete them one by one because the designing method is almost the same. For example, TJA1020.

Firstly, the datasheet of TJA1020 should be read carefully.

- The knowledge about how TJA1020 works.
- The knowledge about how LIN transceiver works.
- The knowledge about how to develop a semi-automatic test setup.
- Knowledge on electric circuit.
- Knowledge of electronics.
Secondly, according to the datasheet, a lot of characteristics should be measured during the test. How to measure these values should be understood. The knowledge about how different pins work in different modes.

- Conversion mode in four operation modes.
- Knowledge on electric circuit.
- Knowledge of electronics.
- Mathematics.

Thirdly, Delphi5 is used as a tool for programming in this thesis. How to use it to write code is necessary to be mastered. The knowledge about how to use Delphi5.

- Knowledge of Pascal language.
- Knowledge about Delphi5 programming.

Fourthly, the test can be separated into manual measurement and automatic measurement. Different devices and measuring methods would be used.

- The knowledge about the measurement devices.
- Knowledge of setting measurement devices in the lab.
- Knowledge of the waveform of the oscilloscope.
- Knowledge of curvetracer.
- Knowledge of test mode.
- Knowledge on electric circuit.

Finally, debugging and improving is required as well. They can be done in both software part and hardware part.

- Knowledge on electric circuit.
- Knowledge on PCB design.
- Knowledge about Delphi5 programming.
- Knowledge on components selection.
- Knowledge on drilling.
3. Method and materials

3.1 Introduction of Structure Design Methodology (SDM)

In this project, SDM is used to do the assignment and write report. Structure design methodology is promoted as a means of improving the management and control of the software development process, structuring and simplifying the process, and standardizing the development process and product by specifying activities to be done and techniques to be used. The method is a waterfall model divided in phases that have a clear start and end. (Sciences, Structure Design Methodology, 2013)

![Figure 8 Structure design methodologies](image)

As what is showed in the figure 8, there are several processes. These processes are very useful to solve the problem. With problem analysis phase, Bird’s Eye View (BEV) is a visual representation of the problem within its environment without much detail. Also it can be used as intermediary when talking to a costumer. End Product Definition (EPD) defines the usability of the system for the end-user. The main features are described in the EPD.

There are 2 parts in the problem definition phase. One is external problem definition and another is internal problem definition. In the external problem definition, the interfaces of the design problems will be defined. The next step is to zoom in the problem and make an internal overview of the problem. Then it goes to the design part: functional design and physical design. In the functional
design, we describes the system in terms of WHAT it does, not how and focused on the USABILITY, not operability. Jargon is not allowed in the functional design. In the physical design, we describes the system in terms of HOW, not what and focuses on OPERABILITY, not usability. JARGON language is allowed in the physical design. (Russo N. L., May 21 24, 1995)

3.2 Justification
There are three methods students in HZ and SMU have already learned. Eggert method, Delft method and SDM. In this thesis project, SDM is more suitable for software design.

The BEV can clearly indicate what the design is. The external overview and internal overview show the function and operation of the system. By using SDM, the assignment can be separated in small blocks, and then step by step, finish the assignment in the end.

Regarding to this assignment. In problem analysis phase, the BEV will give an idea about the whole system, the relationship between the software and hardware. In EPD, the specification of each pin can be found out. According to the schematic design and the datasheet, which components will be connected and used can be viewed directly.

In External problem definition phase, an external overview defines the signal value and type which would transfer among the computer, measurement devices and the test board. In internal problem definition phase, internal overview of the software test setup gives a clear view of the structure. Each block in the internal overview diagram would be programmed and applied in different functions for different pins.

There would be no contract part in this project. The contents of functional design phase showed the design methodology of how to measure the specific characteristic in the datasheet. Which should be used and should not be used would be displayed in the flowchart according to the operation mode.

In physical design, there was code and the interface which has been designed. The code showed the function of how to measure and test the characteristic. The interface was designed based on the flowchart.

In the realization phase, the result can be found. For the improvement of the software and hardware, they would be illustrated in recommendation part.
4. Results

In this section, all of the results got from the design process will be shown here. As stated before in Chapter 3 method and materials, the outcomes will be discussed in the order of structure design methodology. Small conclusions will be drawn at the end of each design phase related to the sub-questions needed to be answered.

• Design brief
Design and build a semi automatic test setup for TJA102x to check and test the data is the goal we want to reach. We have to specify the chip on each pin at different mode. Curvetracer and test mode would also be part of the test in this thesis. So to conclude, we have to develop the software for the semi-automatic test set up to verify the TJA102X family specifications, and implement it using Delphi 5.

• Function of the product
The software is made to test the data and curve of the TJA102X family to check whether the chip has problems or not. The users can easily do measurements both manually and automatically. They can set the value of SMU voltage and DAC value in the manual part. In the test of curvetracer, they can first select the type of chips, write the switch number and value in the table. Then the measurement will be finished automatically.

• Design parts
✓ Design the manual measurement interface for the test setup. It should be user friendly.
✓ Design the automatic measurement interface. This includes all the static characteristics in the datasheet. The test can be done one by one, it can also be tested automatically. The results would be saved in the Excel.
✓ Design the curvetracer system to test whether the chip has Electrostatic discharge problems.
✓ Design the test mode interface to measure some special characteristics

4.1 Problem analysis phase
4.1.1 Bird’s Eye View

Figure 9 is the Bird’s Eye View of the whole testing system. It shows the relationship among the project (Software test setup for TJA102x family), the hardware test board and other measurement devices which are SMU (source measurement unit), DAC/ADC (digital to analog converter/analog to digital converter) Power I/O, Waveform generator, Keithley 2000 Multimeter and Oscilloscope. Besides, the software tool DELPHI 5, used to make the project.

Power I/O is an open collector which can connect the pin to the ground.

Source measurement unit (SMU) can be used to supply and measure voltage and current of each pin. It can be set by the software made in DELPHI. For example, if the user set SMU1 (two SMU are
used in the test, one is for the Battery supply, the other one is used on other pins for measurement), with 12V source voltage and 10mA for clamp current, voltage source. Then, on the screen of the SMU1, 12V will be displayed.

Keithley 2000 Multimeter is used to measure the voltage, current and resistance of the pin.
DAC is used to supply the voltage which ranges from 0V to 5V to make the pin high or low. It should be disabled if SMU is also used for that pin at the same time, and vice versa. Otherwise, if the SMU voltage is higher than 5V, DAC will break.

ADC is used to measure the voltage which ranges from 0V to 5V. ADC can be measured and displayed on the software panel. In RXD pin, there is an edit box which will show ADC value.

Oscilloscope is used to check the voltage curve of each pin. There are four channels supplied on oscilloscope, one connects with the waveform generator which is used to generate a square wave in manual measurement, other channels connect to the hardware test board. When the user wants to do the test, he should first generate a square wave on CH3 (TXD requires the square wave, in TJA1020, TXD is pin3), and then connect the pin to be tested, CH4 is not used in TJA1020, so user can connect CH4 to TXD directly, during the test, the curve for CH3 and CH4 should be the same.
Hardware board is to transmit the data among all the external devices and the test setup just like a transfer station. Different types of chips can be plugged into the same daughter board. For instance, if the user wants to test TJA1020, he just plugs TJA1020 chip in the board, runs the software and selects TJA1020, then he can start testing.

Program designer is the student who works on this thesis project. He should first understand the assignment, the design method and understand how to do the test. Then he will uses DELPHI 5 to make the application and writes the program code. After that he will run the software to check, debug and improve the program. Finally, he will writes a thesis report for this assignment.

Tester will use the software which is programmed by Delphi5 to control the whole test setup. Through the test setup, the suitable measurements tool (SMU/ADC/DAC/Oscilloscope/Power I/O) will be triggered and send back the data to the software so that the testing result can be read easily and clearly by the tester.

4.1.2 End Product Definition
This is the main menu interface of the Software test setup. Users can set the maintenance, where he can select the type of SMU he used. He can also choose the type of the chips which he wants to test.
Figure 11 is the first GUI of the chip TJA1020 which is used for manual measurement. Since the whole family is a series, other chips’ GUI of manual measurement are nearly the same. Only has difference in some pins. For TJA1020, there are 8 pins.

RXD pin is the receive data output.

NSLP pin is sleep control input, controls inhibit output; resets wake-up source flag on TXD and wake-up request on RXD.

NWAKE pin is local wake-up input.

TXD pin is transmit data input.

GND pin is the ground.

LIN pin is LIN bus line input/output.

BAT pin is battery supply.

INH pin is battery related inhibit output for controlling an external voltage regulator.

The first checkbox is TXD, which is the transmit data input. The channel wave, DAC and SMU should be set. In the schematic design, TXD used the circuit for Low power pin which showed in Figure 12.

This is the schematic circuit for Low power pin. There are two types of program code mainly used. The first one is PowerIO11.output_on/off (integer). PowerIO is the driver, output_on/off means whether the optocoupler will conduct. If output_on, then the LED conducted which made the switch on, the optocoupler would conduct. The integer related to which function is used, it has been supplied in the top level of the schematic design (Appendices J).
The second is DAC1.sendvolts (integer, voltage). DAC1 is the VCL used to control DAC. The integer is the number related to the DAC for the pin, it can be set 5V for make pin High and set 0V to make it Low.

The function for other blocks and more detailed explanation will be illustrated in appendices A.

The specification of the pin can be found out according to the BEV and EPD. Different devices connected to the test board, the relationship was displayed obviously.

4.2 External Overview

This is an external overview of the software test setup. It shows the connection among the computer, test board and other external devices.

It is necessary to connect the measurement devices to the test board (each pin of the chip). An adapted board is used to connect the chips and the board. It makes the test convenient, once you want to test a chip, just put the chip in the adapted board then connect it to the board.
The test board hardware consist of two parts, one is the daughter which the previous student made for testing TJA102X family. The second part is the motherboard. Motherboard acts a bridge, which is used to connect DAC/ADC/SMU/Power I/O, the daughterboard and the computer.

According to the BEV, there are 4 input arrows and 6 output arrows. In this external overview, there is a match.

4.3 Internal overview

Figure 14 is an internal Overview of the software test setup. The test setup is made to measure the working condition of each pin such as current and voltage. DELPHI will be used for programming. In the graphic user interface of the test setup, users can choose and set the function for each pin.

DAC and ADC
DAC is used for input while ADC is for output. When the input pin such as TXD and NSLEEP is used, to let the pin high, a high signal which means 5V should be given to DAC. ADC is for output pin, like RXD. To measure the RXD value, the switch for ADC should be switched on at that time.
SMU
SMU is the power supply which can be used to set the voltage and current. To use it, the switch for SMU should be switched on firstly, then use the GUI to set the voltage and current I need. Then on the screen of the SMU, the value for the pin will be displayed.

Channel
To see the waveform in oscilloscope, the switch for the Channel should be put on. The oscilloscope should connect to the board and the waveform generator.

LIN
Lin is the transceiver of the chips, there are three types of termination 500/650/1000Ω as the load to change the slope of LIN can be selected, which will also influence the output RXD waveform.

PUVccon
This is for the pull-up resistor. For pins like RXD, in case the microcontroller port pin does not provide an integrated pull-up, an external pull-up resistor connected to the microcontroller supply voltage Vcc is required. So the switch should be put on when RXD is used.

There are 4 input arrows and 6 output arrows in the internal overview. It matches with BEV and External overview.

In EPD, figure 12 has already showed what every pin referred to in the Altium board. The external and internal overview also mentioned different function like DAC/ADC/SMU, how they sent data and the value used to control them.

4.4 Functional design

In the functional design, the flowchart of the manual measurement design would be displayed, the flowchart for some measurement in automatic measurement design would also be showed.

4.4.1 Manual measurement

Figure 15 is the flowchart based on the State diagram in datasheet (Figure 5), it shows how TJA1020 go from one mode to other mode. Pin = 1 means the DAC set on that pin is 5 V while Pin = 0 means set DAC 0V.

For TJA1021, it does not have low slope mode. Instead, when give power to the battery pin, it goes to power-on mode, then users can make SLP_N high or low to go to other mode.
Each pin has their own flowchart which will be shown in the appendices B.
4.4.2 Automatic measurement

In the datasheet, many characteristics should be tested. Here, the characteristics of the battery pin were given as an example.

- Supply $I_{\text{BAT}}$

Sleep mode ($V_{\text{LIN}}=V_{\text{BAT}}$; $V_{\text{NWAKE}}=V_{\text{BAT}}$; $V_{\text{TXD}}=0V$; $V_{\text{NSLP}}=0V$) (Datasheet requirements)
This function can be used for both TJA1020 and TJA1021. The difference is to go to the sleep mode, TJA1020 should set TXD high and pull up resistor on RXD while for TJA1020, TXD and RXD is not used to go the sleep mode. The code can be found in program code in physical design.

Standby mode; bus recessive ($V_{INH}=V_{BAT}$; $V_{LIN}=V_{BAT}$; $V_{NWAKE}=V_{BAT}$; $V_{TXD}=0V$; $V_{NSLP}=0V$)
This function can be used for both TJA1020 and TJA1021. The difference is same as the sleep mode because to go the stand mode, it is necessary to go to the sleep mode first, then make NWAKE low or LIN low.

Low slope mode; bus recessive ($V_{INH}=V_{BAT}$; $V_{LIN}=V_{BAT}$; $V_{NWAKE}=V_{BAT}$; $V_{TXD}=5V$; $V_{NSLP}=5V$)
For TJA1020, to go to the sleep mode, user should let TXD low and pull up resistor on RXD, then make NSLP high.
For TJA1021, it does not have low slope mode.

Normal slope mode; bus recessive ($V_{INH}=V_{BAT}$; $V_{LIN}=V_{BAT}$; $V_{NWAKE}=V_{BAT}$; $V_{TXD}=5V$; $V_{NSLP}=5V$)
For TJA1020, to go to normal slope mode, user should let TXD high and pull up resistor on RXD, then make NSLP high.
For TJA1021, it is called normal mode. After set the battery voltage typically 12 V, then make SLP_N high, the chip will go to normal mode.
Power on reset

LOW - level power-on reset threshold voltage
For TJA1021, it has a power on mode which come to automatically when the battery switch on. This function is used to check when the battery pin will go from switch on to switch off. This can be checked by measuring the current on battery pin because when it switches off, the current will become lower than 10uA while it is over 150uA when switch on. The battery voltage was swept from 5V to 0V, because in the datasheet, the MAX voltage is 3.9V. The voltage drops 0.05V every time.

HIGH - level power-on reset threshold voltage
This is the same as LOW – level. The voltage was swept from 0V to 5V. With the increase of 0.05V every time.

For other pins the flowchart will be shown in the appendices C.

4.5 Physical Design

In the physical design, the interface which has been made will be displayed, the programming code will also showed in this part. They are related to the flowchart which has been demonstrated in the functional design.

4.5.1 Manual measurement
This is the interface for TJA1020 manual measurement which has been showed in the End product definition. Each pin has its own box and function. If the box is checked then the components related to that box will be connected in the hardware. The function is based on the schematic design and the datasheet. In manual measurement, users can set the mode he would like to do the test, this can be realized by clicked the box according to the flowchart figure 15, or, he can click the box in the mode function, where he can select the operation mode and run directly.

Here, TXD will be used as an example for explaining how the function box works.

There are four checkbox in TXD. CH3 is used for displaying waveform on the oscilloscope. The main programming code is given as followed.

If CH3toTXD.checked then begin
  powerIO11.output_off (CH3toNWAKE);
//powerIO11.output_off is a Visual component Library in DELPHI5 which is designed by NXP. It is controlled by the device of PowerIO. Here output_off (CH3 to NWAKE) means the CH3 on NWAKE pin should be put off. This is because TXD has already used CH3, in order to have the correct waveform with no influence on NWAKE, the switch for CH3 on NWAKE should be put off.
  CH3NWAKE.enabled:=false; // this means the box of CH3.NWAKE was disabled.
  powerIO11.output_on(CH3toTXD1); // this means the switch of CH3.TXD was put on.

Else begin
    powerIO11.output_off(CH3toTXD);
    CH3NWAKE.enabled:=True;
    // after the test, the system had to reset to the original status, the switch which has been put on
    should be put off, the box which has been disabled has to be enabled in order to do other
    measurement.
end;

DAC and H/L are the box to set DAC value in order to make TXD High or Low. Here High means DAC
is 5V while 0V represent for TXD Low.

If DACTXD.checked then begin
    PowerIO11.output_on(Pin4_10a_DACon);
    // this program code means the switch of DAC to TXD has been put on, therefore, DAC is connected
to TXD. Figure 12 has already showed the schematic design of this part. When DAC switched on, the
LED will light which switch on the switch in the optocoupler, the DAC will connect in the circuit.
    SMU2TXD.enabled:=false;
    // this program code is used to disabled SMU on TXD. Both SMU and DAC can be used to set voltage
on TXD. However, DAC cannot be set higher than 5V otherwise it will break. In order to protect the
DAC, the SMU on the pin should be disabled when DAC is used.
If HLTXD.checked then
    DAC1.sendvolt(pin4_pin10a_DAC, 5) // this means set DAC 5V.
else
    DAC1.sendvolt(pin4_pin10a_DAC, 0) // this means set DAC 0V. When the test start, all DAC is
floating, therefore, when the test has finished, they should be set LOW to protect the device.

SMU is the last box in this function. It is used to set the voltage and clamping current during Voltage
source and current with clamping voltage during current source.

If SMU2TXD.checked then
    begin
        powerIO11.output_on(Pin4_10a_SMU2on);
        SMU2RXD.enabled:=false;
        SMU2LIN.enabled:=false;
        SMU2NSLP.enabled:=false;
        SMU2NWAKE.enabled:=false;
        SMU2BAT.Enabled:=false;
        SMU2INH.Enabled:=false;
        DACTXD.enabled:=false;
        // this program code means the switch of SMU to TXD has been put on. From Figure 11, we can find
despite the battery pin has SMU1, other pins all use SMU2. Therefore, in the function for SMU, the
switch connect to other Pins except the pin to be measured should be disabled.
4.5.2 Automatic measurement

- Interface
- TJA1020

Figure 16 is the interface for TJA1020 automatic measurement. Each pin has its own block and characteristic. The characteristic to be measured is based on the datasheet. If the box is checked then the components related to that box will be connected in the hardware. The result of that function will be displayed. Compared the parameter with the data in the datasheet, it is easy to find out whether the pin has problems or not.

Other type of interface will be showed in appendices F.

- Program code
In chapter 4.4.2 functional design automatic measurement, there is flowchart for battery pin measurement. Here, the battery current sleep mode was used as an example.

```
setSMUvoltage(1,GPIBadres_SMU1,VBattery,10e-3);
```

//this is the function to set the SMU voltage. In the brackets there are four parameters. The first number 1 means the SMU number, there are two SMU used in the thesis, in TJA1020, SMU1 is used to set the battery voltage. The second value is GPIB address, it is set in the SMU device before testing. The third value VBattery is defined as a variable which will be set by the users, The fourth
value is the clamping current. Therefore, this code means setting SMU1 Vbattery V with 1mA clamping current.

```
powerIO11.output_on (Pin15_9a_SMU1on); // close the switch to connect SMU1 in the circuit.
DAC1.sendvolt (pin4_pin10a_DAC, 0);
PowerIO11.output_on (Pin4_10a_DACon);
// Pin 4 related to TXD in the sleep mode. TXD should be Low, therefore, DAC is used to set 0V on TXD to make it Low.
DAC1.sendvolt (pin2_pin8_DAC, 5);
PowerIO11.output_on (Pin2_8_DACon);
DAC1.sendvolt (pin2_pin8_DAC, 0);
// Pin 2 related to NSLP pin. In the sleep mode, NSLP should be 0 after 1->0 which is mentioned in the state diagram Figure 5. Therefore, DAC is used to first set NSLP high, then switch to low. Now the chip is in the sleep mode.
value:=measureSMUcurrent(1,GPIBadres_SMU1)*1e6;
// this is the function to measure the current displayed in the screen of SMU. The two values in the brackets means SMU number and GPIB number respectively. It is multiplied by 1e6 because the value measured is A, however, in the datasheet, it should be uA.
lsleep.caption:=floattostr(value)+' uA';
// the left part is the caption of the result. Floattostr() is a function used in Delphi5 to change a float value to string. The caption is a string, therefore, the right part should also be a string to make a match.
setSMUvoltage(1,GPIBadres_SMU1,0,10e-3);
// After the test, the SMU should be set to 0V, all other switches should be switched off.
```

In the automatic measurement, the typical battery voltage value is 12V according to the datasheet.

```
supply_current_on_batteryslmode (12);
// this is the code used to call the function showed above, the 12 in the bracket means VBattery equals to 12V which will be used in setSMUvoltage (1,GPIBadres_SMU1,VBattery,10e-3);
```

Comparing the result showed in lsleep.caption with the parameter in the datasheet, it's clearly to find out whether the pin has problems or not.

Other program code will be showed in appendices G.

### 4.6 Realization phase

The realization phase can be divided into three parts. The results of automatic measurement will be displayed in picture and table. Furthermore, it will also illustrate how to do test mode and curvetracer.
Figure 17 TJA1020 automatic measurement results

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Result</th>
<th>Typical Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Battery supply current</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Battery current sleep mode</td>
<td>2.9 uA</td>
<td>3 uA</td>
</tr>
<tr>
<td>Standby mode recessive</td>
<td>395 uA</td>
<td>400 uA</td>
</tr>
<tr>
<td>Standby mode dominant</td>
<td>948.3 uA</td>
<td>900 uA</td>
</tr>
<tr>
<td>Normal slope mode recessive</td>
<td>470.6 uA</td>
<td>400 uA</td>
</tr>
<tr>
<td>Normal slope mode dominant</td>
<td>1.0242 mA</td>
<td>3.5 mA</td>
</tr>
<tr>
<td>Low slope mode recessive</td>
<td>382.7 uA</td>
<td>400 uA</td>
</tr>
<tr>
<td>Low slope mode dominant</td>
<td>0.9365 mA</td>
<td>3.5 mA</td>
</tr>
<tr>
<td>TXD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low level input voltage</td>
<td>1.2 V</td>
<td>0.8 - 2V</td>
</tr>
<tr>
<td>High level input voltage</td>
<td>1.6 V</td>
<td>0.8 - 2V</td>
</tr>
<tr>
<td>TXD hysteresis voltage</td>
<td>0.4 V</td>
<td>0.03 - 0.5V</td>
</tr>
<tr>
<td>TXD pull down resistance</td>
<td>294.117 k</td>
<td>350 k</td>
</tr>
<tr>
<td>Low level input current</td>
<td>-2.92 uA</td>
<td>0 uA</td>
</tr>
<tr>
<td>Low level output current</td>
<td>2.887 mA</td>
<td>3 mA</td>
</tr>
</tbody>
</table>
Comparing the result with the parameters in the datasheet. Since the battery voltage is set as the
typical value 12V, the result should be almost the same as the typical value which is in the middle of
the table.
However, for the test while LIN bus is dominant, the value is a little bit different. To go to the
dominant time, LIN pin should be set 0V, TXD should be 0V, but when TXD go from high to low, the
battery current changes rapidly and goes back. Because if the propagation time for TXD low is long,
the dominant timer will not go to the transmitter. This also happened in some other characteristic
test. When the chip should be set to dominant time, it is hard to get the correct result because SMU
cannot react to the changes in the short period.

In addition, another two tasks should be added in this assignment, they are test mode and
curvetrace.

4.6.1 Test mode

TJA1020 does not have a test mode, therefore, TJA1021 is used to give an example.
For TJA1021, a test mode measurement is required. In order to go to the test mode, the knowledge and methodology should be learned from the document (DFT document of TJA1021).

To go to test mode, it should follow the steps below:

1. Power on the device
2. Set voltage NSLP lower than V_tcb_en (typically -8V)
3. Shift in the test mode bits (5 bits for one test mode) via RXD (tcb_tdi) while running clock signals at TXD (tcb_clk)
4. Set voltage NSLP lower than V_tcb_hold (typically -11V)
5. Do corresponding tests related to this test mode
6. If another test mode wants to be set, then set voltage NSLP higher than V_tcb_hold (typically -11V) and lower than V_tcb_en (typically -8V)
7. Shift in another 5 test mode bits via RXD and TXD
8. Do corresponding tests related to this test mode
9. Repeat step 6~9 if necessary
tcb_clk is TXD input. In the schematic design, there is a clock line (Figure 19).

In TJA1021, pin4 is TXD, therefore, Clock 4 is used in the test. Clock line is used to control the duty cycle of RXD. Since RXD should be shift in 5 bits, 5 square wave is required from TXD.

In the program, DAC1.sendvolt (pin4_clock, 5); which means set TXD high while DAC1.sendvolt (pin4_clock, 0); which means set TXD low.

tcb_tdi is equal to RXD. The schematic design is called data line. The design method is same as Clock line.

The program for one bit is like DAC1.sendvolt (pin4_clock, 5); DAC1.sendvolt (pin1_data, 5); sleep (10); DAC1.sendvolt (pin4_clock, 5); sleep (20);
Whether data line should be high or low is based on the test mode bits which is mentioned in the document.

To go to the test mode, NSLP pin should be set to negative voltage. It is controlled by Voltage controller in schematic design (Figure 20).

In figure 20, for the amplifier, the gain equals to -3K/1K=-3, therefore, the output voltage will be three times of the input voltage. In test mode, -12V is required on NSLP pin, the input voltage should be 4V set by DAC. DCDC converter was used to supply the voltage for activate the amplifier. Regarding to the current limiter, the triode conducted when the current through the circuit was low, if the current became high, the left triode would limit the current because at this time the voltage on the base of the left triode equals to Vresistor+Vrighttriode. If the result increase, then Vbe for the left triode would decrease which limited the current and protect the circuit.

OSC is the first test in test mode. Test introduction:
- Oscillator on/off via TXD pin (TXD = 1 -> osc on)
- Monitor the output signal of the oscillator via RXD pin. Check the frequency [256Hz].
- Check the start up & turn off behavior if possible.
procedure TTJA1021test.OSCMClick(Sender: TObject);
begin
if OSCM.Checked then begin
  powerIO11.output_on (Pin2_testvoltageOn);
  // this is used to close the switch to use voltage controller circuit to set NSLP negative voltage
  OSC_clk; // this is the function for OSC test to shift the bits in RXD.
end;

Procedure OSC_clk;
Var j: integer; i: integer;
begin
with TJA1021test do begin
  i:=0;
  DAC1.sendvolt (Clock_out, 5);
  DAC1.sendvolt (data_out, 5);
  for j:=0 to 4 do begin
    i:=i+1;
    if i=1 then DAC1.sendvolt(Pin1_data,5);
    if i=2 then DAC1.sendvolt(Pin1_data,0);
    if i=3 then DAC1.sendvolt(Pin1_data,0);
    if i=4 then DAC1.sendvolt(Pin1_data,0);
    if i=5 then DAC1.sendvolt(Pin1_data,5);
    DAC1.sendvolt(Pin7_clock,5);
    sleep(20);
    DAC1.sendvolt(Pin7_clock,0);
    sleep(10);
  end;
end;

//This procedure is for OSC shift in. In the datasheet, the code for this test is 10001. After set Clock out and data out 5V to connect both Clockline and Dataline in the circuit, the testing bits will be shifted via RXD pin which is dataline. Clockline should be shifted via TXD pin. The clockline is a square wave.

The frequency can be checked in Oscilloscope.

The details of the parameters to be tested will be showed in appendices H.

4.6.2 Curvetrace

![Curvetrace interface](image)

Figure 21 Curvetracer
Curvetrace measurement is used for testing Electrostatic Discharge (ESD). ESD is the sudden flow of electricity between two electrically charged objects caused by contact, an electrical short, or dielectric breakdown. A buildup of static electricity can be caused by tribocharging or by electrostatic induction. (Liang, 2014)

ESD is a big problem which customers usually focused on because it can cause problems like equipment failure or electromagnetic interference during the test; it may break the components or make it aging; sometimes it may also cause horrible accidents.

For this project, the ESD protection is required in the test.

In the curvetrace test, every pin expect the GND pin should be tested and compared with the reference. The first test is to check whether hardware board has any components which may influence the results. During the test, the chip does not need to be plugged in. User or tester can run the program on the pin, the result should be the same as Figure 22.

The current should be 0mA while the voltage is sweeping from -5V to +5V. This means there is no other diode or resistor connects to the pin. Then the chip can be plugged in and do the test again. If the curve is the same as the reference shows, then the pin is protected from ESD problem.

Usually, the diode and resistor is used to protect the pin. A diode is the most basic semiconductor device. It is made from a p-type and an n-type junction and has two terminals: an anode at the p-type and a cathode at the n-type. When a large enough voltage is applied from the cathode to the anode (reverse biasing), the diode enters its breakdown region and, in theory, can conduct an infinite amount of current at zero resistance. A voltage applied in the other direction (forward biasing) causes the diode to enter its forward-conducting region. These diodes can respond to a high ESD voltage very quickly and clamp thousands of volts to just tens of volts in a matter of nanoseconds by shunting the ESD current to ground. (Liang, 2014)

For more details will be showed in appendices I.
5 Discussion
The discussion can be separated into 3 parts, manual measurement, automatic measurement and other test tasks.

For the manual measurement, the result is reliable. A function box for mode selection was available. This enabled the test more convenient. When the users would like to test sleep mode, he could click sleep mode but not click every function for every box. The waveform is showed as expected.

For the automatic measurement, there was some data which cannot be tested directly, especially when the data should be tested in dominant time. This is because that if the propagation time for TXD low is long, the dominant timer will not go to the transmitter. The chip will go to normal mode so the data tested is wrong.

For other test tasks, test mode could be used to test some special specifications. To go to the test mode, -12V should be applied on NSLP pin, therefore, the voltage controller circuit was designed for this application. However, the design had problems. The -12V also go through the circuit to the optocoupler which broke DAC board. Meanwhile, the chip for the DCDC converter did not work as expected. In the recommendation part, it will be illustrated what can be improved later.

For Curvetracer, first test the pin while set SMU voltage at the range of -2V to 5V. Though there was error at the beginning caused by the design in the schematic part, after being fixed, the result was as expected. However, it has problems when the voltage range was changed to 15V. The optocoupler on the Clock line clamped the voltage at 5.6V. Therefore, the waveform was wrong, this should also be improved in the next schematic design.

5.1 Answer of sub questions
✓ What is LIN Transceiver (TJA102X family)?
TJA102X family is a series of Local Interconnect Network (LIN) transceiver including TJA1020/21/22/24/27/28/29 and UJA1018.
LIN is a bus system which is aimed mainly at the automotive industry. It has been designed to provide a low-cost networking solution for low-speed, low-risk applications as an alternative to the high speed, high integrity, higher cost Control Area Network (CAN) bus which is required for engine management, braking and safety features.

✓ What is the datasheet of TJA102X family?
The datasheet of TJA102X can be separated into three parts. The first datasheet is product specification which illustrated the background knowledge of the chip and gave characteristic of the pin. The second datasheet is application note which gave information about the pin. The third one is for test mode, which told how to do the test mode and the code of different test.

✓ What is automatic test set-up?
The automatic test set-up is a software interface which can be operated by the users to check and find out the pin characteristics.

✓ What should be measured in the test?
Pin static characteristics, test mode results and curvetracer.
How to use the measurement tools?
The SMU/DAC/ADC/PUVcc can be controlled by the software which has been designed. The oscilloscope and waveform generator can be set by the users.

What is the specification for each pin of the test set-up?
- RXD pin is the receive data output.
- NSLP pin is sleep control input, controls inhibit output; resets wake-up source flag on TXD and wake-up request on RXD.
- NWAKE pin is local wake-up input.
- TXD pin is transmit data input.
- GND pin is the ground.
- LIN pin is LIN bus line input/output.
- BAT pin is battery supply.
- INH pin is battery related inhibit output for controlling an external voltage regulator.

What is the relationship between each pin?
TXD transmit the data to LIN pin, and RXD will receive the data and send the output to the microcontroller. NSLP is used to wake up the chip to go to different mode.

How to use the VCL designed by NXP to set values on measurement tool?
The VCL designed by NXP is used to control DAC/ADC/PUVcc and SMU. According to the program code and the example given by the company mentor, they can be applied in the design.

What is the function block mentioned in Internal Problem Definition phase (S)?
There are six function block in the Internal Problem Definition phase. DAC/ADC/SMU/PUVcc/Channel is the block for different measurement devices. The LIN block is used for LIN to change LIN termination.

What is the function for pins on the chip?
SMU and channels are applied on every pins. SMU is used to set the voltage and clamping current while Channels are used to show the waveform of that pin during the test. For input pin, DAC is used to make pin High or Low. For the output pin, ADC is used to show the output voltage of the pin. For LIN pin, there is block for LIN termination to change different load.

What does every pin refer to in the Altium board?
In the Altium board, every pin has a number related. According to the number, Delphi5 can do program for it.

Which function should be enabled or disabled during the measurement?
During the measurement, while SMU is used on a pin, the DAC on that pin should be disabled and vice versa. Meanwhile, if SMU2 is used on a pin, then SMU2 for other pin should also be disabled.

Which pin or switch should be on at the beginning of the test? What should be reset?
At the beginning, the pin select should be done to connect the right circuit for the chip. After the test, all switches should be switch off and DAC/SMU should be reset to 0V.
What kind of waveform in (LIN/TXD/RXD) pin is expected on the oscilloscope?
According to the diagram in the datasheet, the waveform on the oscilloscope should be the same. TXD is square wave while LIN and RXD has a little delay and slope.

How to sweep the voltage or current in the automatic measurement?
A program function designed by the company mentor is used to sweep the voltage and current for SMU in the measurement.

If the measured data is different from the datasheet, how to solve the problem?
First, checking the program code, run the program step by step to find out where the problem happened. If the program has no problems, then check the schematic design of PCB to analyze whether there is wrong connection in the circuit. Finally, test the components value with the datasheet to check whether the components has broken or not.

What can be improved in the code writing?
Some procedure and functions can be designed to simplify the program. The structure of the program can also be changed to make it more readable.

5.2 Answer of main question

- How to develop the software for the semi-automatic test set up to verify the TJA102X family specifications, and implement it using Delphi 5?

Firstly, learning and understanding the datasheet of the TJA102X family to have a global idea of how the chip works. Using measurement tools to check the hardware board and getting familiar with the circuit.

Secondly, drawing flowchart for every pin of the chip, what will be used and how to measure the data should be found out.

Thirdly, writing the code and testing the program according to the datasheet.

Fourthly, laying out the problems occurred, giving solution and advice.

Finally, solving the problem and testing again.

With these steps and the answer of the sub questions, the main questions can be answered completely.
6 Conclusion

For this thesis, the assignment is to develop a semi automatic test setup for TJA102X family.

First of all, the background knowledge of TJA102X family should be mastered. The program is based on the hardware schematic design which the previous student made. Therefore, it is necessary to understand the schematic design. Different pins have different function and different components, how to transmit the data and what will be used in the design should be found out at the beginning.

The software design is separated into two parts: manual measurement and automatic measurement. In the manual measurement, after learning the block diagram of the chip, the schematic design of the hardware board should be understood. How to use the external measurement device should also be found out.

In the automatic measurement, the characteristic need to be tested is based on the datasheet, the methodology to test these parameters should be thought and designed. It is based on the understanding of the chip block diagram and application note. The external measurement devices are also used in the automatic measurement. SMU can be used to sweep the voltage from high to low and vice versa. Different types of chips have different characteristics to be measured. Despite TJA1020, other chips have test mode.

The test mode is used to test some specifications such as the frequency of the oscillator inside the IC. For some characteristics in the automatic measurement, it is not possible to measure in the normal mode because the propagation time is too short that the SMU and Oscilloscope could not catch it. With the help of test mode, it is possible to do the measurement. NSLP pin would be supplied by the negative voltage which is controlled by the circuit of voltage controller schematic. Data line was created to send the typical TXD value to the pin. Clock line was made to send the clock signal to RXD pin. For instance, supposed that test mode is essential to measure the frequency of the oscillator. According to the required signal, test mode was entered. Then, the switch of RXD would be connected to the oscillator instead of the receiver. In that case, the specification from the oscillator can be measured.

Curvetrace is also part of the project. It is used to check whether the pin has ESD problems or not. The test is to use the software program to check the curve on the pin and compare it with the reference to find out whether the chip is qualified.

Delphi 5 is used to do this project. It took some time to learn this new software tools. However, it was a user friendly system and the VCL was helpful and understandable. In this project, NXP also designed the VCL especially for the test devices DAC/ADC/PowerIO, which helped a lot during programming.

Last but not least, the schematic design circuit should be mastered. Learning how to use Altium board to do cross probe to find out the chip related to the pin was a great advantage for debugging and testing. Learning how to find out the problems and give solution for the circuit also improved the results and useful for the next design.
7 Recommendation

For further research:
Figure 23 is the schematic design for Data line, the resistor is required between the gate and the ground. 2N7002 is an NMOS, since OUT31 connect to DAC, it is floating when power on the device. If there is no resistor, the NMOS may conduct and some unexpected damage may happen.

In the test mode, NSLP pin requires a negative voltage supply connect to data1, this will break the optocoupler. If data1 in figure 20 received a negative voltage, which would conduct the circuit in the optocoupler, then the short would happen in the circuit which broke DAC. Therefore, a switch is also required, in the test mode, the circuit should be switched off to protect the device.

In TJA1020/1021, pin3 is NWAKE pin, the voltage of NWAKE pin should be the same as battery pin. However, since it is connect to one of the schematic design as the picture shows, the voltage is clamped at 5.6V. This caused the waveform displayed in the oscilloscope goes wrong. In this project, the optocoupler should be removed. For the next designer, a switch can be put between the pin and optocoupler. (Figure 24) For Pin 1/4/6/7/9a/10a/12a, they all connect with dataline or clockline. Therefore, the switch should also be connected in the circuit because when do the curvetrace, the voltage should be applied up to 15V, if there is no switch, the voltage will be clamped at 5.6V.

Problem has been solved:
Figure 25 is the schematic design for Pin3/9a which is NWAKE pin in TJA1020. When do the measurement for curve trace, the SMU voltage is supplied from -5V to 5V, when the voltage is negative, diode in the NMOS will conduct, which cause NWAKE pin to the ground. In order to forbid this situation, another diode is required to connect in the circuit. When the voltage is negative, NWAKE will not connect to the ground.

Regarding to TJA1024, Pin20 is LIN2. In the schematic, it uses Pin12 design circuit.
The LIN pin should connect with LIN termination. In the schematic design, there is also a function for this part.

Figure 26 is the schematic design for LIN termination. However, there is no connection to Pin12. Besides, PIN3 does not need LIN in fact. Therefore, the circuit was cut on Pin3 and connect Pin12 to LIN3 as the figure shows.

![Figure 26 LIN termination schematic](image)

Another problem is in TJA1024, Pin13 is VBAT1. However, in the schematic it is connected to the ground. Therefore Pin13 was connected to pin15 because pin15 is not connect in TJA1024 and it can supply the voltage power. (Figure 27)

![Figure 27 VBAT1 redesign TJA1024](image)

The whole automatic measurement can be tested automatically by clicking one button. The result will be saved in the Excel. This function has been finished. The program code can be simplified in the later design.
8 Bibliography


Appendices

A. The relation between the schematic design and the software

Figure 13 is for PINs Ha Lb Out. If 0 is given then it means Pina is used because PMOS conducted and NMOS not conducted, if 1 is set, then Pinb works. This design is used because the hardware test board should fit the whole family. Therefore, this schematic is also used for other pins, Pina is Pin3 which is used in TJA1020, while Pinb is for Pin9a, which is used in other type.

The powerIO11.output_on (CH3toTXD1); this is the code for close Channel3. Then LED works, the switch close, power received on Channel3, the waveform for TXD can be checked on oscilloscope.

For DAC and ADC, DAC is for input and ADC is for output, the difference between them is that for DAC, the value should be given, to let DAC high/low, which is used to make TXD high/low.

If DAC TXD.checked then

begin
  DAC1.sendvolt(pin4_pin10a_select,0);
  {powerIO11.output_off(Pin4_10a_SMU2on); } PowerIO11.output_on(Pin4_10a_DACon);
  DAC1.sendvolt(pin4_pin10a_DAC,5);
end
else
  Begin
  DAC1.sendvolt(pin4_pin10a_DAC,0);
  PowerIO11.output_off(Pin4_10a_DACon);
  end;

Once some switches has been put on, it should be put off when the function has finished and not used to make sure it will not cause any influence for other test.

For the PUVcc, it connect to a pull-up resistor, in TXD it is not used, but in RXD it is necessary. It is because RXD provides an open drain behavior in order to get an output level, which can be adapted to the microcontroller supply voltage. Thus 3.0V/3.3V microcontroller derivatives without 5V tolerant ports can be used. In case the
microcontroller port pin does not provide an integrated pull-up, an external pull-up resistor connected to the microcontroller supply voltage Vcc is required.

For SMU, it is almost the same. When put on SMU2, the device start work.

For the LIN Termination, the Schematic is a little bit different.

500/650/1K. there are three types of load can be chose to change the slope of LIN wave. On the right side, it is the output port for LIN, users can put on the one which is used.

Procedure
all_lin_termination_off;
Begin
With TJA1020 do
Begin
DAC1.SendVolt(n10_LIN,0);
DAC1.SendVolt(n6n8_LIN,0);
DAC1.SendVolt(n1_LIN,0);
powerIO11.output_off(Pin3_LINter);
powerIO11.output_off(Pin11_LINter);
powerIO11.output_off(Pin15_LINter);
powerIO11.output_off(Pin15a_LINter);
powerIO11.output_off(Pin9_LINter);
powerIO11.output_off(Pin14_LINter);
powerIO11.output_off(E500_10n_LIN);
powerIO11.output_off(E6506n8_LIN);
powerIO11.output_off(k11n_LIN);
End;
End;

This procedure is used to switch off all the LIN termination, the load is controlled by DAC value while the switch is controlled by Pout.
The function for NSLP and NWAKE is almost the same. Since NSLP and NWAKE is just used for wake the pin, the channel is not necessary to be displayed in the oscilloscope.

In INH pin, DAC is not required because INH pin is a battery related inhibit output for controlling an external voltage regulator. It’s active high after a wake-up event.
RXD is the output pin. Therefore, ADC is used. In RXD pin, a pull-up resistor is connected. It is because RXD provides an open drain behavior in order to get an output level, which can be adapted to the microcontroller supply voltage. Thus 3.0V/3.3V microcontroller derivatives without 5V tolerant ports can be used. In case the microcontroller port pin does not provide an integrated pull-up, an external pull-up resistor connected to the microcontroller supply voltage Vcc is required.

---

Figure 20 RXD pin flow chart

Figure 21 LIN pin flow chart
There are 2 SMU device can be used in the Lab, therefore, users can select the SMU he wants to use. The capacitor works as filter.
C. Functional design automatic measurement

- Pin TXD

TXD measurement is the same for both TJA1020 and 1021

High-level input voltage

This characteristic is not easy to measure because it happens in dominant. It should be checked on RXD pin, however, with a pull-up resistor on RXD pin, the voltage change quickly for high to low then stay high again. The SMU cannot have that value. Therefore, other method should be used. In this measurement, the battery current can be used in this measurement. When the voltage is swept from 0.8V – 2V on TXD, at first, the battery current increase, however, when it reaches a value where RXD flash, the current will increase over 2uA and goes down. Therefore, two values are set here, the \( I_{\text{BAT}}' \) is the measured value and \( I_{\text{BAT}} \) is the value measured last time, if the old one is over 2 uA higher, then it can be used as the high – level input voltage.

Low-level input voltage

This function is same as HIGH – level input voltage. The difference is the change value is 1 uA.

TXD hysteresis voltage

Hysteresis voltage equals to the high-level minus low-level.

TXD pull-down resistor (\( V_{\text{TXD}} = 5V \))

To measure the resistor value. SMU2 is set for TXD 5V which is mentioned in the datasheet, then measure the current. Ohm law is used to calculate the resistance.
Low level output current (local wake-up request) (Standby mode; $V_{\text{WAKE}}=0V; V_{\text{LIN}}=V_{\text{BAT}}; V_{\text{TXD}}=0.4V)$

Low level input current ($V_{\text{TXD}}=0V$)

For these two functions, based on the orders on the datasheet, set the voltage for TXD and battery pin. It is convenient to measure the current on TXD.
Pin NSLP

NSLP measurement is the same for both TJA1020 and 1021

High-level input voltage

This flowchart uses the same function as TXD. In order to check when NSLP is high or low, different mode should be changed during the test.

When do High-level measurement, the chip should be made to go to sleep mode to make NSLP low, than the voltage will be increased on NSLP from 0.8V to 2V according to the datasheet, the current on battery pin can be measured because in sleep mode, the battery current is only 3/7 uA, but when NSLP is high which means the chip go to normal mode, the current becomes over 100uA.

This function can be also measured by checking INH because in sleep mode, INH is floating while in normal mode, INH is high. However, only two SMU is available to set the battery and NSLP, there is no extra SMU to set and measure INH.

Low-level input voltage

The same measurement method is applied here as high-level, when the current measured on Battery pin is lower than 8uA, it means the chip goes to sleep mode.
NSLP hysteresis voltage

\[ V_{\text{HYS}} = V_{\text{IH}} - V_{\text{IL}} \]

NSLP pull down resistor \((V_{\text{NSLP}}=5V)\)

Set NSLP voltage 5V

Measure NSLP Current

\[ R_{\text{NSLP}} = \frac{5V}{I_{\text{NSLP}}} \]

For these three measurement, they are the same as the measurement in TXD. Hysteresis voltages equals to high-level minus low-level. Using Ohm law to measure the resistance. Obey the orders in datasheet, set voltage and measure the current.

- Pin RXD

RXD measurement is the same for both TJA1020 and 1021

Low level output current (Normal slope mode; \(V_{\text{LIN}}=0V; V_{\text{RXD}}=0.4V\))

High level leakage current (Normal slope mode; \(V_{\text{LIN}}=V_{\text{BAT}}; V_{\text{RXD}}=5V\))

These two measurement is also done based on the requirements in datasheet. Since SMU1 used for battery pin, SMU2 used for RXD. DAC is used to set LIN 0V in low level.

For high level, when chips go to normal mode, LIN is already high, so it is not necessary to set it. DAC also can’t be used while voltage is over 5V.
Pin NWAKE

NWAKE measurement is the same for both TJA1020 and 1021

High level input voltage

There is no available phenomenon when NWAKE is high. To measure this data, a very low current is given on NWAKE pin and measure the voltage on NWAKE. It is because in the block diagram, NWAKE pin is connected to Battery pin through a current source.

Low level input voltage

This measurement can be done by checking the battery current. In sleep mode, when NWAKE pin go from high to low, the chip will go from sleep mode to the standby mode. In sleep mode, the battery current is very low 3/7 uA while in standby mode, the current will rises to over 100uA.

NWAKE pull-up current ($V_{NWAKE} = 0V$)

For these two measurement, based on the requirements in the data, it is convenient to measure the NWAKE current.
• **Pin INH**

INH measurement is the same for both TJA1020 and 1021

Switch-on resistance between pins BAT and INH (standby; low slope or normal slope mode; $I_{INH} = -15\text{mA}$; $V_{BAT} = 12\text{V}$)

- Set battery voltage 12V
- Set INH current -15mA
- Go to normal slope mode
- Measure INH voltage

$$R_{SW(INH)} = \frac{(V_{INH} - V_{BAT})}{-0.015}$$

High leakage current (sleep mode; $V_{INH} = 27\text{V}$; $V_{BAT} = 27\text{V}$)

- Set battery voltage 12V
- Go to sleep mode
- Set INH voltage 27V
- Measure INH current

For these two measurement, they are tested based on the requirements in the datasheet. Using ohm law to measure the resistance between Pin BAT and INH.

• **Pin LIN**

LIN measurement is the different in TJA1020 and 1021 because they measured different characteristics.

LIN recessive output voltage ($V_{TXD} = 5\text{V}; I_{LIM} = 0\text{mA}$)

- Set battery voltage 12V
- Use DAC to set TXD 5V
- Use SMU set LIN current 0 mA
- Measure LIN Voltage

High level leakage current ($V_{IN} = V_{SAT}$)

- Set battery voltage 12V
- Set LIN voltage $V_{SAT} = 12\text{V}$
- Measure LIN current

For these two characteristics which should be measured in TJA1020, according to the requirements in the datasheet. Setting the required voltage and current to do the measurement.
Receiver threshold voltage (recessive) \( (V_{\text{BAT}} = 7.3 \text{ – } 27\text{V}) \)

This measurement is same in TJA1020 and TJA1021.

The chip will be set to normal mode at first, then the LIN voltage will be set to 0V, the RXD will become low. Then increase the voltage on LIN, to a certain value, the RXD will become high. SMU1 used to set battery voltage and SMU2 used to set LIN voltage. ADC can be used to measure the voltage on RXD.

Receiver threshold voltage (dominant)

This measurement is the same in TJA1020 and TJA1021.

The chip is set to go to normal mode, then LIN voltage is set to the battery voltage, the RXD is high at this time. Then decrease the voltage on LIN, to a certain value, the RXD will become low. SMU1 used to set battery voltage and SMU2 used to set LIN voltage. ADC can be used to measure the voltage on RXD.

Receiver center voltage \( (V_{\text{BAT}} = 7.3 \text{ – } 27\text{V}) \)

\[
V_{\text{CTR}(\text{rx})} = \frac{V_{\text{th}(\text{rx})(\text{recessive})} + V_{\text{th}(\text{rx})(\text{dominant})}}{2}
\]

Receiver threshold hysteresis voltage \( (V_{\text{BAT}} = 7.3 \text{ – } 27\text{V}) \)

\[
V_{\text{th}(\text{hys})(\text{rx})} = V_{\text{th}(\text{rx})(\text{recessive})} - V_{\text{th}(\text{rx})(\text{dominant})}
\]

The center voltage equals to \( (V \text{ recessive plus } V \text{ dominant})/2 \).

The hysteresis voltage equals to \( V \text{ recessive minus } V \text{ dominant} \).
D. Physical design manual measurement

TJA1021

TJA1022

TJA1024

TJA1027

TJA1028
E. Program code

TJA1020

- TXD

procedure TTJA1020.DACTXDClick(Sender: TObject);
{TXD DAC}
begin
If DACTXD.checked then
Power1011.output_on(Pin4_10a_DACon)
else
Power1011.output_off(Pin4_10a_DACon)
end;

procedure TTJA1020.HLTXDClick(Sender: TObject);
{TXD DAC High/Low}
begin
If HLTXD.checked then
DAC1.sendvolt(pin4_pin10a_DAC,5)
{TXD DAC High}
else
DAC1.sendvolt(pin4_pin10a_DAC,0)
{TXD DAC Low}
end;

procedure TTJA1020.SMU2TXDClick(Sender: TObject);
{TXD SMU}
begin
If SMU2TXD.checked then
begin
powerIO11.output_on(Pin4_10a_SMU2on);
SMU2RXD.enabled:=false;
SMU2LIN.enabled:=false;
SMU2NSLP.enabled:=false;
SMU2NWAKE.enabled:=false;
SMU2INH.Enabled:=false;
end
else
Begin
powerIO11.output_off(Pin4_10a_SMU2on);
SMU2RXD.enabled:=true;
SMU2LIN.enabled:=true;
SMU2NSLP.enabled:=true;
SMU2NWAKE.enabled:=true;
SMU2INH.Enabled:=true;
end;
end;

- RXD

procedure TTJA1020.CH1RXDClick(Sender: TObject);
{RXD Channel}
begin
If CH1RXD.checked then
begin
powerIO11.output_off(CH1_toINH);
CH1INH.enabled:=false;
powerIO11.output_on(Pin1_7_CH1on);
end
else
begin
CH1INH.enabled:=true;
powerIO11.output_off(Pin1_7_CH1on);
end;

procedure TTJA1020.ADCRXDClick(Sender: TObject);
{RXD Value}
var
value:real;
n:real; // define two real number
end;

procedure TTJA1020.PURXDClick(Sender: TObject);
{pull up resistor}
begin
If PURXD.checked then
PowerIO11.output_on(Pin1_7_PUVccon)
else
PowerIO11.output_off(Pin1_7_PUVccon)
end;
procedure TTJA1020.CH2LINClick(Sender: TObject);
// LIN Channel
begin
If CH2LIN.checked then begin
powerIO11.output_off(CH2toNSLP);
CH2NSLP.enabled:=false;
powerIO11.output_on(Pin14_CH2on);
end
else begin
CH2NSLP.enabled:=true;
powerIO11.output_off(Pin14_CH2on);
end;
end;

procedure TTJA1020.LINoffClick(Sender: TObject);
// LIN termination Off
begin
If LINOFF.checked then begin
all_lin_termination_off;
// Call function
end;
end;

Procedure all_lin_termination_off;
// Define a function all_lin_termination_off
Begin
With TJA1020 do
Begin
DAC1.SendVolt(n10_LIN,0);
DAC1.SendVolt(n6n8_LIN,0);
DAC1.SendVolt(n1_LIN,0);
powerIO11.output_off(Pin3_LINter);
powerIO11.output_off(Pin11_LINter);
powerIO11.output_off(Pin15_LINter);
powerIO11.output_off(Pin15a_LINter);
powerIO11.output_off(Pin9_LINter);
powerIO11.output_off(Pin14_LINter);
powerIO11.output_off(E500_10n_LIN);
powerIO11.output_off(E6506n8_LIN);
powerIO11.output_off(k11n_LIN);
End;
End;

procedure TTJA1020.LIN500Click(Sender: TObject);
// LIN termination 500Ω
begin
If LIN500.Checked then begin
all_lin_termination_off;
DAC1.SendVolt(n10_LIN,5);
PowerIO11.output_on(E500_10n_LIN);
powerIO11.output_on(Pin14_LINter);
end;
end;

procedure TTJA1020.LIN650Click(Sender: TObject);
// LIN termination 650Ω
begin
if LIN650.Checked then begin
all_lin_termination_off;
DAC1.SendVolt(n6n8_LIN,5);
PowerIO11.output_On(E6506n8_LIN);
powerIO11.output_on(Pin14_LINter);
end;
end;

procedure TTJA1020.LIN1KClick(Sender: TObject);
// LIN termination 1KΩ
begin
if LIN1k.Checked then begin
all_lin_termination_off;
DAC1.SendVolt(n1_LIN,5);
PowerIO11.output_On(k11n_LIN);
powerIO11.output_on(Pin14_LINter);
end;
end;
o NSLP

procedure TTJA1020.CH2NSLPClick(Sender: TObject);
// NSLP Channel
begin
If CH2NSLP.checked then
begin
CH2LIN.Enabled:=false;
powerIO11.output_on(CH2toNSLP);
end
else
Begin
CH2LIN.Enabled:=true;
powerIO11.output_off(CH2toNSLP);
end;
end;

procedure TTJA1020.DACNSLPClick(Sender: TObject);
// NSLP DAC
begin
If DACNSLP.checked then
PowerIO11.output_on(Pin2_8_DACon)
else
PowerIO11.output_off(Pin2_8_DACon)
end;

procedure TTJA1020.HLSLPClick(Sender: TObject);
// NSLP DAC H/L
begin
If HLSLP.checked then
DAC1.sendvolt(pin2_pin8_DAC,5) // NSLP DAC High
else
DAC1.sendvolt(pin2_pin8_DAC,0) // NSLP DAC Low
end;

o NWAKE

procedure TTJA1020.CH3NWAKEClick(Sender: TObject);
// NWAKE Channel
begin
If CH3NWAKE.checked then
begin
CH3toTXD.enabled:=false;
powerIO11.output_on(CH3toNWAKE);
end
else
Begin
CH3toTXD.enabled:=true;
powerIO11.output_off(CH3toNWAKE);
end;
end;

procedure TTJA1020.HLDACClick(Sender: TObject);
begin
if HLDAC.checked then begin
DAC1.SendVolt(pin3_pin9a_DAC,0);
//Set Nwake low to make TJ1020 to
standby mode
powerIO11.output_on(Pin3_9a_DACon);
end
else
powerIO11.output_off(Pin3_9a_DACon)
end;

o INH

procedure TTJA1020.CH1INHClick(Sender: TObject);
// INH Channel
begin
If CH1INH.checked then begin
CH1RXD.Enabled:=false;
powerIO11.output_on(CH1toINH)
end
else
Begin
CH1RXD.enabled:=true;
powerIO11.output_off(CH1toINH)
end;
end;

o BAT

procedure TTJA1020.CH4BATClick(Sender: TObject);
// BAT Channel
begin
If CH4BAT.checked then begin
powerIO11.output_on(Pin15_9a_CH4on)
end
else
powerIO11.output_off(Pin15_9a_CH4on)
end;
procedure TTJA1020.C1Click(Sender: TObject);
  // BAT capacitor 10uf
begin
  If C1.checked then
  DAC1.sendvolt(pin15_9a_10ufOn,5)
  else
  DAC1.sendvolt(pin15_9a_10ufOn,0)
end;

procedure TTJA1020.C2Click(Sender: TObject);
  // BAT capacitor 100nf
begin
  If C2.checked then
  DAC1.sendvolt(pin15_9a_100nfOn,5)
  else
  DAC1.sendvolt(pin15_9a_100nfOn,0)
end;

o  Sleep Mode

procedure TTJA1020.SLPMClick(Sender: TObject);
begin
  if SLPM.checked then begin
  STBM.enabled:=false;
  NSM.enabled:=false;
  LSM.enabled:=false; // Disable other three mode
  SMU1LIN.checked:=true;
  PURXD.checked:=true; // Pull up resistor on RXD
  HLTXD.checked:=true; // Set TXD high
  DACNSLP.checked:=true;
  HLSLP.checked:=true; // Set NSLP high
  sleep(500);
  HLSLP.checked:=false; // Go to Sleep Mode
  HLDAC.checked:=true; // Set NWAKE low
  end
  else begin
  SMU1LIN.checked:=false;
  PURXD.checked:=false;
  DACTXD.checked:=false;
  HLTXD.checked:=false;
  DACNSLP.checked:=false;
  HLSLP.checked:=false;
  HLDAC.checked:=false;
  SMU1LIN.checked:=true;
  NSM.enabled:=true;
  LSM.enabled:=true;
  end
  end;

o  Standby Mode

procedure TTJA1020.STBMClick(Sender: TObject);
begin
  if STBM.checked then begin
  SLPM.enabled:=false;
  NSM.enabled:=false;
  LSM.enabled:=false;
  SMU1LIN.checked:=true;
  PURXD.checked:=true;
  DACTXD.checked:=true;
  HLTXD.checked:=true;
  DACNSLP.checked:=true;
  HLSLP.checked:=true;
  sleep(500);
  HLSLP.checked:=false; // Go to Sleep Mode
  HLDAC.checked:=true; // Set NWAKE low
  end
  else begin
  SMU1LIN.checked:=false;
  PURXD.checked:=false;
  DACTXD.checked:=false;
  HLTXD.checked:=false;
  DACNSLP.checked:=false;
  HLSLP.checked:=false;
  HLDAC.checked:=false;
  SMU1LIN.checked:=true;
  NSM.enabled:=true;
  LSM.enabled:=true;
  end
  end;
procedure TTJ1A020.LSMClick(Sender: TObject);
begin
if LSM.checked then begin
STBM.enabled:=false;
SLPM.enabled:=false;
NSM.enabled:=false;
SMU1LIN.checked:=true;
PURXD.checked:=true;
DAC1.sendvolt(pin4_pin10a_DAC,0);
// Set TXD low
DACTXD.checked:=true;
DACNSLP.checked:=true;
HLNSLP.checked:=true;
// Set NSLP high
end
else begin
STBM.enabled:=true;
SLPM.enabled:=true;
NSM.enabled:=true;
SMU1LIN.checked:=false;
PURXD.checked:=false;
DACTXD.checked:=false;
DACNSLP.checked:=false;
HLNSLP.checked:=false;
end;
end;

procedure TTJ1A020.NSMClick(Sender: TObject);
begin
if NSM.checked then begin
STBM.enabled:=false;
SLPM.enabled:=false;
NSM.enabled:=false;
SMU1LIN.checked:=true;
PURXD.checked:=true;
DACTXD.checked:=true;
HLTXD.checked:=true; // Set TXD high
DACNSLP.checked:=true;
HLNSLP.checked:=true;
end
else begin
STBM.enabled:=true;
SLPM.enabled:=true;
NSM.enabled:=true;
SMU1LIN.checked:=false;
PURXD.checked:=false;
DACTXD.checked:=false;
DACNSLP.checked:=false;
HLNSLP.checked:=false;
end;
end;
F. Physical design automatic measurement

TJA1021

TJA1022

TJA1024
G. Program Code

1. Pin Battery

1.1 Sleep mode

function supply_current_on_batterySRmode
(VBattery:real):real;
var
value:real;
Begin
With TJA1020meas do
Begin
setSMUvoltage(1,GPIBAdres_SMU1,VBattery,10e-3);
// Set SMU1 voltage 12V;
DAC1.sendvolt(pin4_pin10a_DAC,0);
// TXD low;
DAC1.sendvolt(pin2_pin8_DAC,5);
// NSLP high;
PowerIO11.output_on(Pin4_10a_DACon);
PowerIO11.output_on(Pin1_7_PUVccon);
PowerIO11.output_on(Pin4_10a_DACon);
DAC1.sendvolt(pin2_pin8_DAC,5);
// Low slope mode;
PowerIO11.output_on(Pin2_8_DACon);
// NSLP low-> sleep mode;
sleep(100);
value:=measureSMUcurrent(1,GPIBAdres_SMU1)*1e6;
ISR.caption:=floattostr(value)+' uA';
setSMUvoltage(1,GPIBAdres_SMU1,0,10e-3);
powerIO11.output_off(Pin15_9a_SMU1on);
powerIO11.output_off(Pin4_10a_DACon);
powerIO11.output_off(Pin2_8_DACon);
// Close all device;
end;
end;

1.2 Standby mode all recessive

function supply_current_on_batterySRmode
(VBattery:real):real;
var
value:real;
Begin
With TJA1020meas do
Begin
setSMUvoltage(1,GPIBAdres_SMU1,VBattery,10e-3);
powerIO11.output_on(Pin15_9a_SMU1on);
DAC1.sendvolt(pin4_pin10a_DAC,0);
PowerIO11.output_on(Pin4_10a_DACon);
PowerIO11.output_on(Pin2_8_DACon);
// Sleep mode;
sleep(100);
value:=measureSMUcurrent(1,GPIBAdres_SMU1)*1e6;
ISR.caption:=floattostr(value)+' uA';
setSMUvoltage(1,GPIBAdres_SMU1,0,10e-3);
powerIO11.output_off(Pin15_9a_SMU1on);
powerIO11.output_off(Pin4_10a_DACon);
powerIO11.output_off(Pin2_8_DACon);
end;
end;

1.3 Normal slope mode all recessive

function supply_current_on_batteryNRmode
(VBattery:real):real;
var
value:real;
Begin
With TJA1020meas do
Begin
setSMUvoltage(1,GPIBAdres_SMU1,VBattery,10e-3);
powerIO11.output_on(Pin15_9a_SMU1on);
DAC1.sendvolt(pin4_pin10a_DAC,5);
PowerIO11.output_on(Pin1_7_PUVccon);
PowerIO11.output_on(Pin4_10a_DACon);
DAC1.sendvolt(pin2_pin8_DAC,5);
// Normal slope mode;
sleep(100);
value:=measureSMUcurrent(1,GPIBAdres_SMU1)*1e6;
ISR.caption:=floattostr(value)+' uA';
setSMUvoltage(1,GPIBAdres_SMU1,0,10e-3);
powerIO11.output_off(Pin15_9a_SMU1on);
powerIO11.output_off(Pin4_10a_DACon);
powerIO11.output_off(Pin2_8_DACon);
end;
end;
1.4 Low slope mode all recessive
function supply_current_on_batteryLRmode(VBattery:real):real;
var
value:real;
Begin
With TJA1020meas do
Begin
setSMUvoltage(1,GPIBadres_SMU1,VBattery,10e-3);
powerIO11.output_on(Pin15_9a_SMU1on);
DAC1.sendvolt(pin4_pin10a_DAC,5);
PowerIO11.output_on(Pin4_10a_DACon);
PowerIO11.output_on(Pin2_8_DACon);
DAC1.sendvolt(pin4_pin10a_DAC,0);
// Low slope mode;
sleep(100);
value:=measureSMUcurrent(1,GPIBadres_SMU1)*1e6;
ILR.caption:=floattostr(value)+' uA';
setSMUvoltage(1,GPIBadres_SMU1,0,10e-3);
powerIO11.output_off(Pin15_9a_SMU1on);
PowerIO11.output_off(Pin4_10a_DACon);
PowerIO11.output_off(Pin2_8_DACon);
DAC1.sendvolt(pin4_pin10a_DAC,0);
end;
end;

2.2 Low-level input current
function supply_lcurrent_on_TXD(VTXD:real):real;
var
value:real;
Begin
With TJA1020meas do
Begin
setSMUvoltage(1,GPIBadres_SMU1,12,10e-3);
// Set battery voltage 12V;
powerIO11.output_on(Pin15_9a_SMU1on);
setSMUvoltage(2,GPIBadres_SMU2,VTXD,10e-3);
// Set TXD voltage 0 V;
powerIO11.output_on(Pin4_10a_SMU2on);
sleep(100);
value:=measureSMUcurrent(2,GPIBadres_SMU2)*1e6;
// Measure TXD current;
LCT.caption:=floattostr(value)+' uA';
setSMUvoltage(2,GPIBadres_SMU2,0,10e-3);
powerIO11.output_off(Pin4_10a_SMU2on);
end;
end;

2.3 Low-level output current
function supply_lcurrentout_on_TXD(VBattery:real):real;
var
value:real;
Begin
With TJA1020meas do
Begin
setSMUvoltage(2,GPIBadres_SMU2,VTXD,10e-3);
// Set SMU2 voltage for TXD 5V;
powerIO11.output_on(Pin4_10a_SMU2on);
setSMUvoltage(1,GPIBadres_SMU1,0,10e-3);
powerIO11.output_on(Pin15_9a_SMU1on);
sleep(100);
value:=measureSMUcurrent(2,GPIBadres_SMU2)*1e6;
// Measure current on TXD;
LCT.caption:=floattostr(value)+' uA';
setSMUvoltage(1,GPIBadres_SMU1,0,10e-3);
powerIO11.output_off(Pin15_9a_SMU1on);
end;
end;

2 Pin TXD

2.1 TXD pull-down resistor
function pulldown_resistor_on_TXD(VTXD:real):real;
var
value:real;
Re:real;
Begin
With TJA1020meas do Begin
setSMUvoltage(2,GPIBadres_SMU2,VTXD,10e-3);
// Set SMU2 voltage for TXD 5V;
powerIO11.output_on(Pin4_10a_SMU2on);sleep(100);
value:=measureSMUcurrent(2,GPIBadres_SMU2)*1e3;
// Measure current on TXD;
re:=Trunc(5/value*1000)/1000;
// R=U/I;
PUR.caption:=floattostr(re)+' k';
setSMUvoltage(2,GPIBadres_SMU2,0,10e-3);
powerIO11.output_off(Pin4_10a_SMU2on);
end;
end;
DAC1.sendvolt(pin2_pin8_DAC,0);  
DAC1.SendVolt(pin3_pin9a_DAC,0);  
// Set VNWK 0V; 
powerIO11.output_on(Pin3_9a_DACon); 
PowerIO11.output_off(Pin4_10a_DACon);  
setSMUvoltage(2,GPIBadres_SMU2,0.4,10e-3);  
// Set TXD voltage 0.4V;  
powerIO11.output_on(Pin4_10a_SMU2on); 
sleep(100);  
value:=measureSMUcurrent(2,GPIBadres_SMU2)*1e3;  
// Measure TXD current; 
LOCT.caption:=floattostr(value)+' mA';  
setSMUvoltage(1,GPIBadres_SMU1,0,10e-3);  
setSMUvoltage(1,GPIBadres_SMU1,0,10e-3);  
powerIO11.output_off(Pin15_9a_SMU1on); 
PowerIO11.output_off(Pin1_7_PUVcon);  
PowerIO11.output_off(Pin2_8_SMU2on);  
powerIO11.output_off(Pin4_10a_SMU2on);  
end;  
end;  
3
Pin NSLP  
3.1 High-level input voltage  
function Highinput_voltage_NSLP(VBattery:real):real;  
var  
value:real;  
counter:integer;  
Ustep:real;  
ct:integer;  
Begin  
With TJA1020meas do  
Begin  
setSMUvoltage(1,GPIBadres_SMU1,VBattery,10e-3);  
setSMUvoltage(1,GPIBadres_SMU1,VBattery,10e-3);  
powerIO11.output_on(Pin15_9a_SMU1on);  
DAC1.sendvolt(pin4_pin10a_DAC,5);  
PowerIO11.output_on(Pin1_7_PUVcon);  
PowerIO11.output_on(Pin4_10a_DACon);  
PowerIO11.output_on(Pin2_8_SMU2on);  
Counter:=Round((abs(2-0.8)/0.025)+0.025);  
// Set NSLP Voltage from 0.8V-2V;  
Ustep:=0.025;  

if value >=470 then break;  
end;  
NSLPIL.Caption:=floattostr(Ihigh)+' V';  
setSMUvoltage(2,GPIBadres_SMU2,0,10e-3);  
setSMUvoltage(1,GPIBadres_SMU1,0,10e-3);  
powerIO11.output_off(Pin15_9a_SMU1on); 
DAC1.sendvolt(pin4_pin10a_DAC,0);  
PowerIO11.output_off(Pin1_7_PUVcon);  
PowerIO11.output_off(Pin4_10a_DACon);  
PowerIO11.output_off(Pin2_8_SMU2on);  
end;  
end;  
3.2 Low-level input voltage  
function Lowinput_voltage_NSLP(VBattery:real):real;  
var  
value:real;  
counter:integer;  
Ustep:real;  
ct:integer;  
Begin  
With TJA1020meas do  
Begin  
setSMUvoltage(1,GPIBadres_SMU1,VBattery,10e-3);  
setSMUvoltage(1,GPIBadres_SMU1,VBattery,10e-3);  
powerIO11.output_on(Pin15_9a_SMU1on);  
DAC1.sendvolt(pin4_pin10a_DAC,5);  
PowerIO11.output_on(Pin1_7_PUVcon);  
PowerIO11.output_on(Pin4_10a_DACon);  
PowerIO11.output_on(Pin2_8_SMU2on);  
Counter:=Round((abs(2-0.8)/0.025)+0.025);  
// Set NSLP Voltage from 2V-0.8V;  
Ustep:=0.025;  
ilow:=2;  
for ct:=0 to counter do begin  
setSMUvoltage(2,GPIBadres_SMU2,ilow,10e-3);  
value:=measureSMUcurrent(1,GPIBadres_SMU1)*1e6;  
// Measure current on Battery, if it's  
if value >470 then ilow:=ilow-Ustep;  
lower then 470uA it go from normal mode to sleep mode;  
if value <=470 then break;  
end;  
ilow:=ilow+Ustep;  
NSLPIL.Caption:=floattostr(ilow)+' V';  
setSMUvoltage(2,GPIBadres_SMU2,0,10e-3);  
setSMUvoltage(1,GPIBadres_SMU1,0,10e-3);  
powerIO11.output_off(Pin15_9a_SMU1on); 
DAC1.sendvolt(pin4_pin10a_DAC,0);  
PowerIO11.output_off(Pin1_7_PUVcon);  
PowerIO11.output_off(Pin4_10a_DACon);  
PowerIO11.output_off(Pin2_8_SMU2on);  
end;  
end;
3.3 NSLPhysteresis voltage
procedure TTJA1020meas.HVNSLPClick(Sender: TObject);
var
i:real;
begin
if HVNSLP.checked then begin
  i:=ihigh-ilow;
  // Hysteresis voltage = Lehigh-Lelow;
  VHNSLP.Caption:=floattostr(i)+' V';
end
else
  VHNSLP.Caption:='0 V';
end;

3.4 NSLP pull-down resistor
function pulldown_resistor_on_NSLP(Vnslp:real):real;
var
value:real;
Re:real;
Begin
With TJA1020meas do
Begin
  setSMUvoltage(2,GPIBadres_SMU2,Vnslp,10e-3);
  // Set NSLP voltage 5V;
  powerIO11.output_on(Pin2_8_SMU2on);
sleep(100);
  value:=measureSMUcurrent(2,GPIBadres_SMU2)*1e3;
  re:=trunc(5/value*1000)/1000;
  RNSLP.caption:=floattostr(re)+' k';
  setSMUvoltage(2,GPIBadres_SMU2,0,10e-3);
  // Set NSLP voltage 0V;
  powerIO11.output_off(Pin2_8_SMU2on);
end;
end;

3.5 Low-level input current
function supply_lcurrent_on_NSLP(VBattery:real):real;
var
value:real;
Begin
With TJA1020meas do
Begin
  setSMUvoltage(1,GPIBadres_SMU1,VBattery,10e-3);
  // Set Battery voltage 12V;
  powerIO11.output_on(Pin15_9a_SMU1on);
  setSMUvoltage(2,GPIBadres_SMU2,0,10e-3);
  // Set NSLP voltage 0V;
  powerIO11.output_on(Pin2_8_SMU2on);
sleep(100);
  value:=measureSMUcurrent(2,GPIBadres_SMU2)*1e6;
  // Measure RXD current;
  ILHRXD.caption:=floattostr(value)+' uA';
  setSMUvoltage(1,GPIBadres_SMU1,0,10e-3);
  powerIO11.output_off(Pin15_9a_SMU1on);
  powerIO11.output_off(Pin2_8_SMU2on);
end;
end;

4. Pin RXD
4.1 High-level leakage current
function supply_Lecurrent_on_batteryRXD(VBattery:real):real
var
value:real;
Begin
With TJA1020meas do
Begin
  setSMUvoltage(1,GPIBadres_SMU1,VBattery,10e-3);
  powerIO11.output_on(Pin15_9a_SMU1on);
  PowerIO11.output_on(Pin4_10a_DACOn);
  DAC1.sendvolt(pin4_pin10a_DAC,5);
  PowerIO11.output_on(Pin2_8_DACOn);
  DAC1.sendvolt(pin2_pin8_DAC,5);
  powerIO11.output_on(Pin1_7_SMU2on);
  // Set RXD voltage 5V to go to normal slope mode;
  setSMUvoltage(2,GPIBadres_SMU2,5,10e-3);
  sleep(100);
  value:=measureSMUcurrent(2,GPIBadres_SMU2)*1e6;
  // Measure RXD current;
  ILHRXD.caption:=floattostr(value)+' uA';
  setSMUvoltage(1,GPIBadres_SMU1,0,10e-3);
  setSMUvoltage(2,GPIBadres_SMU2,0,10e-3);
  powerIO11.output_off(Pin15_9a_SMU2on);
  PowerIO11.output_off(Pin4_10a_DACOn);
  PowerIO11.output_off(Pin2_8_DACOn);
  powerIO11.output_off(pin1_7_SMU2on);
end;
end;
5 Pin NWAKE

5.1 Low-level input voltage

function Lowinput_voltage_NWAKE(VBat:real):real;
var
value:real;
counter:integer;
Ustep:real;
ct:integer;
i:real;
Begin
With TJA1020meas do
Begin
setSMUvoltage(1,GPIBadres_SMU1,VBat,10e-3);
powerIO11.output_on(Pin15_9a_SMU1on);
DAC1.sendvolt(pin4_pin10a_DAC,5);
PowerIO11.output_on(Pin1_7_PUVcon);
PowerIO11.output_on(Pin4_10a_DACon);
DAC1.sendvolt(pin2_pin8_DAC,5);
powerIO11.output_on(Pin4_10a_DACon);
DAC1.sendvolt(pin2_pin8_DAC,0);
// Go to sleep mode;
setSMUvoltage(2,GPIBadres_SMU2,VBat-1,10e-3);
// Set NWAKE voltage from VBat-3.3 to VBat-1;
Counter:=Round((abs(VBat-1-VPat+3.3)/0.1)+0.1);
Ustep:=0.1;
i:=VPat-1;
for ct:=0 to counter do begin
setSMUvoltage(2,GPIBadres_SMU2,i,10e-3);
value:=measureSMUcurrent(1,GPIBadres_SMU1)*1e6;
// Measure current on Battery pin, if it's higher than 350uA, then it goes to standby mode;
if value <350 then i:=i-Ustep;
if value >=350 then break;
end;
i:=VPat-i;
VILNWAKE.Caption:=''VPat-''+floattostr(i);
setSMUvoltage(2,GPIBadres_SMU2,0,10e-3);
setSMUvoltage(1,GPIBadres_SMU1,0,10e-3);
powerIO11.output_off(Pin15_9a_SMU1on);
DAC1.sendvolt(pin4_pin10a_DAC,0);
PowerIO11.output_off(Pin1_7_PUVcon);
PowerIO11.output_off(Pin4_10a_DACon);
PowerIO11.output_off(Pin3_9a_SMU2on);
end;
end;

5.1 NWAKE pull-up current

function pull_up_on_NWAKE(VNWAKE:real):real;
var
value:real;
Begin
With TJA1020meas do
Begin
setSMUvoltage(1,GPIBadres_SMU1,12,10e-3);
// Set battery voltage 12V;
powerIO11.output_on(Pin15_9a_SMU1on);
setSMUvoltage(2,GPIBadres_SMU2,VNWAKE,10e-3);
// Set NWAKE voltage 0V;
powerIO11.output_on(Pin3_9a_SMU2on);
sleep(100);
value:=measureSMUcurrent(2,GPIBadres_SMU2)*1e6;
// Measure NWAKE current;
IILNWAKE.caption:=floattostr(value)+' uA';
setSMUvoltage(2,GPIBadres_SMU2,0,10e-3);
powerIO11.output_off(Pin3_9a_SMU2on);
end;
end;

5.2 High-level leakage current

function high_leakage_on_NWAKE(VBattery:real):real;
var
value:real;
Begin
With TJA1020meas do
Begin
setSMUvoltage(1,GPIBadres_SMU1,VBattery,10e-3);
// Set battery &NWAKE voltage 27V;
setSMUvoltage(2,GPIBadres_SMU2,VBattery,10e-3);
powerIO11.output_on(Pin15_9a_SMU1on);
powerIO11.output_on(Pin3_9a_SMU2on);
sleep(100);
value:=measureSMUcurrent(2,GPIBadres_SMU2)*1e6;
// Measure NWAKE current;
ILHNWAKE.caption:=floattostr(value)+' uA';
setSMUvoltage(2,GPIBadres_SMU2,0,10e-3);
powerIO11.output_off(Pin3_9a_SMU2on);
end;
end;
6. Pin INH

6.1 Switch-on resistance between pins BAT and INH

function Rbetween_pinBAT_pinINH(inh:real):real;
var
value:real;
R:real;
Begin
With TJA1020meas do
Begin
setSMUvoltage(1,GPIBadres_SMU1,12,100e-3);
powerIO11.output_on(pin15_9a_SMU1on);
setSMUcurrent(2,GPIBadres_SMU2,inh,12);
powerIO11.output_on(pin16_SMUon);
// Normal slope mode, Set battery voltage 12 V, set INH current -15mA;
PowerIO11.output_on(Pin1_7_PUVccon);
DAC1.sendvolt(pin2_pin8_DAC,5);
PowerIO11.output_on(Pin2_8_DACon);
sleep(100);
value:=measureSMUvoltage(2,GPIBadres_SMU2);
R:=(value-12)/-0.015;
// Resistance between Pin BAT and INH equals to (Vbat-VINH)/I;
RSWINH.Caption:=floattostr(R);
setSMUvoltage(1,GPIBadres_SMU1,0,10e-3);
powerIO11.output_off(pin15_9a_SMU1on);
setSMUvoltage(2,GPIBadres_SMU2,0,10e-3);
powerIO11.output_off(pin16_SMUon);
end;
end;

6.2 High-level leakage current

function supply_current_on_INH(vbattery:real):real;
var
value:real;
Begin
With TJA1020meas do
Begin
setSMUvoltage(1,GPIBadres_SMU1,vbattery,10e-3);
powerIO11.output_on(Pin2_8_DACon);
DAC1.sendvolt(pin2_pin8_DAC,5);
powerIO11.output_on(Pin15_9a_SMU1on);
// Sleep mode, Set VBAT=VINH=27V;
DAC1.sendvolt(pin2_pin8_DAC,0);
setSMUvoltage(2,GPIBadres_SMU2,vbattery,10e-3);
powerIO11.output_on(Pin16_SMUon);
sleep(100);
value:=measureSMUcurrent(2,GPIBadres_SMU2)*1e-6;
// Measure current on INH
ILIHINH.caption:=floattostr(value)+' uA';
setSMUvoltage(1,GPIBadres_SMU1,0,10e-3);
setSMUvoltage(2,GPIBadres_SMU2,0,10e-3);
powerIO11.output_off(Pin15_9a_SMU1on);
powerIO11.output_off(Pin2_8_DACon);
powerIO11.output_off(Pin16_SMUon);
end;
end;
Pin LIN

7.1 LIN recessive output voltage

Function Lin_Routput_voltage(VBattery:real):real;
var
value:real;
Begin
With TJAJ020meas do
Begin
setSMUvoltage(1,GPIBadres_SMU1,VBattery,10e-3);
DAC1.sendvolt(pin4_pin10a_DAC,5);
setSMUvoltage(2,GPIBadres_SMU2,0,12);
PowerIO11.output_on(Pin14_SMUon);
sleep(100);
value:=Trunc(measureSMUvoltage(2,GPIBadres_SMU2)/VBattery*1000)/1000;
VORLIN.caption:=floattostr(value)+' VBat';
setSMUvoltage(1,GPIBadres_SMU1,0,10e-3);
PowerIO11.output_off(Pin15_9a_SMU1on);
DAC1.sendvolt(pin4_pin10a_DAC,0);
PowerIO11.output_off(Pin4_10a_DACon);
PowerIO11.output_off(Pin14_SMUon);
end;
end;

a. High-level leakage current

Function High_leakagecurrent_LIN(VBattery:real):real;
var
value:real;
Begin
With TJAJ020meas do
Begin
setSMUvoltage(1,GPIBadres_SMU1,VBattery,10e-3);
powerIO11.output_on(Pin15_9a_SMU1on);
setSMUvoltage(2,GPIBadres_SMU2,VBattery,1e-3);
PowerIO11.output_on(Pin14_SMUon);
sleep(100);
value:=measureSMUcurrent(2,GPIBadres_SMU2)*1e6;
ILHLIN.caption:=floattostr(value)+' uA';
setSMUvoltage(1,GPIBadres_SMU1,0,10e-3);
powerIO11.output_off(Pin15_9a_SMU1on);
setSMUvoltage(2,GPIBadres_SMU2,0,10e-3);
powerIO11.output_off(Pin14_SMUon);
end;
end;
H. Test mode

There are two test mode controls signals available from NSLP block for digital TCB: tcb_en and tcb_hold. The tcb_en goes high if voltage NSLP is lower than \( V_{tcb\_en} \) (typically -8V) and the tcb_hold goes high if voltage NSLP is lower than \( V_{tcb\_hold} \) (typically -11V).

<table>
<thead>
<tr>
<th>Code</th>
<th>Test mode</th>
<th>LIN</th>
<th>INH</th>
<th>TXD</th>
<th>RXD</th>
<th>NWAKE</th>
</tr>
</thead>
<tbody>
<tr>
<td>N.A.</td>
<td>TXB mode</td>
<td>I</td>
<td>I</td>
<td>IO(default I)</td>
<td>IO(default O)</td>
<td>I</td>
</tr>
<tr>
<td>10001</td>
<td>OSC</td>
<td>Tcb_clk</td>
<td>Tcb_tdi(I)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10100</td>
<td>RxLp</td>
<td>OSC on/off</td>
<td>OSC signal</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10101</td>
<td>Filter</td>
<td>Vtest</td>
<td></td>
<td>NSLP_filter_tst</td>
<td>NWAKENSLP</td>
<td>Vtest</td>
</tr>
<tr>
<td>10110</td>
<td>POR</td>
<td></td>
<td></td>
<td>POR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11001</td>
<td>Vbat_mon</td>
<td></td>
<td></td>
<td>Vbat_mon</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11010</td>
<td>Tempprotect</td>
<td></td>
<td></td>
<td>TXD</td>
<td>Overtemp</td>
<td></td>
</tr>
<tr>
<td>10011</td>
<td>Info</td>
<td></td>
<td>select</td>
<td>Bit_b</td>
<td>Bit_a(slope)</td>
<td></td>
</tr>
<tr>
<td>11101</td>
<td>Tx_rec</td>
<td>Vtest</td>
<td>INH active</td>
<td>TXD</td>
<td>Recsw_test(I)</td>
<td></td>
</tr>
</tbody>
</table>

Table 1 Test mode overview

OSC

Oscillator on/off via TXD pin (TXD = 1 -> osc on)
Monitor the output signal of the oscillator via RXD pin. Check the frequency [256Hz].
Check the start up & turn off behavior if possible.

**DATA Line use RXD**

**Clock Line use TXD**

**Procedure OSC_clk;** //program for shifting bits

```pascal
var
j:integer; i:integer;
begin
if OSCM_Checked then begin
  powerIO11.output_on(Pin2_testvoltageOn);
  OSC_clk;
end
else begin
  DAC1.sendvolt(clock_out,0);
  DAC1.sendvolt(Pin1_data,0);
  DAC1.sendvolt(data_out,0);
end;
```

The frequency can be checked in Oscilloscope.
RXLP

Test wake-up threshold of rcv & wu42. Apply the voltage source at LIN pin and monitor TXD pin and RXD pin.

<table>
<thead>
<tr>
<th>LIN</th>
<th>wu_on (TXD)</th>
<th>rcvlp (RXD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vbat</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Vbat – 2V</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0.6 * Vbat</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0.4 * Vbat</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Filter

Test NWAKE and NSLP filter.

- Part can be in any mode
- Test NWAKE threshold and the output filter. Ramp the voltage at NWAKE and monitor the RXD pin
- Threshold of NSLP filter can't be measured in test mode as NSLP pin is locked for tcb, NSLP filter is tested via nslp_filter_tst (= TXD) by changing the level at TXD and monitoring the output at RXD pin
- Testoutput (= RXD) is defined as the OR of the output of NWAKE filter and NSLP filter

<table>
<thead>
<tr>
<th>NWAKE filter</th>
<th>NSTXD</th>
<th>RXD (with external pull up)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>high</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>high</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Vbat_mon

Ramping down the voltage at VBAT, read output of vbat_mon block [NLOWBAT] via RXD.
NLOWBAT = 0 if Vbat < Threshold.
According to the design, the threshold is around 4.5V.
Part can be in any mode, test mode turns on vbat_mon block.

POR

This test mode is to check the POR threshold if Vbat.
Ramp the Vbat, and monitor RXD pin.
RXD is HIGH before POR.
RXD pulled down to LOW on POR -> if vbat too low pull down will be limited.
Info

This test mode is used to read out 2 bit ID, slope and bias error.

INH function as select

<table>
<thead>
<tr>
<th>TXD</th>
<th>RXD</th>
<th>INH</th>
</tr>
</thead>
<tbody>
<tr>
<td>l_bias_error</td>
<td>Low_slope</td>
<td>0</td>
</tr>
<tr>
<td>RomB</td>
<td>RomA</td>
<td>1</td>
</tr>
</tbody>
</table>

(Value for N1A: \{RomB, RomA\} = \{0,1\})

RomA, RomB, low_slope are derived from POR, As a double check purpose, after reading out the value, the voltage at Vbat can be ramped down.

Once Vbat is lower than Vpor, the read out bits will be inverted.

Tx_rec {TX recessive switch test}

Activate TX and RTLIN
LIN can be driven via TXD and no txddom time out
Txrex_sw signal controlled via RXD (input)

Tempprotect

TempHigh =1 if temperature is too high.

Drive TEMptest (test pad) with a small current, lower than mentioned in the table below and increase it with steps until TEMPHIGH [pin RX D] changes from low to high and measure the TEMPTEST current.
I. Curvetrace

Many electronic components, especially microchips, can be damaged by Electrostatic discharge (ESD). ESD is one of the leading causes of failure in power modules and integrated circuits. A strong ESD susceptibility can generate damage in the device that reduce its performance or destroy it. All MOSFET and IGBT power modules are sensitive to ESD because the thickness of the gate isolation only amounts to some ten nanometers. Sensitive components need to be protected during and after manufacture, during shipping and device assembly, and in the finished device.

ESD can occur in a variety of forms. One of the most common is through human contact with sensitive devices. Human touch is only sensitive on ESD levels that exceed 4,000V. In order to avoid this problem, it is necessary to do a curve trace test to check whether the pin has problems of ESD.

In the program, all pins except the GND pin will be connected to the ground during the test, the SMU used for the pin will switch on and sweep voltage from -5V to +5V. When the chip has not plugged in the hardware board, it is required to check whether the board has other diode or resistor connect to that pin. As the picture shows above, when nothing is connected, the curve should be flat with no current. After checking every pins and make sure that there is no problems, the chip will be plugged in.

ESD is avoided by preventing the buildup of static electricity. An ESDS and bodies within proximity of about one meter of the ESDS should be grounded so that electrostatic charge is bled off. Discrete ESD-protection diodes have become necessary to guarantee sufficient system-level ESD protection.

A diode is the most basic semiconductor device. It is made from a p-type and an n-type junction and has two terminals: an anode at the p-type and a cathode at the n-type. When a large enough voltage is applied from the cathode to the anode (reverse biasing), the diode enters its breakdown region and, in theory, can conduct an infinite amount of current at zero resistance. A voltage applied in the other direction (forward biasing) causes the diode to enter its forward-conducting region. These
Diodes can respond to a high ESD voltage very quickly and clamp thousands of volts to just tens of volts in a matter of nanoseconds by shunting the ESD current to ground.

The right picture shows the IV curve of a basic diode with the anode grounded and voltage swept across the cathode. This is applied in the EDS protection.

This is the curve trace on pin RXD for TJA1020. When the voltage sweeps from -5V to -0.7V, when the diode conducts, then it goes from forward bias to reverse bias, the current will become 0mA.
J. Schematic design in Altium board

Top Level

PCB layout