

1

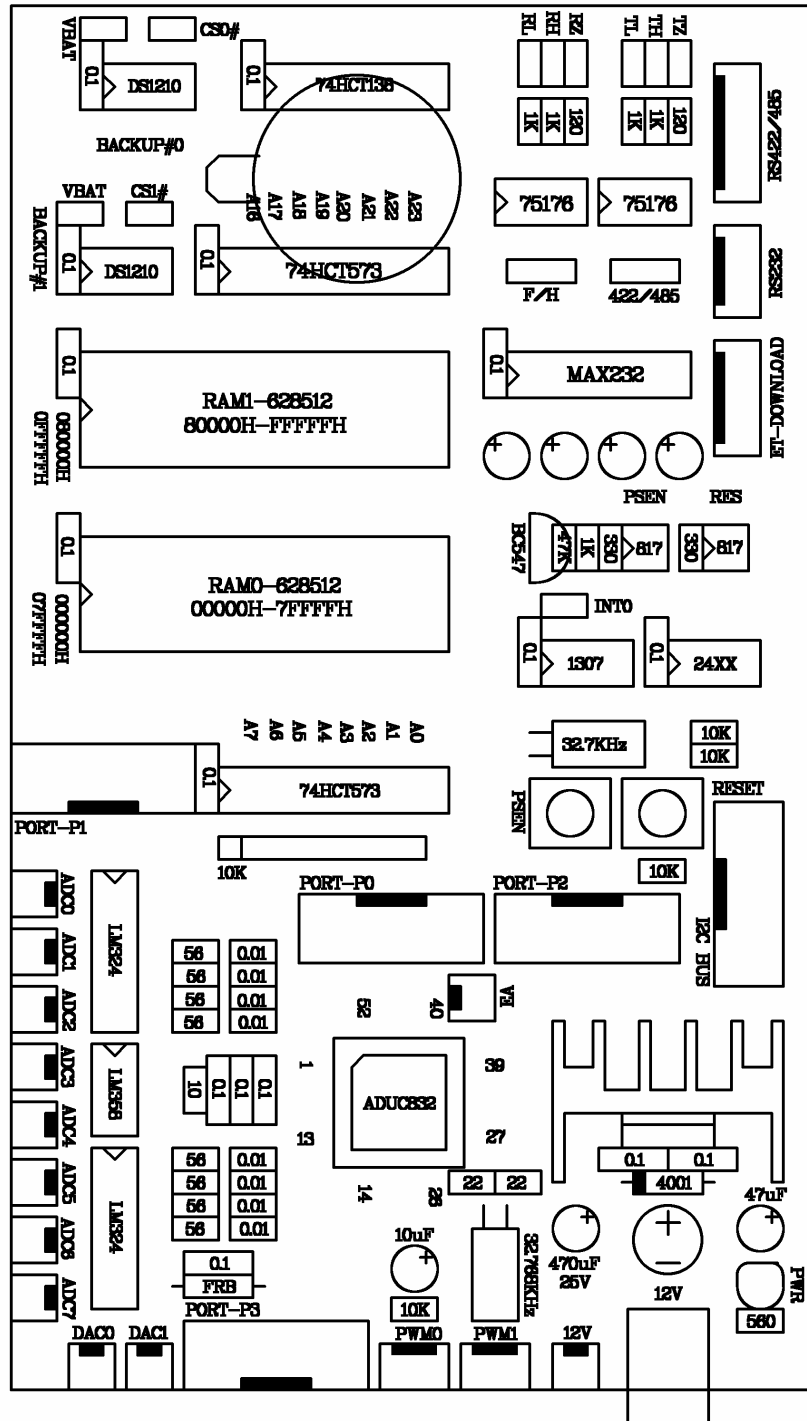
General Features of Board

CP-JR51-ADU832 V1.0

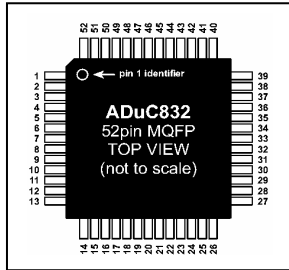
General Feature

Board Microcontroller CP-JR51-ADU832 V1.0 which uses Microcontroller No.ADuC832 and internal Flash/EE Memory can use qualifications absolutely such as **8-ch. Multichannel 12-bit ADC, 2-ch. 12-bit DAC, 2-ch. Output 16-bit PWM, Serial UART, I²C, SPI and external RAM with maximum 16 Mbyte** (for Board CP-JR51-ADU832 can access into external memory maximum 1 Mbyte). Moreover, there's other interesting features, all of them is included in one Chip ADuC832 and we will mention in the next content. Operation of MCU AduC832 is Microcontroller 8052, so we can use command 8051 to develop program but only there are additional Register and Functions.

Even if this Board Microcontroller uses Chip Microcontroller No. ADuC832 that has many functions, but some functions have to interface with external devices such as PCB for external RAM, PCB for Memory 24xx and RTC that are communication on I²C Bus and etc. So, ETT designs demonstrative Board to be more convenient and easier to develop program in the future.



Picture 1.1 Show structure of Board CP-JR51-ADU832 V1.0.



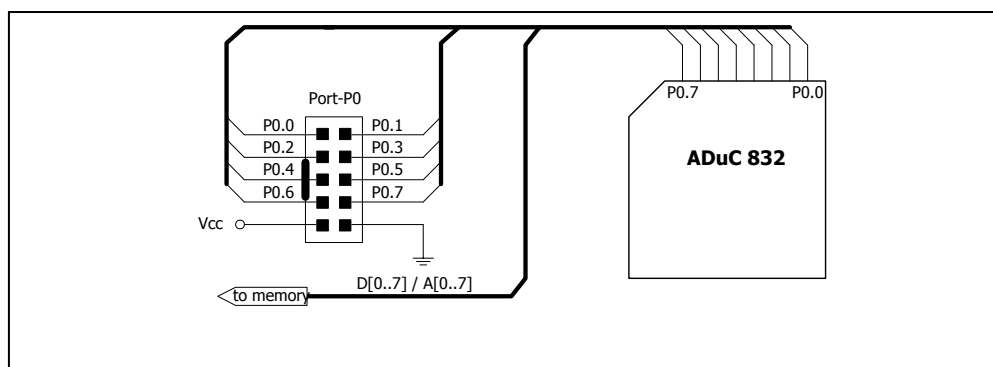
2

For Using Ports and Resources on Board CP-JR51-ADU832

I/O Selection of Board CP-JR51-ADU832 is designed the most flexible circuit, so user can use the board or modify the least structure of Board. We provide various Port I/O function for Chip MCU ADuC832, so user can use as required. User can configure value of some Pins from program but user can configure value of some Pins from Jumper on-board. MCU ADuC832 has 4 Port pins of Chip i.e., P0, P1, P2, P3. ETT provides these Port Pins for actual using and has other ports on board are described below.

Using Port-P0

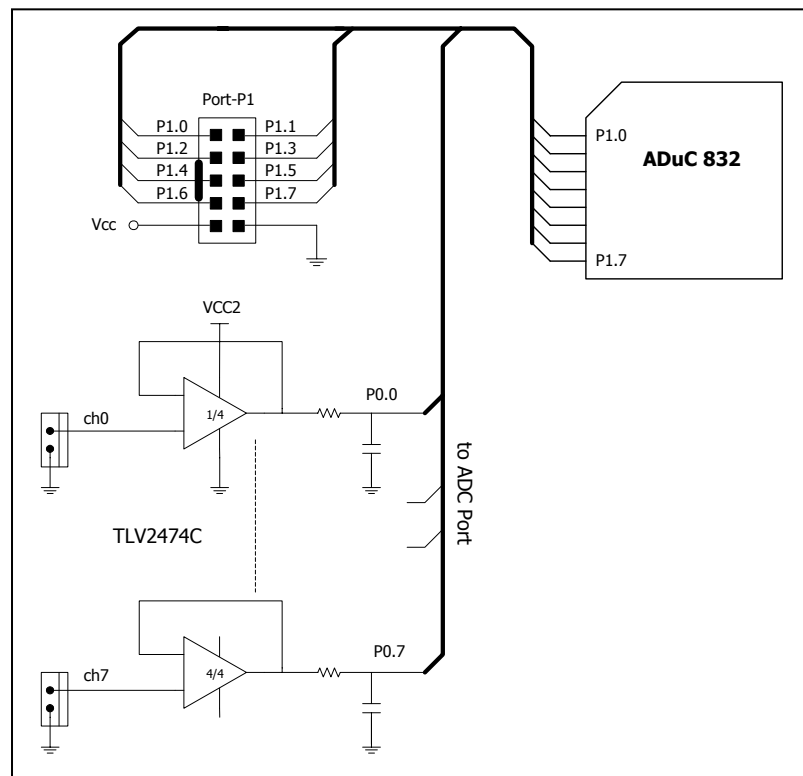
We can use this 8-bit Open Drain Port to be Input or Output depends on application. After reset the value of Port to be FFH and we can use command **MOV P0, #XXH** in case of we want to change value of the Port. Moreover, we can use Port P0 to be Low Address Bus or Data Bus when using command **MOVX** to communicate with External Memory. The P0 Port connection is shown in the picture below.



Picture 2.1

Using Port1-P1

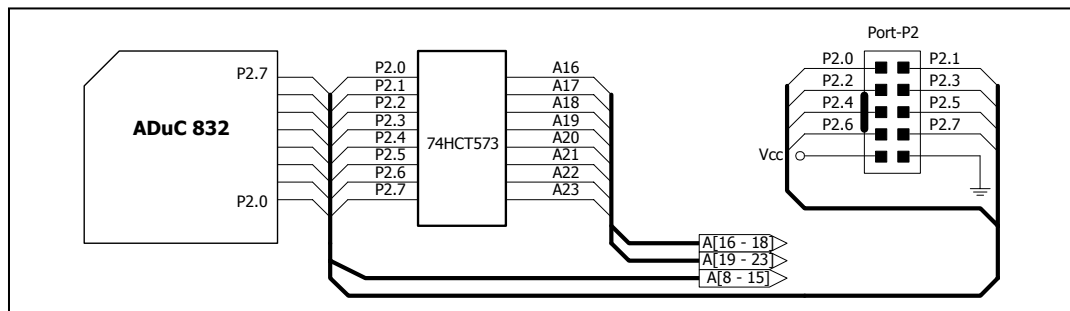
It is 8-bit Port and we can use this Port to be Input only. After reset board, the value of Port is FFH. The circuit of board is designed into 2 parts: one for interfacing with ADC 8 Channel circuit and other one for Port-P1 interface in case of user uses this Port to be Input. The Port P1 interface is shown in the picture below.



Picture 2.2

Using Port-P2

We can use 8-bit Port-P2 that has internal Register Pull-up to be Input or Output depends on application. After reset the port, the value of Port is FFH. The circuit of board is designed into 2 parts: one for general purpose interfacing with Port-P2 and other one for using High Address Bus and Page Address Bus in case of communication with External memory. The Port-P2 interface is shown in the picture below.

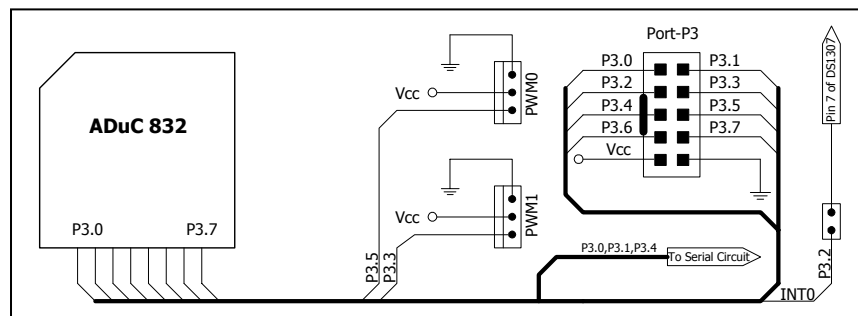


Picture 2.3

Using Port-P3

We can use 8-bit Port-P3 that has internal Register Pull-up to be Input or Output depends on application. After reset the port, the value of Port is FFH. Some circuit is designed interfacing Port-P3 for general purpose, so we do not interface bit P3.0 and P3.1 with Port-P3 because we use only Serial communication. Moreover, some Pins are selected for other various function as following:

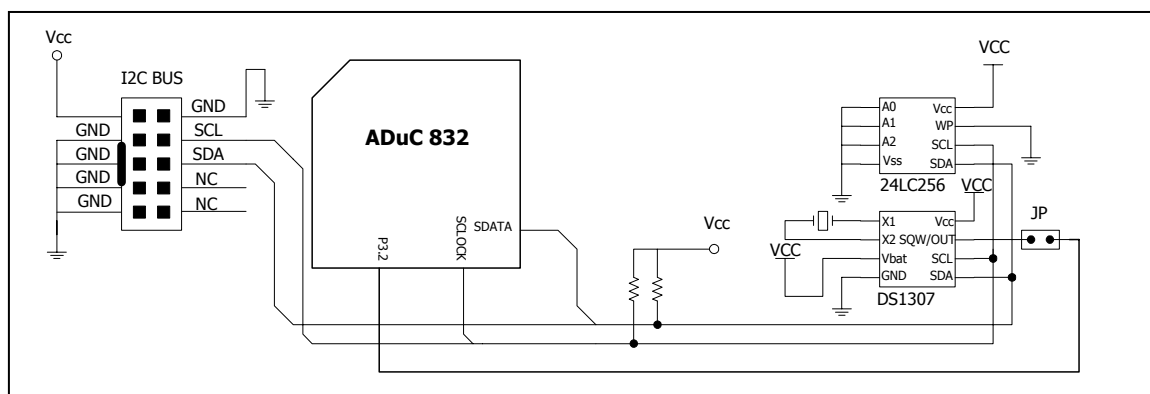
- P3.2 function is $\overline{\text{INT0}}$ Pin to interface with Pin 7 of IC DS1307 which is IC RTC (Serial Real Time Clock).
- P3.3 function is interfacing with Port PWM1.
- P3.5 function is interfacing with RS422/485 Serial Port.



Picture 2.4

Using I2C BUS Port

MCU ADuC832 provides functions for using I^2C on-Chip. There's 2 pins: SDATA Pin for sending/receiving data and SCLOCK Pin for sending Clock and controlling the sending/receiving data on I^2C Bus. Moreover, there's IC 24xx that is EEPROM and IC DS1307 that is IC RTC (Serial Time Clock) on board. Additionally, we can use I2C port for adding devices on I2C BUS. Type of pin and circuit arrangement are shown in the picture below.



Picture 2.5

From the picture 2.5 above, in the part of EEPROM Address is set at address "000" by notice Pin A0, A1, A2 is interfaced into Ground.

Using I2C Devices for Interrupt

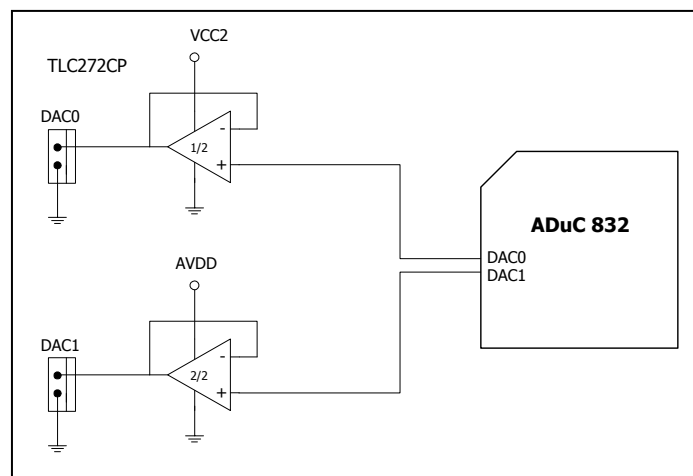
Signal Interrupt of I^2C devices, due to the signal receiving circuit of board is designed to receive external Interrupt (INT0) and is interfaced with Interrupt of RTC Pin, Board CP-JR51-ADU832 has jumper for connection or disconnection signal Interrupt at pin of RTC devices. So, in case of using Interrupt with RTC devices, must interface Jumper circuit as in the picture 2.5. In this case I2C Bus Port from ETT does not support external Interrupt.

Using ADC Port

It is 2 pins 8 channels ADC Port internal Board is interfaced through Buffer Op-Amp of Pin P1.0 - P1.7 of MCU ADuC832. One Pin is GROUND Pin and other one is Buffer that is interfaced to Op-Amp. Pin position and ADC circuit interface are shown in the picture 2.2. This Port function is transforming signal external Analog into signal Digital.

Using DAC Port

It is 2 pins DAC Port: one Pin is GROUND Pin and other one is Pin Output that is interfaced from Output of Op-Amp and its function is Buffer. Internal Board has 2 DACs or 2 channels i.e., DAC0 Port and DAC1 Port that are interfaced from Pin DAC0 and Pin DAC1 of MCU ADuC832. DAC circuit interface is shown in the picture 2.6. This Port function is transforming signal Digital into signal Analog (Output).



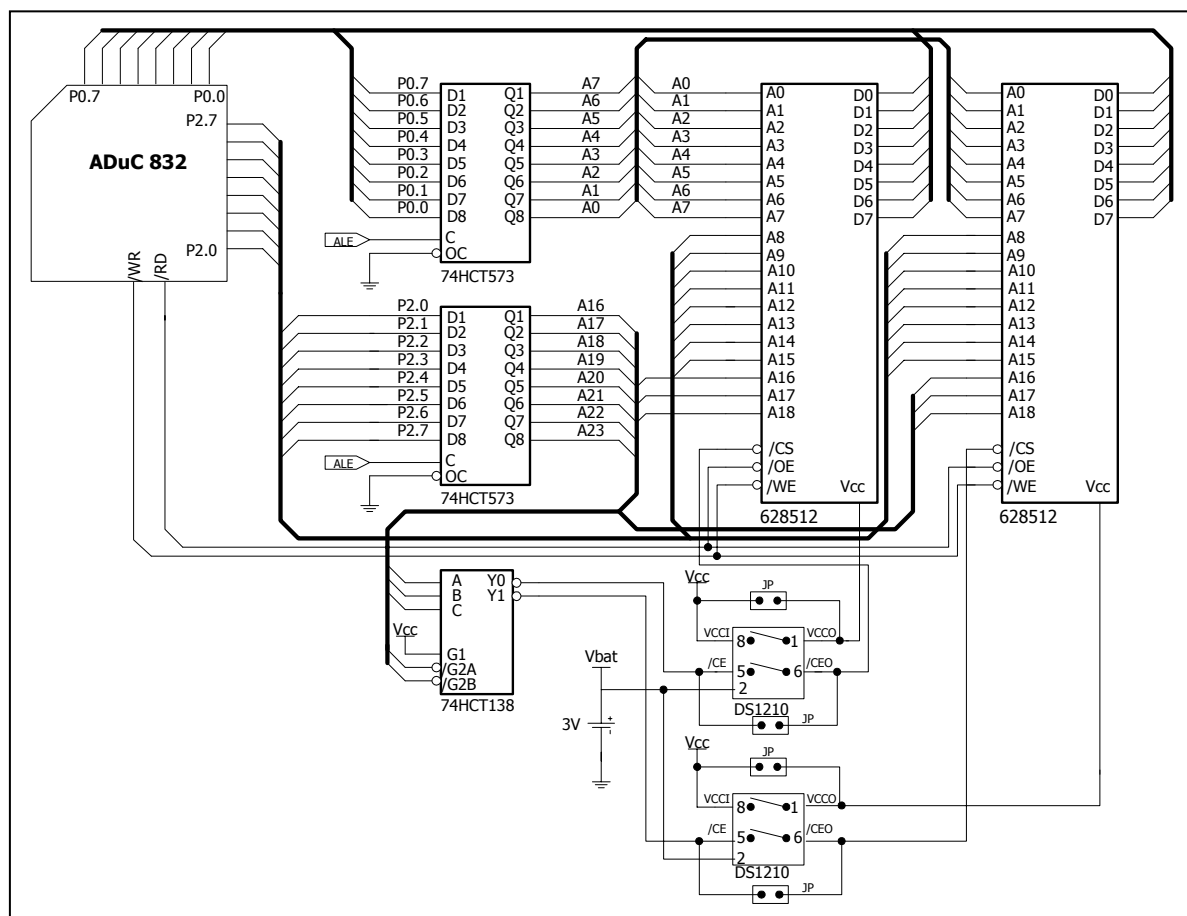
Picture 2.6

PWM Port

It is 3 pins PWM Port: Signal Pin, Anode Pin and Ground Pin. Internal Board has 2 PWMs: PWM0 and PWM1. Pin arrangement and PWM circuit are shown in the picture 2.4. This Port function is generating Pulse width Modulator (PWM) with 6 modes selection.

External Memory

The remarkable specification of the MCU No. is using 16 Mbyte External Memory connection but Board CP-JR51-ADU832 is designed to support External Memory only 1 Mbyte. For circuit arrangement, we use Port 0 to be Multiplex of signal DATA Bus and Low Address Bus, and Port 2 to be High Address and Page Address. Its circuit interface is shown in the picture 2.7.



Picture 2.7

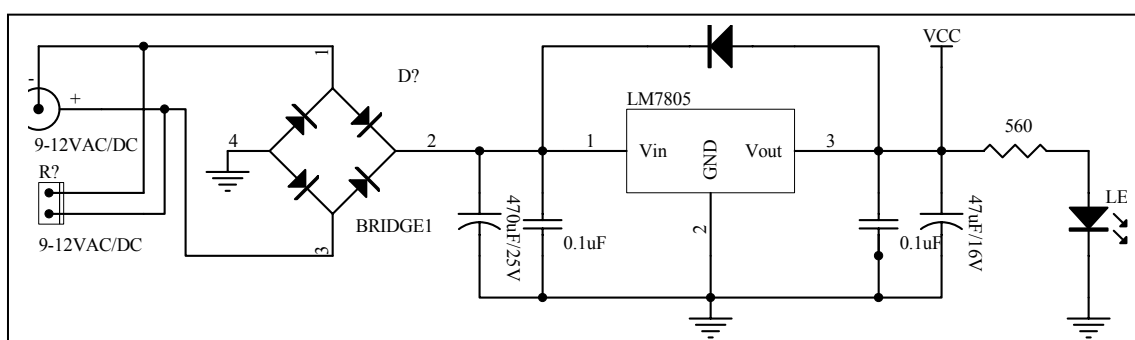
From picture 2.7, it connects external Memory by using two 512 KByte memory No.628512. Internal circuit has two IC 74HCT573 to be latch function at the position Address value of External Memory connection. Moreover, there's Back up circuit on board by using IC DS1210 to be detection and compare status of Power Supply as following:

- It is power supply selection between VCC and Battery Back Up. IC DS1210 will detect both Voltages and it will select the highest one for supplying to External Memory. It has internal Voltage Drop is less than 0.3V.
- It detects Voltage of VCC is in Power-fail status or not. Status value of Pin Chip Enable (CEO) is remained, in case of finding Power-fail status.
- The good Power-fail status protection of Pin Input \overline{CE} , Pin Output \overline{CEO} must save voltage range between 0.2V of Voltage VCCI or of Battery. The Voltage range of Power-Fail status detection is between 4.75-4.5 Volt, when Pin3 is interfaced into Ground and the Voltage range of Power-Fail status detection is between 4.5-4.25 Volt when Pin3 is interfaced at VCC0.

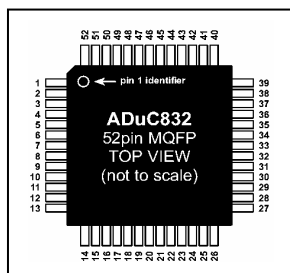
Moreover, user can read more detailed IC DS1210 operation from Data Sheet. In case of not using IC DS1210, user can select Jumper on board and must use all 4 Jumpers.

Power Supply Connector

We use external 9-12 VAC/DC Power Supply Connector. It can use both DC and AC because board provides 2 types of Connector depends on application as in the picture 2.8.



Picture 2.8



3

Using RS232/RS422/RS485 Serial Port Communication

There's UART Serial communication circuit internal MCU No. ADUC832 for using with Board CP-JR51-ADU832. We can use the circuit for writing program to control MCU communication with other devices as required such as Baud Rate, Data Bit, or Parity. In this case we do not mention about details, user can read from User manual of Hardware Architectonics or Data Sheet of MCU No. ADuC832.

Normally, sending/receiving data Pin of MCU can be interfaced with sending/receiving data Pin of other devices. To interface TX sending pin Of MCU with RX receiving pin of connective device and to interface RX receiving pin of MCU with TX sending pin Of connective device. Remember, TX and RX Pin of MCU can be only interfaced with TTL Logic level. In case of interface sending/receiving signal of MCU with device directly, it can be only communicated in the short distance or in the same internal circuit. So, Board CP-JR51-ADU832 designs Line Driver circuit to be Buffer for transforming electric level of sending/receiving data of MCU into TTL Logic, it can send/receive data for more long distance. We can select 3 modes of Serial communication from Line Driver circuit as following:

RS232 Serial Communication

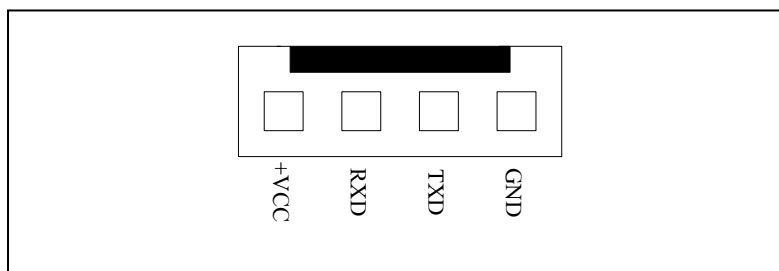
In this case, we must install IC Line Driver to transform electric level of sending/receiving TTL data of MCU (RX and TX) to be RS232 ($\pm 12V$). To install IC No. MAX232 to transform TTL level from TX sending Pin of MCU to be $\pm 12V$ level for sending to RX receiving Pin of external device, on the other hand, it transforms RS232 TX sending signal of external device to be TTL level for sending to RX receiving Pin of MCU. After transformed TTL sending/receiving signal level into RS232, we can send/receive electric data to external device more long distance, approximately 50 feet or 15 m. It can be only sending/receiving data with devices as Point-To-Point.

We only use 2-3 cables for RS232 Serial communication depends on the one directional or bidirectional communication.

- **Bidirectional RS232 Communication:** It can be sending/receiving data between receptor and sender. In this case, we must use 3 cables : RXD for receiving data, TXD for sending data and GND for reference signal. For cable connection, the cable must be crossed with destination device. For example, TXD sending signal from Board CP-JR51-ADU832 must interface with RXD receiving signal of destination device and TXD sending signal of destination device must interface with RXD receiving signal of Board. In the part of GND reference signal of Board and destination device must interface correspondingly, so it can be sending/receiving data completely.
- **One Directional RS232 Communication:** It maybe only wait for receiving data or sending data to destination device but not respond to each other. We only use 2 cables. For sender, we must only interface TXD sending signal and GND reference signal and for receptor, we must only interface RXD receiving signal and GND reference signal.

RS232 Connector of Board CP-JR51-ADU832 is connection point of sending/receiving data that is transformed into RS232 signal level. it is 4 Pin CPA Connector to be a

connection point of sending/receiving data with external device and signal arrangement is shown in the picture 3.1.



**Picture 3.1 displays RS232 Connector
of
Board CP-JR51-ADU832 V1.0.**

RS232 Connector of Board has 4 cables but for general sending/receiving data uses only 3 cables i.e., RXD, TXD and GND. In this case, we do not use +VCC that is +5V power supply for the communication. +VCC or +5V is designed to use in case of the destination device is a small circuit and can not supply power into destination device. So, we can interface +VCC power supply into it.

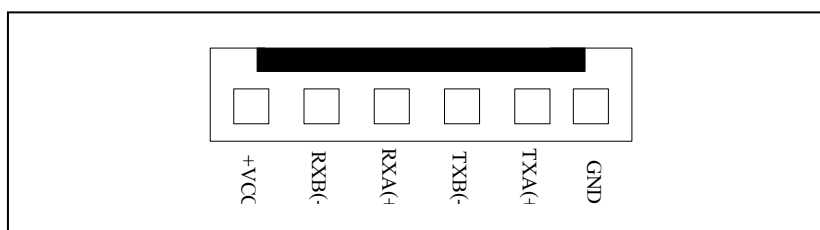
Note: For RS232 IC Line Driver is standard component of board CP-JR51-ADU832 that has already installed on board. Remember, user must install only one type of IC Line Driver to select electric Serial communication. For example, if user installs RS232 IC Line Driver and installs MAX232 on board, does not install other types of IC Line Driver such as RS422 or RS485 because it makes sending/receiving data error.

RS422 Serial Communication

In this case, we must install 1 or 2 IC Line Drivers No. 75176 or MAX3088 to transform electric level of sending/receiving TTL data of MCU (RX and TX) to be Balance Line. In case of using bidirectional communication, we must install 2 IC Line Drivers because it is divided into 2 parts: one for transforming signal on receptor side and other one for transforming signal on sender side. In case of using only one directional communication, we can install only one IC Line Driver. To assign only receiving data, only install IC Line Driver at the position of "RXD/422" and to assign only sending data, only install IC Line Driver at the position of "TXD/485".

We can use RS422 Communication instantly without any modification or change program. The advantage of RS422 is able to communicate data more long distance than RS232 approximately 4000 feet or 1200 m. or 1.2 km. Remember, must only use cable that is designed to support RS422 operation such as UTP (Un-Shielded Twist Pair) cable or STP (Shielded Twist Pair) cable. The RS422 communication is electric signal level communication, in case of using bidirectional communication (both sending/receiving data), it can be sending/receiving data with devices as Point-to-Point as same as RS232. In case of using only one directional communication, can interface many points of parallel signal on the receptor with only one point of sender. If we use IC Line Driver No. 75176, we can interface 32 points of parallel signal on the receptor but is we use IC Line Driver No. MAX3088, we can interface maximum 256 points of parallel signal on the receptor. On the other hand, we can not interface many points of parallel signal on the sender, we can interface maximum 1 point of parallel signal on the sender.

The characteristic of RS422 Connector is 6 Pin CPA as in the picture 3.2. We use 4 cables for sending/receiving data i.e., 2 receiving data cables: RXA(RX+) and RXB(RX-) and 2 sending data cables: TXA(TX+) and TXB(TX-). For cable connection, the cable must be crossed with destination device. For example, we must interface RXA(RX+) with TXA(TX+) and must interface RXB(RX-) with TXB (TX-).



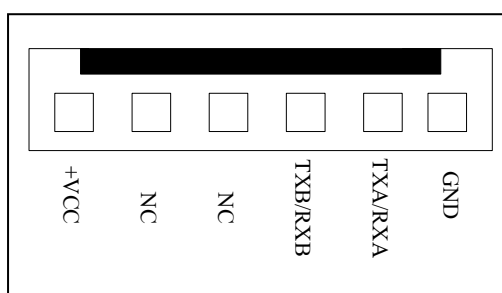
RS485 Serial Communication

The electric specifications of RS485 communication are as same as RS422 but RS485 communication uses only 2 cables and can adjust signal direction from program. We can configure function of both cables can be used as receptor or sender as require by using MCU. PTC3 which is directional controller function to be sender or receptor are controlled by Board CP-JR51-ADU832 as following. If PTC3 is configured to be '1', PTC3 function is sender and if PTC3 is configured to be '0', PTC3 function is receptor. So, RS485 communication can be interfaced many points of parallel devices in the same line. If we use IC Line Driver No. 75176, we can interface 32 points of parallel devices and if we use IC Line Driver No. MAX3088, we can interface maximum 256 points of parallel devices. Remember, in case of interface parallel device more than 2 points in the same line, must write program to control sending direction only 1 points. In case of we configure sending direction more than 1 point, it makes data clash and communication error.

When using RS485 communication, must only install IC Line Driver No. 75176 or MAX3088 at the position of "TXD/485" and then configure RS485 options as following:

- Select Jumper for "422/485" selection on the 485(RS485) side.
- Select Jumper "F/H" on H (Half Duplex) side.
- Short Jumper for Fail Safe Resister i.e., "TL".
- Short Jumper for Fail Safe Resister i.e., "TH".
- Interface only 2 cables from TXB(TX-) and TXA(TX+).

We must write program to support RS485 communication because data direction is directly configured by program. The advantage is using less sending/receiving cable but the communication takes a long time more than other method because it can not be sending or receiving data simultaneously. Nowadays, price of 2 cables and 4 cables are not different, so it is better to use RS422 communication because it is not difficult to write program to control sending/receiving data of MCU.



Jumper Configuration for RS422/485 communication

Line Driver circuit of Serial Port is designed many mode selections, so it must use Jumper Selection in case of using RS422 and RS485 communication as following.

- **Jumper 422/485:** It is Jumper selection to configure IC Line Driver at position of TXD/485 to be function of RS422 or RS485. If we configure IC Line Driver to be RS422 function, we must configure Jumper on "422" side and it makes IC Line Driver at position of TXD/485 to be only sender. On the other hand, if we configure IC Line Driver to be RS485 function, we must configure Jumper on "485" side and it makes IC Line Driver at position of TXD/485 to be both sender and receptor as controlled by PTC3.
- **Jumper F/H (Full/Half):** It is Jumper selection to configure types of communication to be Full Duplex (F) or Half Duplex (H). if we use RS422, we must configure Jumper on Full Duplex (F) side, on the other hand, if we use Rs485, we must configure Jumper on Half Duplex (H) side.
- **Jumper RL:** It is Jumper selection to configure resister connection and makes status RXB (RX-) stable or Fail Safe Resister, it makes RXB (RX-) stable when not sending any signal from line. In case of line is interfaced for short/long distance but not sending any signal from line, always should be Short Jumper, especially position of device at the beginning and the destination line. Other position that not far from at the beginning and the destination line, should be Open Jumper. Remember, user always should Short Jumper minimum 1 point with connective device in line.
- **Jumper RH:** It is Jumper selection to configure resister connection and makes status RXA (RX+) stable or Fail Safe Resister, it makes RXA (RX+) stable when not sending any signal from line. In case of line is interfaced for short/long distance but not sending any signal from line, always should be Short Jumper, especially position of device at the beginning and the destination line. Other

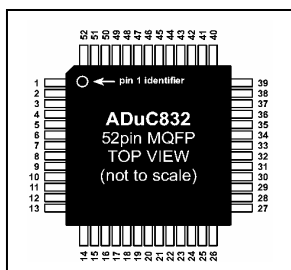
position that not far from at the beginning and the destination line, should be Open Jumper. Remember, user always should Short Jumper minimum 1 point with connective device in line.

- **Jumper RZ:** It is Jumper selection to configure RZ resistor connection and to compensate Impedance on receptor side. In case of line is interfaced for long distance, always should be Short Jumper because it makes resistor inside the long line. So, we must use external resistor to compensate Impedance. When we Short Jumper at the RZ position, it is resistor connection between RXA (RX+) and RXB (RX-). In case of interface line for short distance, should be Open Jumper.
- **Jumper TL:** It is Jumper selection to configure resistor connection and makes status TXB (TX-) stable or Fail Safe Resistor, it makes TXB (TX-) stable when not sending any signal from line. In case of line is interfaced for short/long distance but not sending any signal from line, always should be Short Jumper, especially position of device at the beginning and the destination line. Other position that not far from at the beginning and the destination line, should be Open Jumper. Remember, user always should Short Jumper minimum 1 point with connective device in line.
- **Jumper TH:** It is Jumper selection to configure resistor connection and makes status TXA (TX+) stable or Fail Safe Resistor, it makes TXA (TX+) stable when not sending any signal from line. In case of line is interfaced for short/long distance but not sending any signal from line, always should be Short Jumper, especially position of device at the beginning and the destination line. Other position that not far from at the beginning and the destination line, should be Open Jumper. Remember, user always should Short Jumper minimum 1 point with connective device in line.
- **Jumper TZ:** It is Jumper selection to configure TZ resistor connection and to compensate Impedance on sender side. In case of line is interfaced for long distance, always should be Short Jumper because it makes resistor inside the long line. So, we must

use external resistor to compensate Impedance. When we Short Jumper at the TZ position, it is resistor connection between TXA (TX+) and TXB (TX-). In case of interface line for short distance, should be Open Jumper.

Tip: In this case, we can see Line Driver circuit of both RS422 and RS485 are quite similar. The remarkable difference between RS422 and RS485 is: the sending/receiving data's direction of RS422 can not be changed by program because it is fixed by circuit, on the other hand, the sending/receiving data's direction of RS485 can be changed by program, so we can configure its function to be sender or receptor as required.

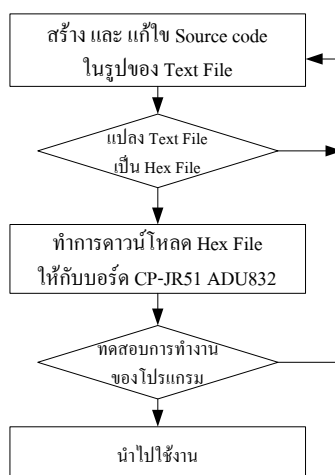
4



Development Program on Board CP-JR51-Adu832 V1.0

We use 2 programs for development program on board CP-JR51-ADU832: **Program WSD** and **Program Cross32 V4.0**. We use program WSD for downloading File program .HEX (Intel Hex Format) that is written on computer PC and is sent into Board CP-JR51-ADU832. On the other hand, Program Cross32 V4.0 is program Assembler which transforms File .ASM into file .HEX (Intel Hex Format).

User can be development program on board CP-JR51-ADU832 without using more component except computer PC and RS-232 Serial Port cable for interfacing between board CP-JR51-ADU832 and computer PC. The proceeding of development program is shown as in the Flow Chart:



Picture 4.1 displays the proceeding of development program of Board CP-JR51-ADU832.

Program WSD Installation

Program WSD is provided with demonstrative board and now it is newly 3.33 MB File version, user can download data or program WSD newly form website <http://www.analog.com/microconverter>. If user downloads program from website, you will see blank of **Select a resource** on the left hand of monitor, next select title **QuickStart Development to** and then move to title **Software Tools Updates** below, click [Windows Serial Downloader V6.02](#). It starts downloading program WSD.

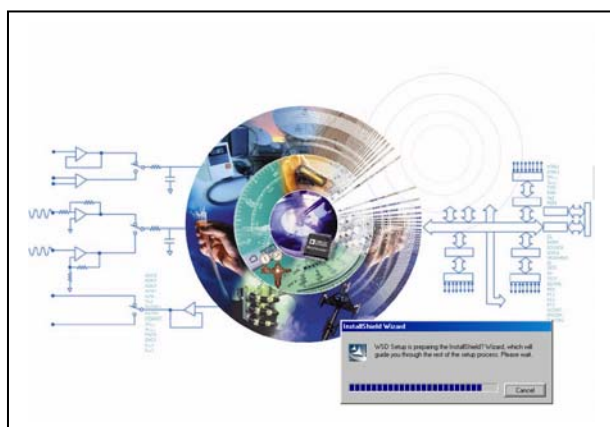
In this case, we would like to mention program installation from the provided CD-ROM in demonstrative board as following:

1. Insert CD-ROM into computer PC, next access into Folder WSD V6.02, and then double click file WSD_V6.02_setup.exe.

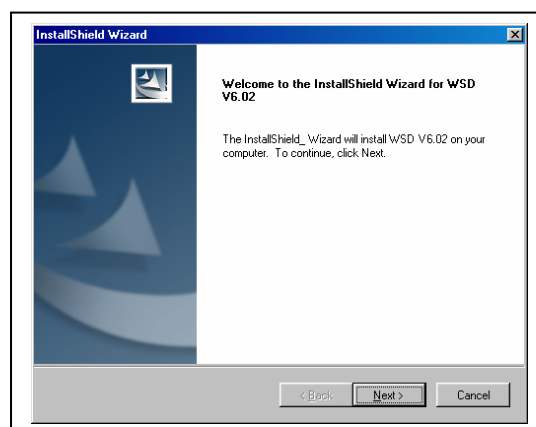


Picture 4.2

2. It will display message as below after double clicked file WSD_V6.02_setup.exe and then Click Next to start program installation.

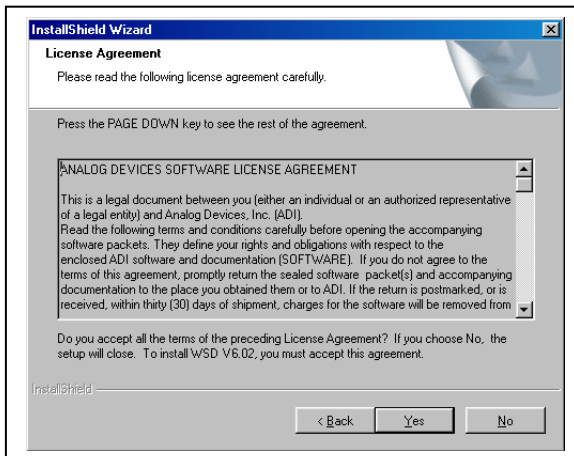


Picture 4.3

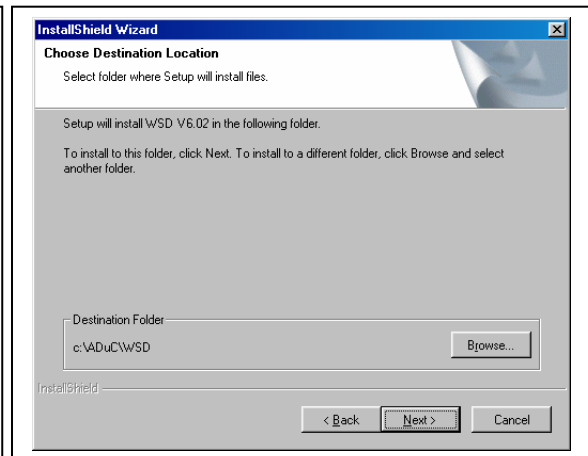


Picture 4.4

- It will display window of License Agreement that is conditions between user and company, In this case, click YES. Next, user must select destination position of setup program, if want to change position setup program, click Browse... In this case, we do not change, click Next.

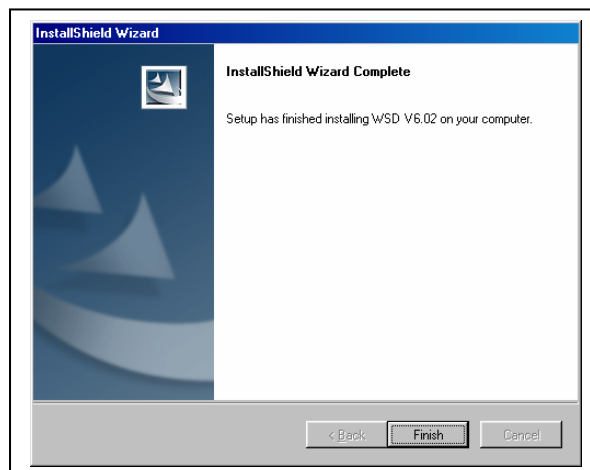


Picture 4.5



Picture 4.6

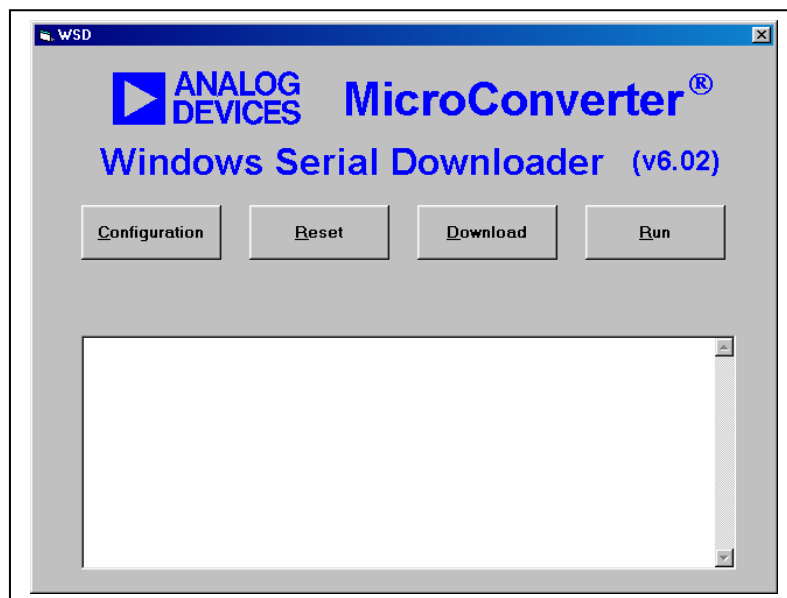
- It will display window of InstallShield Wizard Complete after the end of program installation, click Finish.



Picture 4.7

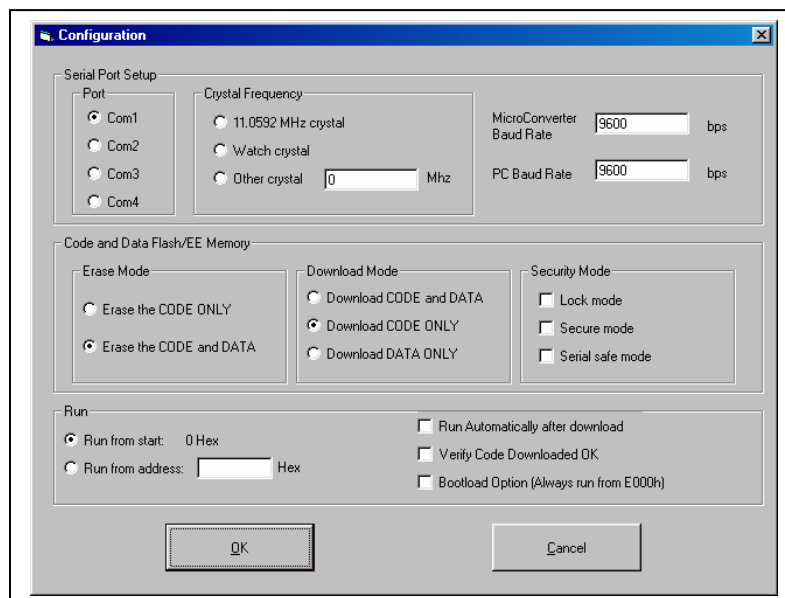
To setup value before using program WSD

When installed completely, oper program WSD by pressing Strat => Programs => ADuC => WSD as in the picture below.



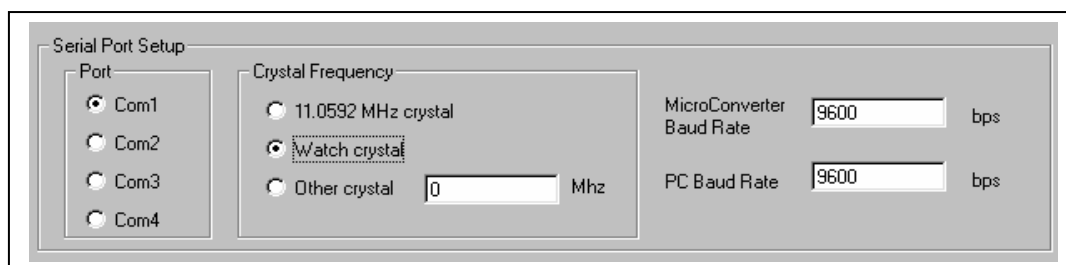
Picture 4.8

First of all, must setup Configuration by pressing Configuration and it will display window as in the picture below.



Picture 4.9

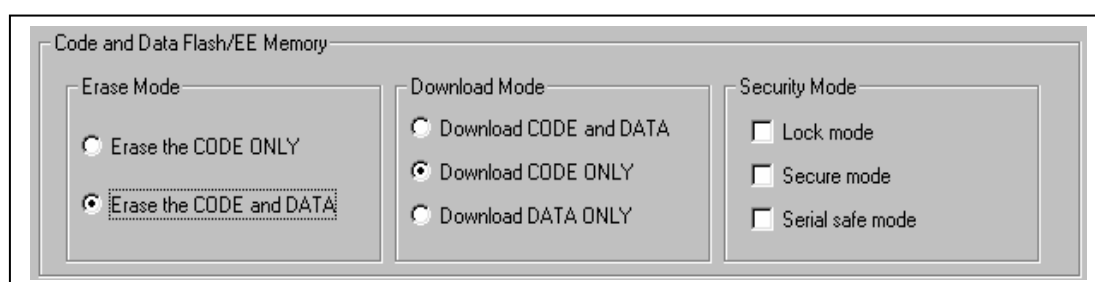
User must understand the meanings inside window Configuration well before adjustment its value. The first frame is Serial Port Setup with 2 sub-frame insides: **Port** and **Crystal Frequency**. The functions are shown as the below.



Picture 4.10

- **Port Frame:** It is assignment for RS232 Serial Port Communication and there's minimum 1 port inside computer PC i.e., Com1. some computer PC has more than 1 Port i.e., Com 2, Com 3. So, we must assign the true usage, in this case we use Com 1.
- **Crystal Frequency Frame:** It is frequency assignment. We use Watch crystal for Board version CP-JR51-ADU832.
- On the end of right hand, it is assignment Baud rate of Chip on board and computer PC. in this case, we use 9600 bps without any change.

The next frame below, it is Code and Data Flash/EE Memory which has 3 sub-frame insides: **Erase Mode**, **Download Mode** and **Security Mode**. The functions are shown as the below.



Picture 4.11

- **Erase Mode Frame:** There are 2 selections. First, **Erase the CODE ONLY** means that it will auto erase Flash/EE Program memory before downloading program.

Tip: If we store data in Flash/EE Data memory and must download new programs many times, we must select Erase the CODE ONLY. remember, do not select Erase the CODE and DATA, otherwise, data in Flash/EE Data memory is erased while downloading new program.

Second, **Erase the CODE and DATA** means that it will auto erase both Flash/EE Program memory and Flash/EE Data memory before downloading program.

- **Download Mode Frame:** There are 3 selections. First, **Download CODE and DATA** means that it is programming into both Flash/EE Program and Flash/EE Data Memory. Second, **Download CODE ONLY** means that it is only programming into Flash/EE Program memory. Third, **Download DATA ONLY** means that it is only programming into Flash/EE Data memory.
- **Security Mode Frame:** There are 3 selections i.e., Lock Mode, Security Mode and Serial safe Mode. In this frame, we can select more than 1 mode.

The last is **Run frame**, it is Run value adjustment such as beginning address adjustment for running program or auto run adjustment. Its details are shown below:

The screenshot shows a 'Run' frame with the following elements:

- Radio button selected: **Run from start: 0 Hex**
- Radio button unselected: **Run from address: [] Hex**
- Checkbox unselected: **Run Automatically after download**
- Checkbox unselected: **Verify Code Downloaded OK**
- Checkbox unselected: **Bootload Option (Always run from E000h)**

Picture 4.12

- **Run Frame:** There are 3 blanks to fill completely as following:
 1. **Run Automatically after download Blank:** If we tick in this blank, when download program into Board completely, it will auto run program.
 2. **Verify Code Downloaded OK Blank:** If we tick in this blank, it will download and verify program that is sent into the demonstrative board

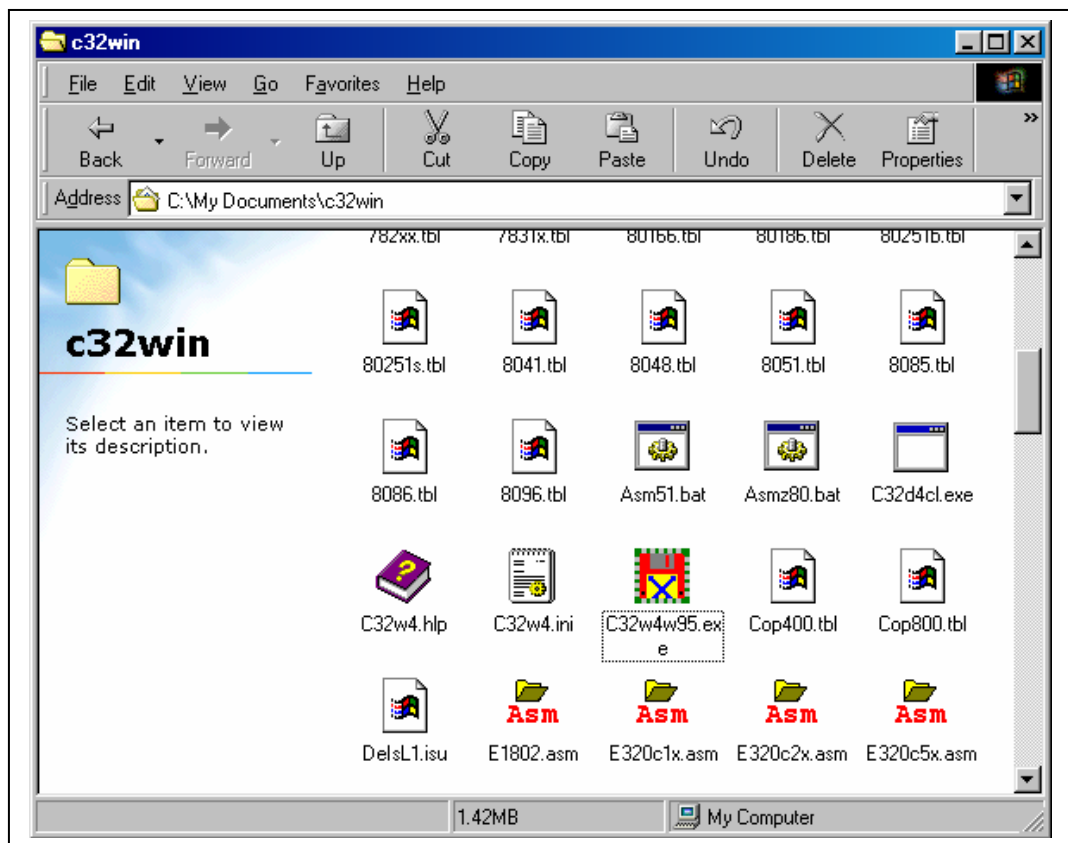
simultaneously. In this case, we do not select this blank.

3. **Bootload Option (Always run from E000H) Blank:**
If we tick in this blank, it will make the beginning address for running program at E000H. in this case, we do not select this blank.

Note: If we select type of Running Mode 2 and 3, program does not run until we click Run. If we select type of Running Mode 1, after downloading completely, it will be auto running program.

Program Cross 32 V4.0 Installation

Program Cross 32 is designed to be Program Cross-Assembler and is developed by Data Sync Engineering Co., Ltd. It is type of Universal Cross-Assembler program can compile commands of CPU MCS51, Z80 and other families. Program Cross 32 that is provided with board is demonstrative version and easy to install program. User only copies file program Cross 32 and paste in Drive C (or other) and double click file C32w4w95.exe that is the red icon when opening program as shown in the picture below.

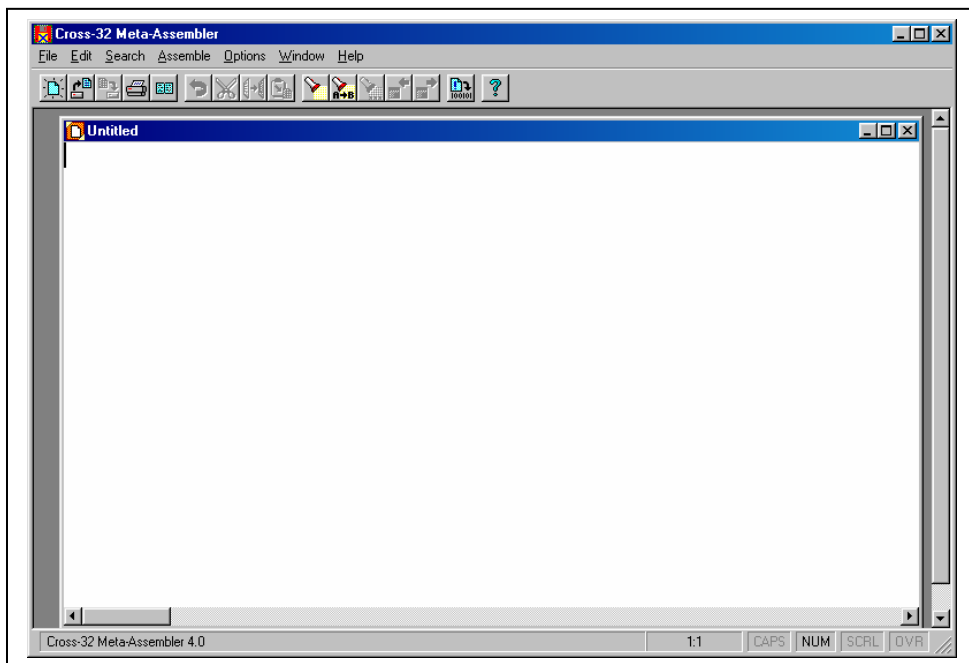


Picture 4.13

Remember, the folder that stores file program is .ASM, so must also copy file 8051.tbl in the folder because it will be error when compile file.

Using Program Cross 32 V4.0

Program Cross 32 is program Assembler with Text Editor insides. So, user can develop program by using this program or using NotePad or other Program text Editor. In this case, we will mention about using Program cross 32 approximately and using Text Editor inside program. User can read more detailed from the appendix C or HELP of program. When installed program Cross 32 completely, open program Cross 32, and then click **File => New**, it will display window as in picture below.



Picture 4.14

When we start writing program in the first two lines, must assign artificial commands of CPU No. and type of Output File into program because the program Cross 32 can be used with many CPU No. and it can create Output File for many Formats. So, demonstrative board CP-JR51-ADU832 uses MCU No. ADuC832 to be main 8052 operation (Core), we can use it with 8051 commands and method to write command as shown below.

CPU	"8051"	; To translate to be Code command of MCS-51.
HOF	"INT8"	; To create Output File to be Intel Hex 8 bit.

After two lines above, the next line is space for writing program and the principle of writing program is described below.

- To start writing program with artificial command `ORG 0000H`: It means that we must configure the beginning address of Memory program and follow by the beginning address. In this case, it is Address `0000H`. Remember, do not write at the end of left hand, should be spaced minimum 1 character.
- Configure the fixed value into variable by using artificial command `EQU`, can see the sample below and variable name is on the end of left hand (do not omit).
- We write Label to configure reference address of Memory program. Remember, must use colon ":" follow by Label name without omitting and in this case not allow to use character "&" for writing Label.
- For writing commands into program such as `MOV A,#3EH`, always should be spaced minimum 1 character on the left hand
- For writing program in the last line, must close with command `END` otherwise it may be error.

Example : The sample below is power running through Port 0 and running to the right hand.

```

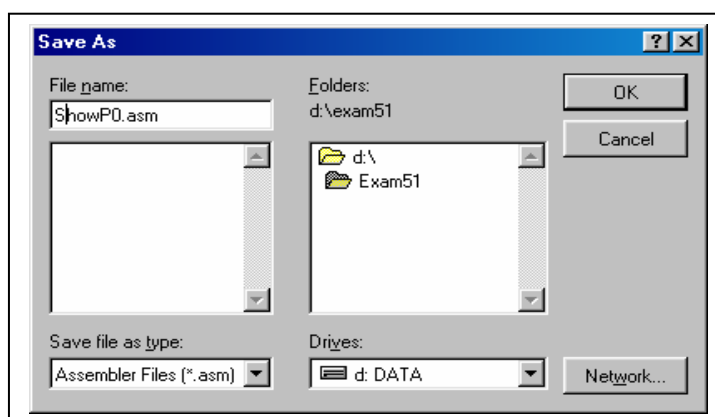
        CPU    "8051.TBL"    ; Processor declaration
        HOF    "INT8"        ; Intel 8-bit hexcode

        ORG    0000H        ; Reset vector
DATA    EQU    0FEH
START:  MOV    A,#DATA      ; DEFINE FIRST VALUE
loop:   RR     A
        MOV    P0,A
        MOV    R2,#100
L1:     mov    R3,#5h
L2:     mov    R4,#5h
        djnz  R4,$
        djnz  R3,12
        djnz  R2,L1
        sjmp  loOP

        END
```

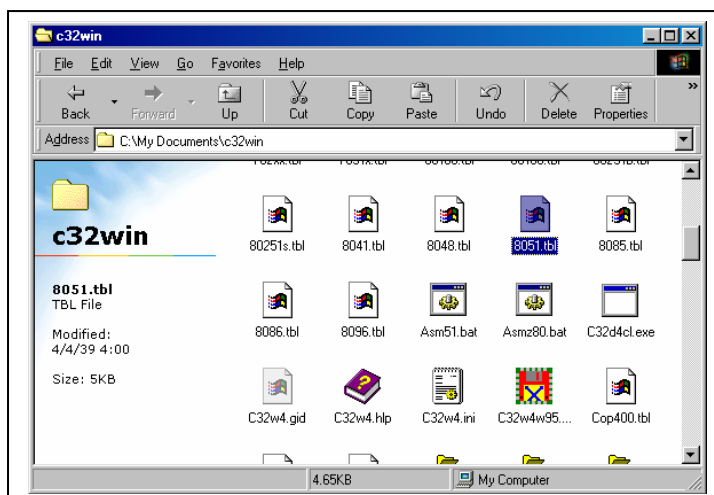
Picture 4.15 displays program ShowP0.asm.

After writing program completely, save file at drive D:\Exam51, configure name as ShowP0.asm in file name blank, and then click OK as in the picture below.



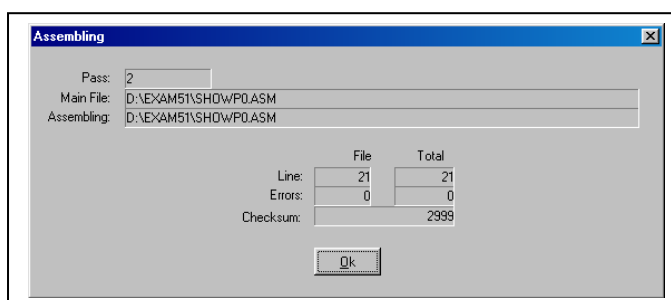
Picture 4.16 displays save File.

Then also copy file 8051.tbl in folder Exam51, this file is in folder c32win as in the picture below.



Picture 4.17

In this step, we can start translation program by selecting Menu **Assemble => Mail File...** to configure desired file and translate into .HEX. Next, select file D:\Exam51\ShowP0.asm and press OK. Select menu **Assemble => Assemble...**, it will display window as below, then click Assemble. In this case, user must notice in Error: blank must be 0. It means that the translation program is completely. In folder Exam51 has additionally 3 files i.e., ShowP0.ERR, ShowP0.HEX and ShowP0.LST. In this step, we get File .HEX.



Picture 4.18

Using Program WSD

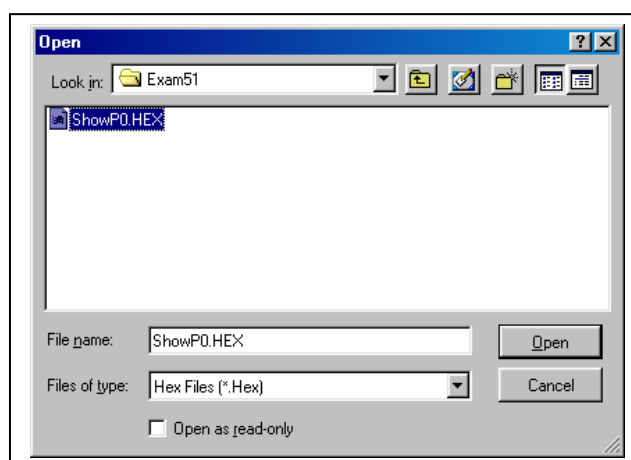
Next, it is using program WSD for downloading file ShowP0.HEX into demonstrative board CP-JR51-ADU832. We use this method to test operation program well or not and proceeding is described below.

Open program WSD.EXE, must be configurations first (user can read details from **"To setup value before using program WSD"** above). Then access into In-Circuit Serial Download Mode (user can read from **"Using In-Circuit Serial Download Mode"** on Page 91). Finally, click RESET on program WSD, it will display message in the window program as below if nothing is error.

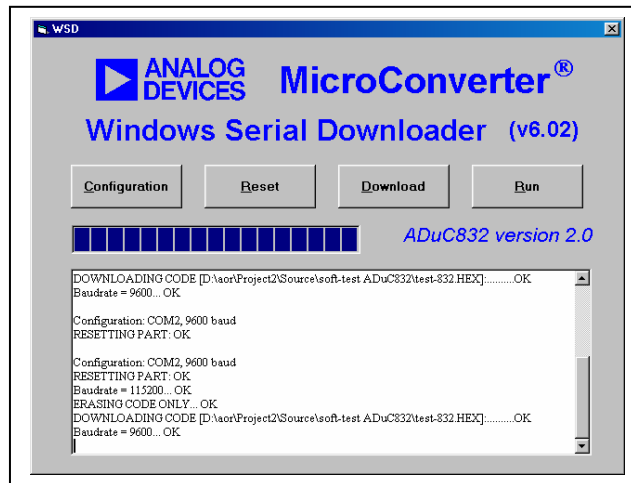
Configuration: COM1, 9600 baud

RESETTING PART: OK

Next, click Download, it will displays window Open, then select file ShowP0.HEX and click Open. In this step, Code program will be downloaded into demonstrative board and then click Run. Program will run instantly without click Run again in case of Setting Configuration in Run frame is Run Automatically after download.



Picture 4.19



Picture 4.20

Note: If we download new program in each time, we must also access into In - Circuit Serial Download mode.

Using In - Circuit Serial Download Mode

We must set Hardware status by using Switch PSEN and RESET to access into In-Circuit Serial Download mode for downloading program from computer PC into demonstrative board. The proceeding is described below.

1. Supply power into demonstrative board and interface RS232 download cable.
2. Press PSEN for a while.
3. Press RESET for 1 time (press and let button instantly) while remain pressing PSEN, then let PSEN.
4. Press RESET on program WSD.
5. It will display message on the window program of WSD as below in case of noting is error.

Configuration: COM1, 9600 baud
RESETTING PART: OK

Problems maybe occur while using Program WSD

For using program WSD, the connection between demonstrative board and computer PC maybe problems as following.

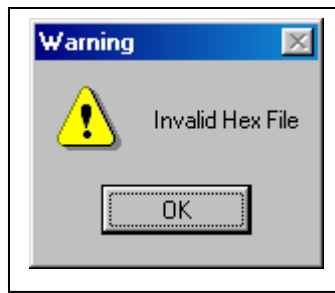
- The first window is Warning, it may be occur because of many causes.



Picture 4.21

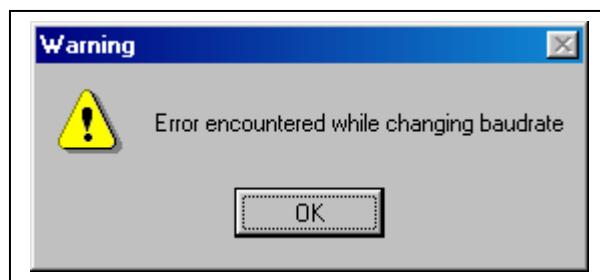
1. Not interface Power supply.
2. Not interface Download cable.
3. Not access into In-Circuit Serial Download mode.
4. Select incorrect Crystal frequency.

- The next warning window is Invalid Hex File because of the incorrect Hex File's type. So, user must check Hex File correct or not.



Picture 4.22

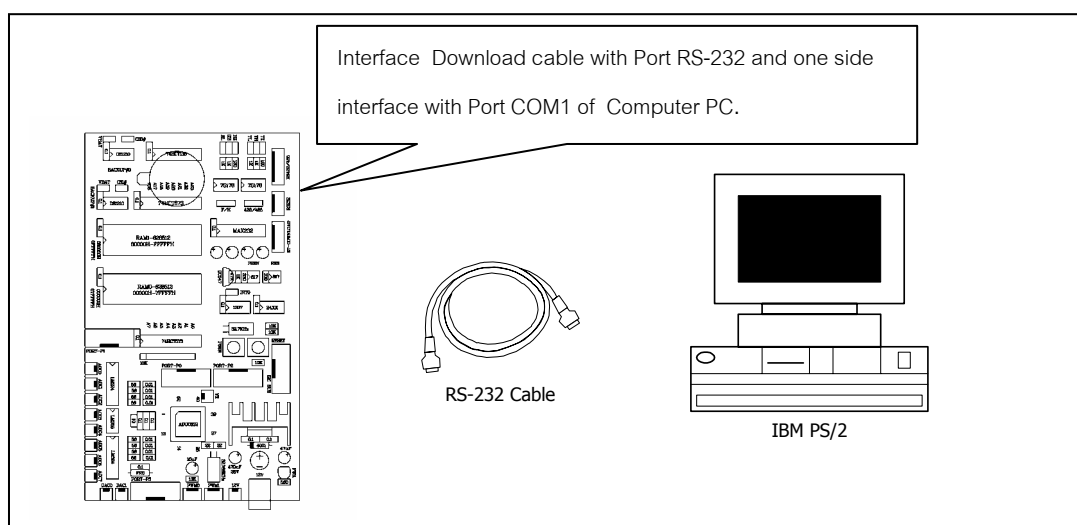
- The next warning window is Error encountered while changing Baud rate because of Download cable gets loose while sending data.



Picture 4.23

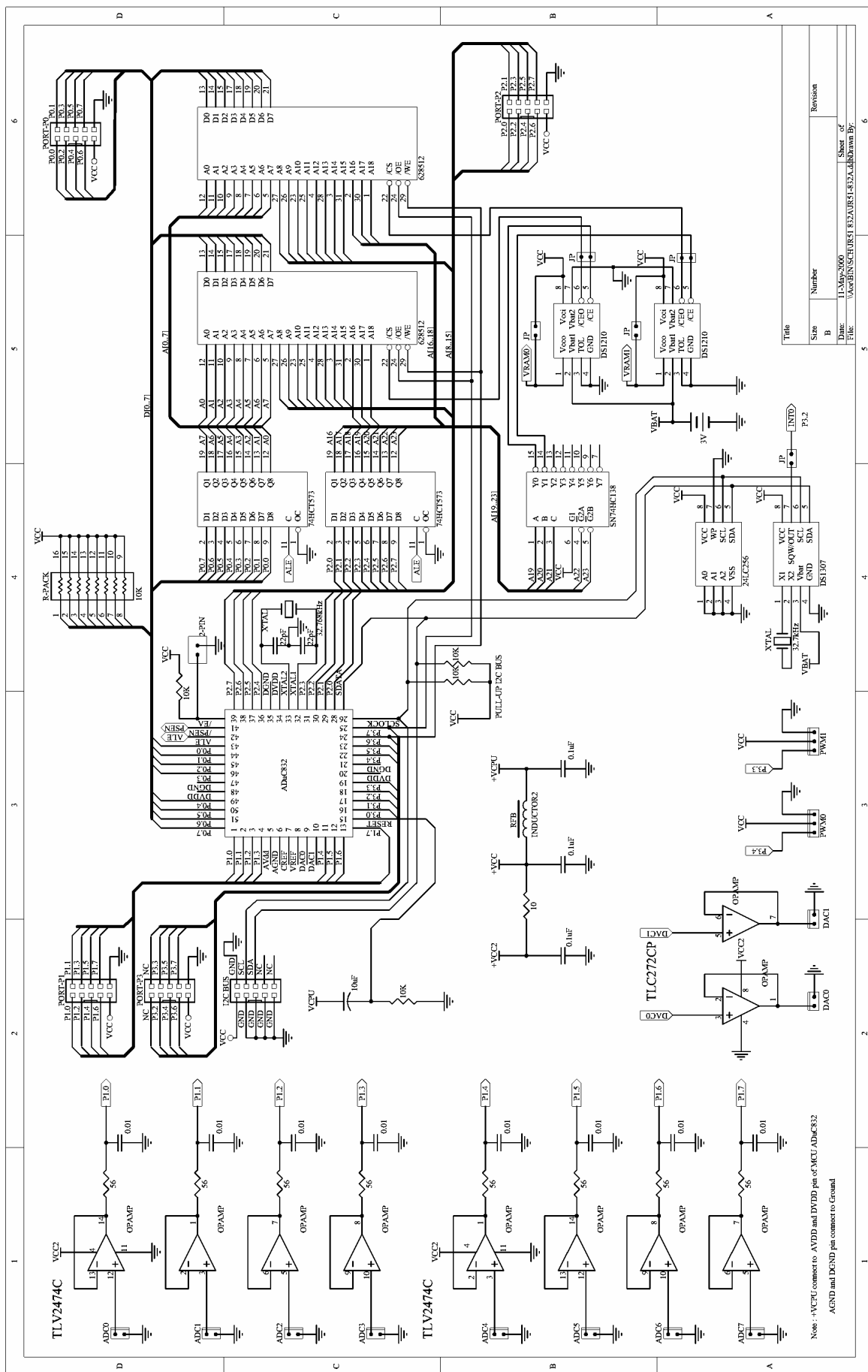
The Sample of Development Program

In this step, we hope most user can install program and use programs well. Now, we will mention about interfacing download cable and using programs with demonstrative board to develop project work. First, interface RS232 Download cable between demonstrative board and computer PC. Then, power will be supplied into demonstrative board, we can see red LED is in status ON. It means that it is ready to develop program.



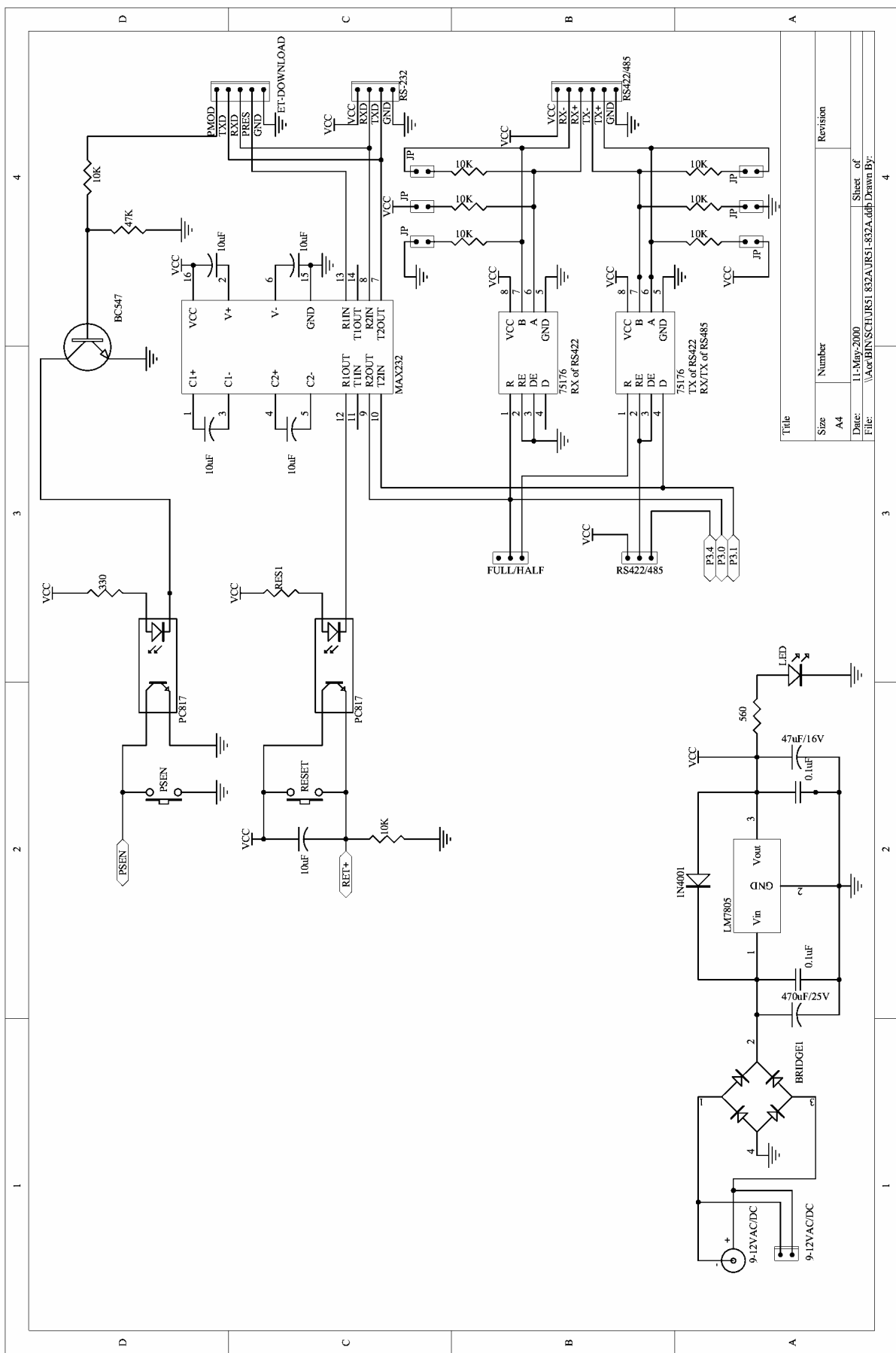
Picture 4.24

Write program as same as the sample in the picture 5.15 and save as .ASM file. Open program Cross 32 for assembling into .HEX file (can read from "**Using program Cross V4.0**"). Open program WSD that is Program Serial Download for downloading program .HEX file into demonstrative board, adjust Configurations suitably, and access into Download mode (can read from "Using In-Circuit Serial Download Mode") by using Download Button and PSEN button on board. Next, click RESET on program WSD and then click Download button and click Run button sequence (can read from "Using Program WSD"). In this step, Program is downloaded into demonstrative board CP-JR51-ADU832 completely and can run instantly.



Title	
Size	Number
B	7
Date	11-May-2000
File	Word\IN\SCHUR51\B2AURS1-832A.dbl
Drawn By	
Sheet of	6
Revision	

Note: +V_{CPU} connect to AVDD and DVDD pin of MCU ADuC832
AGND and DGND pin connect to Ground



Title		Revision	
Size	A4	Number	
Date:	11-May-2000	Sheet of	4
File:	\\Acrc\BIN\SCH\JRS1-832A\JRS1-832A.dtb	Drawn By:	