NEC

User's Manual

# F_Line – Drive it!

## Demonstration Kit for the F_Line Family

Document No. F_LINEDRIVEITV101
Date Published January 2005

- The information in this document is current as of 18.01.2005. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC sales representative for availability and additional information.

- No part of this document may be copied or reproduced in any form or by any means without prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.

- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such NEC Electronics products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.

- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of customer's equipment shall be done under the full responsibility of customer. NEC Electronics no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.

- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.

- NEC Electronics products are classified into the following three quality grades: “Standard”, “Special” and “Specific”.

The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated “quality assurance program” for a specific application. The recommended applications of NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.

"Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots.

"Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support).

"Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

The quality grade of NEC Electronics products is “Standard” unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact NEC Electronics sales representative in advance to determine NEC Electronics 's willingness to support a given application.

Notes: 1." NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.
2. " NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

M8E 02.10

*F_Line - Drive it!* complies with the EMC protection requirements

**CAUTION**

This equipment should be handled like a CMOS semiconductor device. The user must take all precautions to avoid build-up of static electricity while working with this equipment. All test and measurement tool including the workbench must be grounded. The user/operator must be grounded using the wrist strap. The connectors and/or device pins should not be touched with bare hands.

EEDT-ST-004-10

## Revision History

| Date | Revision | Chapter | Description |
|---|---|---|---|
| 30.01.04 | V1.00 | --- | First release |
| 18.01.05 | V1.01 | --- | Update |
| | | | |
| | | | |

## Table of Contents

## List of Figures



## List of Tables

# 1. Introduction

*F_Line – Drive it!* is a battery powered demonstration kit for the NEC's microcontroller family. It supports download and real time execution of small application programs up to 4 KBytes of program code for the V850 CPU module. The board is prepared to be connected to user hardware parts such as digital I/O or analogue signals.

## 1.1 Main features of *F_Line – Drive it!*

- Easy to use device demonstration capabilities
  *F_Line - Drive it!* contains elements to easily demonstrate simple I/O-functions, i.e. push buttons, LED output, AD reference voltage, I/O lines, RS232, LIN and CAN serial interfaces.

- Battery powered
  *F_Line - Drive it!* is battery powered for portability reasons.

- Serial communication interface program
  A Windows based download GUI allows to select and download application programs to *F_Line - Drive it!* for evaluation purposes.

- Analogue to digital signal conversion is supported

- Various input / output signals available, such as

  - All I/O ports
  - Timer input / output signals
  - Two or three wire serial I/O
  - RS232 signals including RTS / CTS
  - LIN bus support with transceiver TJA 1020
  - High Speed CAN bus interface with transceiver TJA 1040
  - 8 analogue input lines
  - 9 I/O ports connected to LED
  - 2 push buttons prepared for external interrupt generation

- Embedded Workbench for V850 and C-Spy Simulator / Debugger are included. These packages are restricted in such that maximum program code size is limited to 4 KB of program code.

- Full documentation is included for the NEC V850ES/FE2, NEC V850ES/FJ2, IAR Systems Embedded Workbenches, IAR Systems C-Spy and NEC FPL FLASH programmer software.

***F_Line - Drive it!* is not intended for code development. NEC does not allow and does not support in any way any attempt to use *F_Line - Drive it!* in a commercial or technical product.**

### 1.2 System requirements

**HOST PC**

A PC supporting Windows 95, Windows 98, Windows NT, Windows 2000 or Windows XP is required for the IAR Systems Workbench.

Pentium 166 MHz (at least), 64 MB of RAM, 256-color display (1024 * 768), mouse, CD-ROM drive and 40 Mbytes of free hard disk space are required to install the tool packages.

Above listed requirements are valid if the IAR Systems Embedded Workbench shall be installed. The communication interface program does not need to be installed to the local hard disk. It can be started from the CDROM.

**Host interface**

Serial (RS232C) interface capable to handle communication at 9600 baud.

### 1.3 Package contents

Please verify that you have received all parts listed in the package contents list, document "README.pdf" in the documentation directory of the CD of your *F_Line - Drive it!* package. If any part is missing or seems to be damaged, please contact the dealer from whom you received your *F_Line - Drive it!*.

**Note:** Updates to this User Manual, additional documentation and/or utilities for *F_Line - Drive it!*, if available, may be downloaded from the NEC WEB page(s) at http://www.ee.nec.de/updates.

## 2. *F_Line - Drive it!* system configuration

The *F_Line - Drive it!* system configuration is given in the diagram below:

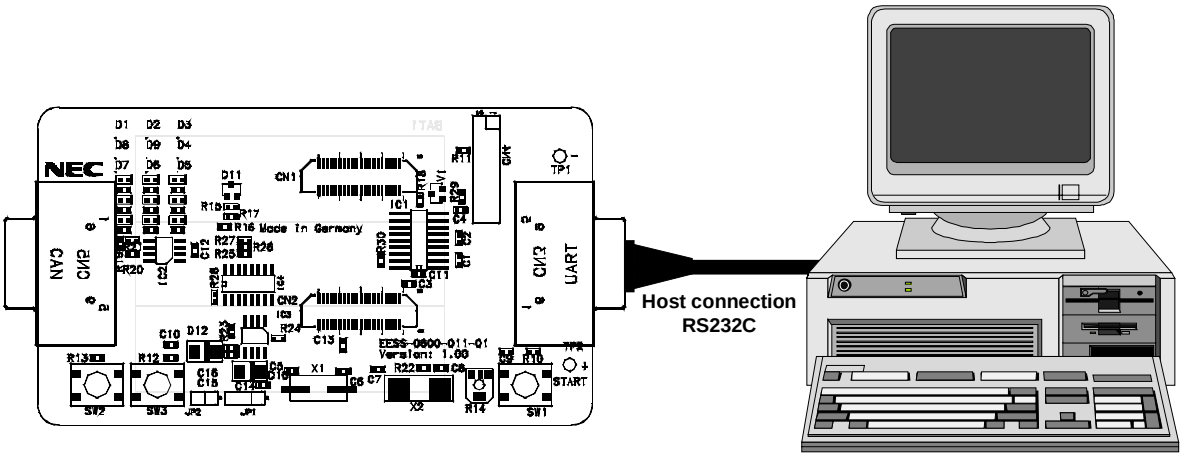



**Figure 1: *F_Line - Drive it!* system configuration**

### 2.1 *F_Line - Drive it!*

*F_Line - Drive it!* is a demonstration kit for the NEC F_Line family devices. Two CPU modules are included that represent the smallest and largest device of the F_Line with V850ES core.
The demonstration kit is equipped with a serial interface and a small monitor program to accept download program files for program execution. For optimized support a CAN and LIN interface is equipped on the board.
The *F_Line - Drive it!* board is connected to the host system via RS232C serial interface cable. The host system may be used to download and start application programs on *F_Line - Drive it!* platform.
*F_Line - Drive it! u*ses for both CPU modules a 5.0 MHz crystal so that programs can run on the internal PLL frequency of up to 20MHz .Sub-clock is provided with 32.768 kHz.

### 2.2 Host computer

The RS232 host interface enables communication to the *F_Line - Drive it!*. RS232 data transfer speed must be set to 9600 bps.

For a detailed specification of the host interface please refer to the chapter “Connectors and Cables” of this document.

### 2.3 Battery power supply

The *F_Line - Drive it!* is equipped with three batteries of AA type.

Before using *F_Line - Drive it!* please remove the isolation strip inserted between the battery and the battery holder.

Please note that a battery voltage below 3V may cause malfunction of the serial interface driver.

**Note:** The built-in monitor program switches *F_Line - Drive it!* to a power saving STOP mode in case serial communication is not established within ten minutes after power on. Press the START button to switch *F_Line - Drive it!* back to operation mode.

## 3. *F_Line - Drive it!* Baseboard components

The *F_Line - Drive it!* baseboard is equipped with push buttons, LEDs and with several connectors in order to be connected to host computers, Flash-programmer, LIN and High Speed CAN busses.

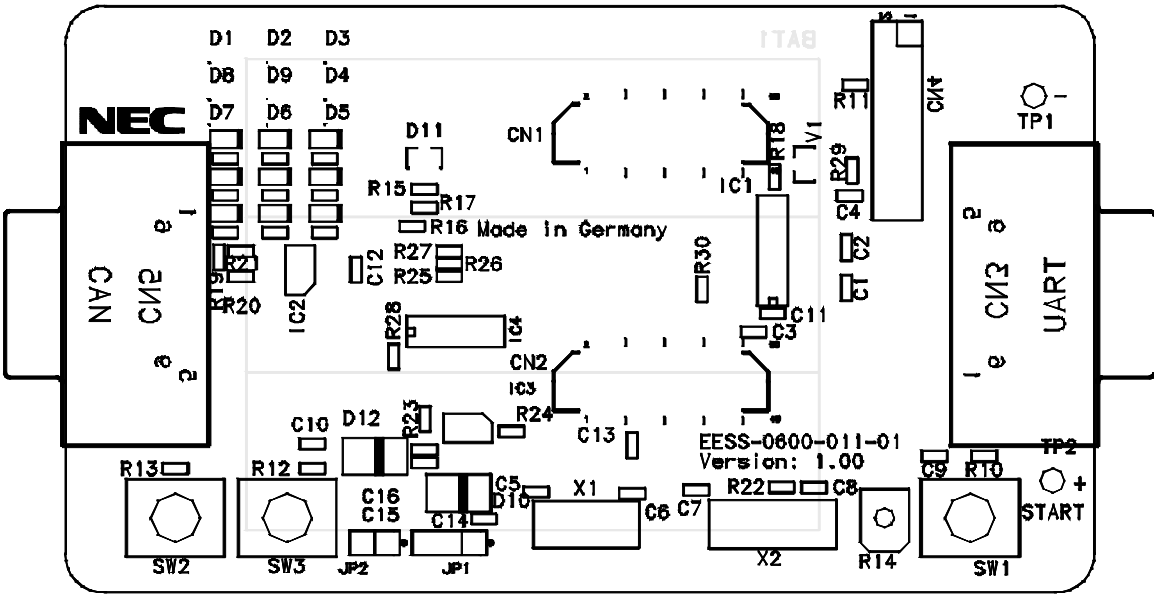



**Figure 2: *F_Line - Drive it!* baseboard jumpers, connectors, switches and LEDs**

Some of the F_Line - Drive it! components are free for user application hardware and software. Please read the user's manual of the V850ES/Fx2-device carefully to get information about the electrical specification of the available I/O ports before you connect any external signal to the F_Line - Drive it! board!

### 3.1 Start button SW1

SW1 is a reset button. It activates the power on reset. It is connected to the reset input of the CPU module.

### 3.2 User button SW2

SW2 is a push button connecting VCC to external interrupt input INTP1 of port P04 of the CPU module. The port may be programmed to generate interrupt INTP1. The necessary initialisation for this purpose is described in the user's manual of the V850ES/Fx2-device. The port is connected to a 1.2K pull down resistor.

### 3.3 User button SW3

SW3 is a push button connecting VCC to external interrupt input INTP3 of the CPU module. This is port P06 of the V850ES CPU module. The port may be programmed to generate interrupt INTP3. The necessary initialisation for this purpose is described in the user's manual of the V850ES/Fx2-device. The port is connected to a 1.2K pull down resistor.

### 3.4 LIN Plug JP1

JP1 is a 3 pin connector for the LIN bus, connected to the transceiver TJA1020.

| JP1 | PIN | Description |
|---|---|---|
| 1 | BAT | Reference voltage for the LIN bus level |
| 2 | LIN | LIN bus line |
| 3 | GND | Ground |

The LIN transmitter can be switched off by output a low level to the port P00

### 3.5 Jumper JP2

JP2 is a mode selector that enables to program application programs via FPL FLASH programmer GUI into the CPU internal flash memory of the connected CPU module. It will take effect only after RESET release provoked e.g. by pressing SW1.

Closing the jumper JP2 will enable the programming interface and connect the RS232 serial interface at the 9pin female Sub-D connector CN3 via a multiplexer to the UARTA0 of the V850/Fx2 CPU module. A user program is not able to run with closed jumper JP2 and the LIN bus should not be connected in this mode, because the LIN and the FPL FLASH programmer use same UARTA0, so that they effect each other.

The FPL FLASH programmer software a separate product from NEC and is included in this package.

### 3.6 RS232 serial interface connector CN3

RS232 interface lines on the 9pin female Sub-D connector CN3 are connected via a driver MAX3222 and a multiplexer to the V850ES/Fx2-device. For the monitor program the control lines RTS and CTS are connected as well.

| CN3 | JP2 = open (monitor) | JP2 = closed (FPL FLASH programmer) | RS232 |
|---|---|---|---|
| 1 | -- | -- | NC |
| 2 | RXDA1 | RXDA0 | RxD |
| 3 | TXDA1 | TXDA0 | TxD |
| 4 | -- | -- | NC |
| 5 | -- | -- | Ground |
| 6 | -- | -- | NC |
| 7 | P97 | P97 | CTS |
| 8 | P96 | Logic 1 | RTS |
| 9 | -- | -- | NC |

**Table 1: CN3 connection to RS232**

For program download serial communication speed is fixed to 9600 baud. Data format is 8 data bits, 1 stop bit, no parity.
If the serial interface is used from a downloaded application program, any other communication speed or data format may be used according to the V850ES/Fx2 capabilities. Please refer to the user's manuals for details.

**Note:**

P99 output signal may be used to shut down the MAX3222 serial line driver. It is recommended to shut down the MAX3222 driver if the serial interface is not used. This will extend the battery life time.
A low signal output at P99 will shut down the MAX3222. A high signal will wake up MAX3222.
P98 at V850ES/Fx2 output signal must be set to low signal to enable MAX3222.

### 3.7 High Speed CAN connector CN5

CN5 is a D-SUB 9 connector for High Speed CAN with CiA standard pin assignment. The used transceiver is the TJA1040 with bus termination. The transceiver is connected to the CAN0 of the V850/Fx2, whereby the standby mode control is selected by port P35.

### 3.8 Connector CN1 and CN2

CN1 and CN2 are connectors for the CPU module.

### 3.9 Connector CN4

CN4 connector allows connecting the PG-FP4 flash programmer to F_Line - Drive *it!* in order to program application programs into the CPU internal flash memory of the connected CPU module. Programming via PG-FP4 flash programmer is more powerful than programming via the FPL FLASH programming GUI. Reprogramming the CPU modules will irretrievably overwrite the monitor program.
PG-FP4 is a separate product from NEC and it is not included in this package.

### 3.10 External Potentiometer R14

A 10K potentiometer R1 is connected between P915 of V850ES/Fx2 and ground. The potentiometer arm is connected to ANI2 analogue input.

### 3.11 AD converter reference voltage input

A 1.235V reference voltage is connected to ANI0 input. The reference voltage must be enabled by port P913 of V850ES/Fx2.

### 3.12 Supply voltage monitor

For reference purposes $V_{CC}$ / 2 is connected to ANI3 input. The supply voltage monitor must be enabled by port P914 of V850ES/Fx2.

### 3.13 External LED1 – LED9

LED9 to LED1 are LED connected to PDL of the V850/Fx2. LED9 is an LED connected to PCM0 of the V850/Fx2.

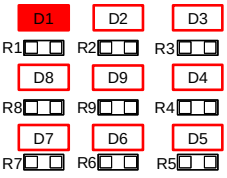
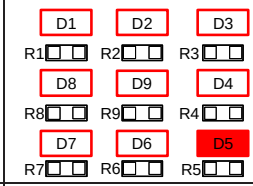
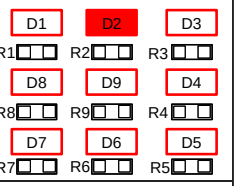
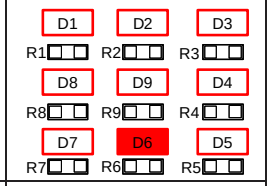
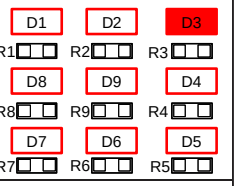
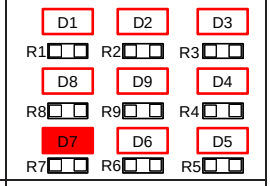
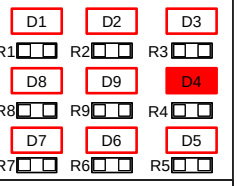
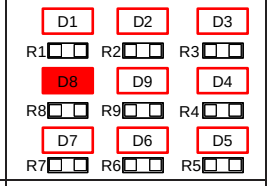
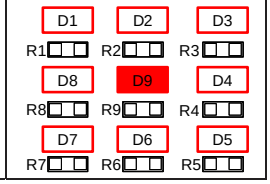
| Port | Segment | Display | Port | Segment | Display |
|---|---|---|---|---|---|
| PDL0 V850ES/Fx2 | LED1 |  | PDL4 V850ES/Fx2 | LED5 |  |
| PDL1 V850ES/Fx2 | LED2 |  | PDL5 V850ES/Fx2 | LED6 |  |
| PDL2 V850ES/Fx2 | LED3 |  | PDL6 V850ES/Fx2 | LED7 |  |
| PDL3 V850ES/Fx2 | LED4 |  | PDL7 V850ES/Fx2 | LED8 |  |
| | | | PCM0 V850ES/Fx2 | LED9 |  |

**Table 2: LED1 – LED9 connection**

A low signal output at each port switches the corresponding LED on.

## *4. F_Line - Drive it! CPU module components*

Both F_Line - Drive it! CPU modules are equipped with 4 connectors in order to be connected to user defined hardware. One module is the V850ES/FE2 CPU in a 64 pin package and the other is the V850ES/FJ2 CPU in a 144 pin package.

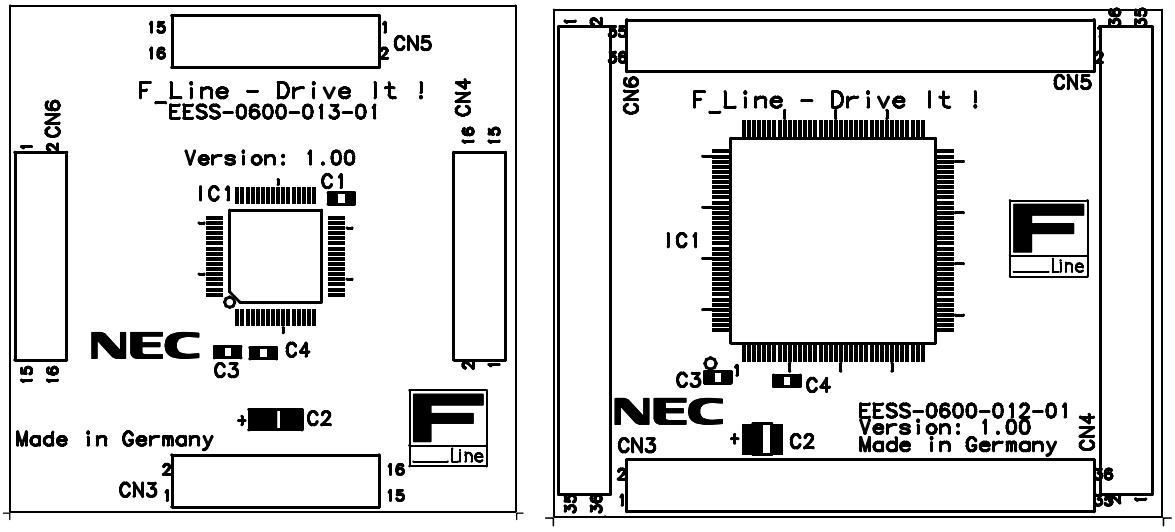



**Figure 3: *F_Line - Drive it!* CPU module components**

### 4.1 External connector CN3,CN4, CN5, and CN6

CN3, CN4, CN5, and CN6 are connectors for external user hardware. The pinning equals the pinning of the related device. Therefore each of the connectors CN3 to CN5 can be assembled with a 16 pin plug with a 2.54 mm pitch for the V850ES/FE2 and a 36 pin plug for the V850/FJ2 respectively. Please read the user's manual of the V850ES/Fx2-device carefully to get information about the electrical specification of the available I/O ports and check the connected signal on the base board before you connect any external signal to the *F_Line - Drive it!* board!

### 4.2 N-Wire On-Chip Debug Unit

The V850ES/FE2 and V850ES/FJ2 include an on-chip debug unit. By connecting an N-Wire emulator, on-chip debugging can be executed with the V850ES/Fx2. The interface signal pins that must be connected are located on the CN4 and CN5 connector of the V850ES/FE2 and on CN4 and CN5 of V850ES/FJ2 respectively. The interface signal pins are

| Name | I/O | Function | FE2 pin | FJ2 pin |
|---|---|---|---|---|
| DCK | in | Clock input | 34 | 41 |
| DMS | in | Mode select input | 35 | 42 |
| DDI | in | Data input | 30 | 39 |
| DDO | out | Data output | 31 | 40 |
| DRST | in | On-chip debug unit reset input | 17 | 20 |
| FLMD0 | in | Flash Programming Mode Setting | 3 | 8 |
| RESET | in | RESET | 9 | 14 |

**Table 3: Signal pins for On-Chip Debug Unit**

(in and out indicates the direction when seen from the target device).

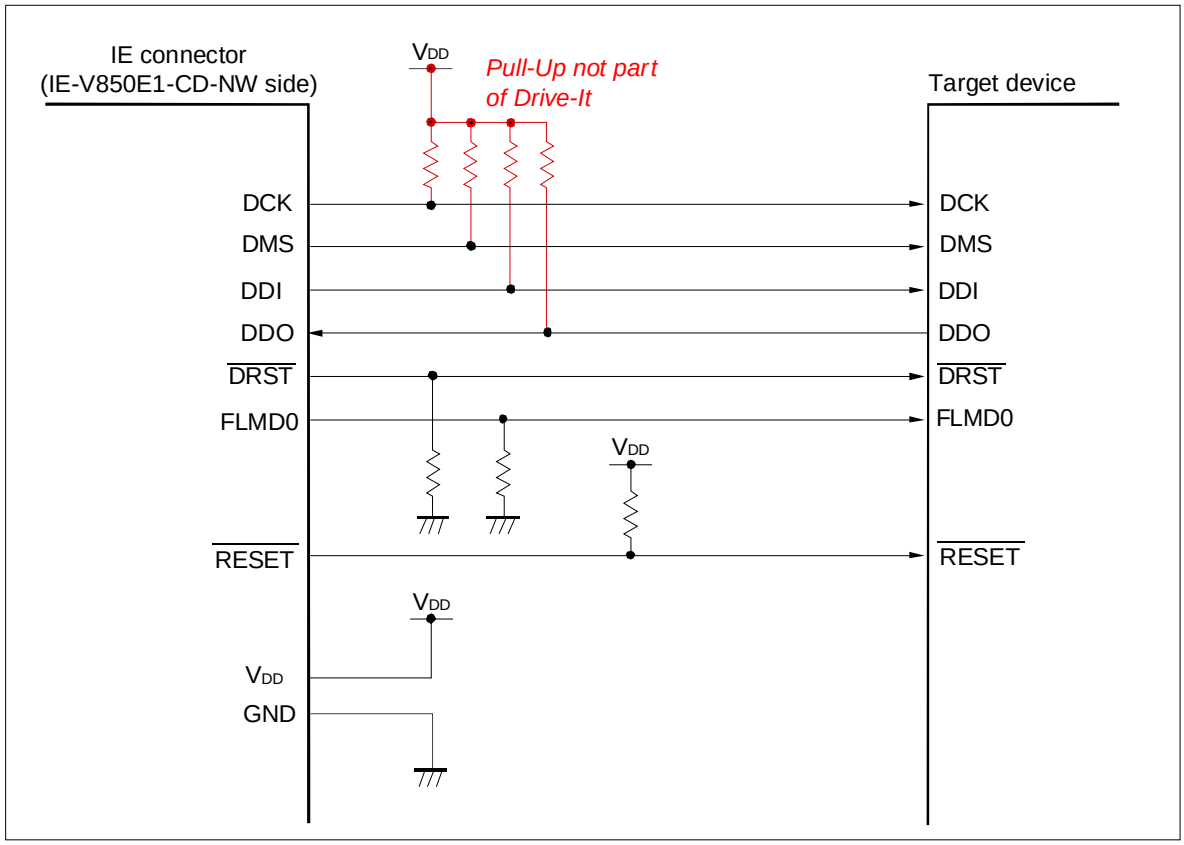



**Figure 4: *N-Wire Connection Circuit Example***

For debugging please connect this signal pins to the connector of the N-Wire emulator. Refer to the user's manual of the emulator or to the user's manual of the V850ES/Fx2-device for more details. The CPU can be started in the debug mode immediately after reset of the CPU is released.

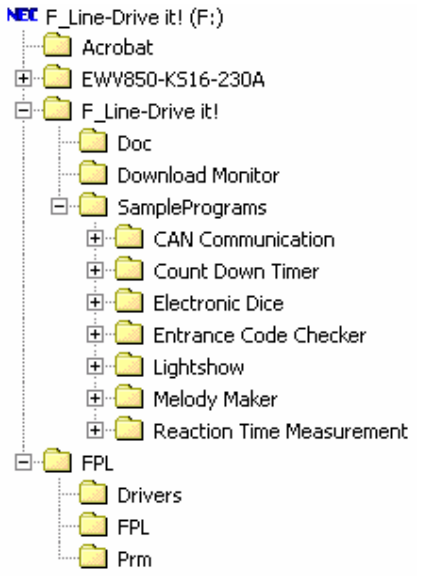
# 5. *F_Line - Drive it!* installation and operation

## 5.1 Getting started

*F_Line - Drive it!* is equipped with a simple download monitor that allows communication with a PC host system via RS232 serial interface line. Before you can run and download a program, hardware and software must be installed properly.

### 5.1.1 CD-ROM contents

The CD-ROM shows following directory structure:




| CD-ROM ROOT |
|---|
| Acrobat Reader |
| IAR Embedded Workbench V850 |
| F_Line - Drive it! |
| - Documentation |
| - Download Monitor |
| - Sample programs for … |
| … CAN communication |
| … count down timer |
| … electronic dice |
| … entrance code checker |
| … lightshow |
| … melody maker |
| … reaction time measurement |
| FPL FLASH programmer software |
| - Drivers for USB |
| - FPL setup directory |
| - PRM Parameter files |

**Table 4: F_Line - Drive it! CD-ROM directory structure**

### 5.1.2 Hardware installation

After unpacking *F_Line - Drive it!* please remove the battery isolation strip inserted between the battery and the power connection on the back side of *F_Line - Drive it!.*
After pressing the START button LED1 to LED8 will flash one after the other. If you press one of the buttons SW2 or SW3 the flashing speed will increase.
LED flashing indicates that *F_Line - Drive it!* is ready to be connected to a host computer.

Connect *F_Line - Drive it!* to your host computer using the provided serial interface cable. The communication speed is fixed to 9600 bps.

### 5.1.3 Software installation

The *F_Line - Drive it!* package comes with the several software demo packages:

- IAR Systems Embedded Workbench for V850, including C compiler, assembler, linker, librarian and C-SPY Simulator.
- Communication interface for program download
- Sample programs
- FPL FLASH programmer software for flash programming of the NEC V850ES/F_Line

The IAR Systems Embedded Workbench and FPL FLASH programmer must be installed on your PC. For detailed installation hints, refer to the documentation of the corresponding products.

The communication interface can be started from the CDROM without installation. Also sample programs can be downloaded and executed directly form the CDROM. Only if you intend to modify some of the sample programs it is necessary to copy them to your local hard disk.

#### 5.1.3.1 IAR Systems Embedded Workbench for V850 installation

To install the IAR Systems Embedded Workbench for V850, select the SETUP program in the directory \EWV850-KS16-230A\EWV850\ of the CDROM.
The setup dialogues will guide you through the installation process.

#### 5.1.3.2 Communication interface installation

The communication interface program is available in the \F_Line-Drive it!\ directory of the CDROM. It does not require any installation. It can be started directly from the CDROM or it can be copied into any directory of your local hard disk. Please make sure that you copy both files, Driveit_GUI.exe and neccom.dll.

#### 5.1.3.3 Sample program installation

The sample programs do not require any installation for download.
If the sample programs shall be modified it is required to copy them into any directory of your local hard disk. A file copy using the Windows explorer is the recommended procedure.

#### 5.1.3.4 FPL FLASH programmer software installation

The FPL FLASH programmer software must be installed. You may start the installation program from the auto-start program or by running the SETUP.EXE program available in the \FPL directory of the CDROM.

### 5.2 Download monitor resources

The ROM monitor program is contained in the internal flash area of the micro controller of the *F_Line - Drive it!*. It uses some resources for its own purposes.
As soon as an application program starts execution, resources are returned to the system and all of them are available for user application programs.
Despite of that, some features of the V850ES/F_Line micro controllers are not available to the user.

Resources not available to the user:

- The user application program cannot use the RESET vector.
- The internal expansion RAM cannot be used for data storage because the user application program will be downloaded into this memory area.

The user application program can use all interrupt functions (except RESET). The download monitor will redirect any interrupt to a shadow interrupt vector table. Depending on the used CPU module this shadow interrupt vector is starting according to the following table. The user must make sure that the interrupt vector table has been located from that address onwards for *F_Line - Drive it!*. For detailed explanation, please refer to the sample program descriptions.

| CPU module | Start address |
|---|---|
| V850ES/F_Line | 0x03FFD800 |

**Table 5: Shadow interrupt vector table**

## 6. *F_Line - Drive it!* communication interface

Driveit_GUI.exe is the communication interface for program download. It downloads user application programs to *F_Line - Drive it!* and starts them automatically.
Driveit_GUI.exe can be executed in Windows 9x, Windows NT, Windows 2000, Windows ME and Windows XP environment.

When Driveit_GUI starts, following screen will appear:


**Figure 5: Driveit_GUI startup screen**

Several actions may be initiated by menu selections from this communication interface.

### 6.1 Settings menu

The Settings menu allows selecting the communication port and establishing communication to the F_Line - *Drive it!* hardware.

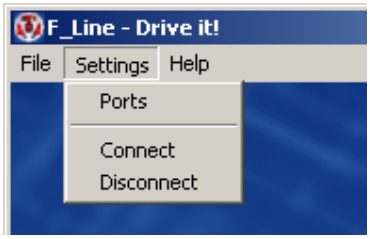



**Figure 6: Settings menu**

### 6.1.1 Ports

When activating the Ports menu, a communication setting dialogue opens.

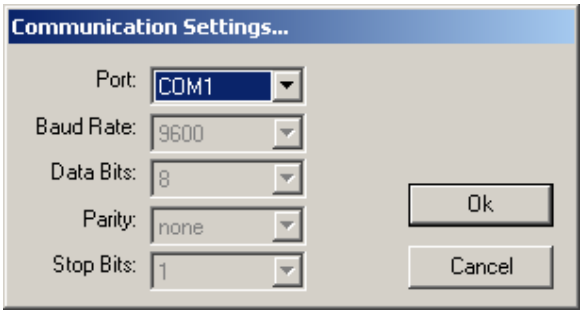



**Figure 7: Communication Settings dialogue**

In this dialogue, only the communication port can be selected. Other parameters are fixed.

### 6.1.2 Connect

When the Connect menu item is activated, communication to the *F_Line - Drive it!* will be established. Make sure that the *F_Line - Drive it!* is connected to the correct port of your PC before you activate this menu. On successful communication link, the main window will display a message depending on the used CPU module in its headline:

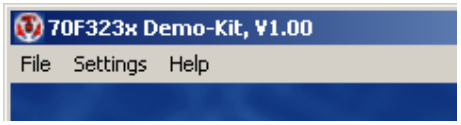



**Figure 8: Well-established communication message for F_Line CPU module**

On the *F_Line - Drive it!* the LED2, LED4, LED6 and LED8 will be switched on to indicate the connection status. In case an error occurs, an error message will be displayed. For details about the error messages please have a look at chapter 6.4.

## 6.2 File menu

The File menu allows selecting program files for download.

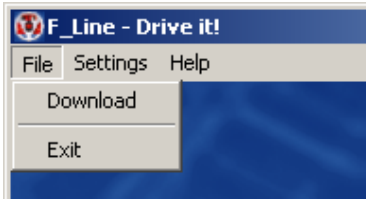



**Figure 9: File menu**

### 6.2.1 Download

When the Download menu is activated, a file open dialogue allows to select a HEX file for downloading to the *F_Line - Drive it!*

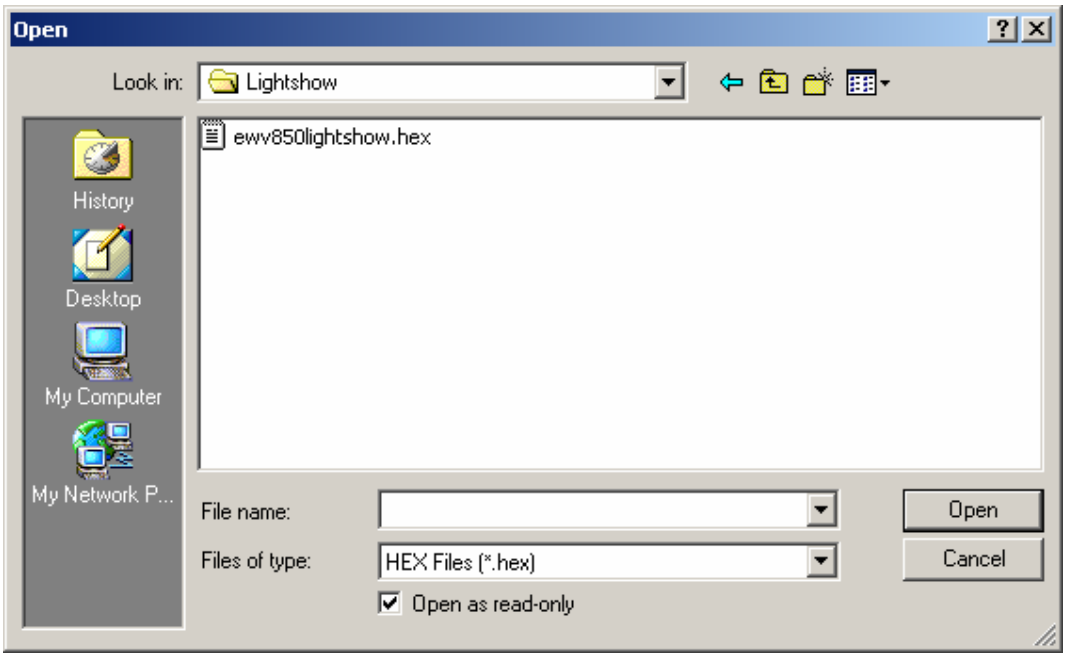



**Figure 10: Select download file dialogue**

Select the download file and activate OK. The selected file will be downloaded to *F_Line - Drive it!* . The *F_Line - Drive it!* will indicate the download activity by some flashing LED.
When the download is completed, a message box will indicate that the application program will start automatically. Please note that after program download the *F_Line - Drive it!* is disconnected automatically. Since the downloaded application program may use UART and other resources, communication interface program cannot communicate to *F_Line - Drive it!* any more. Therefore you have to d re-connect before downloading a new application. On the *F_Line - Drive it!* you have to press the reset-button SW1 to leave the application before re-connecting.

### 6.2.2 Exit

The Exit menu will leave the communication interface program.

## 6.3 Help menu

The Help menu allows selecting an About box. On-line help is not yet provided with this communication interface program.


**Figure 11: Help menu**

### 6.3.1 About menu

The About box shows the version number of the Driveit_GUI.exe communication interface program.

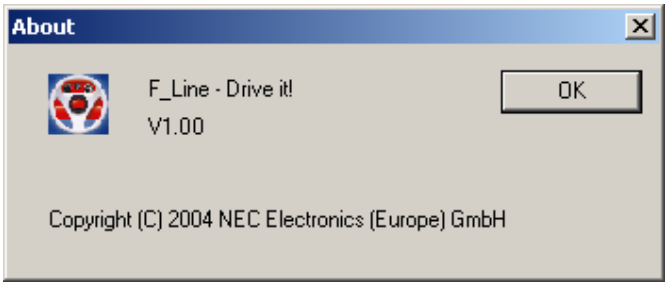



**Figure 12: About box**

## 6.4 Questions and Messages

During operation the following message boxes may occur informing the user or forcing a confirmation.

### 6.4.1 Questions to confirm a selected action

The user must confirm the following actions:

| Selected action | Question | Possibilies |
|---|---|---|
| Exit while connected | Do you really want to disconnect and exit? | Yes / No |
| Exit while disconnected | Do you really want to exit? | Yes / No |

**Table 6: User confirmation messages**

### 6.4.2 Informational Messages

The following messages inform the user about the current action or a currently restriction of the selected action:

| Messages |
|---|
| Program downloading and executing... |
| Parameters cannot be changed while being connected to the target board!! |
| The target board has been disconnected. Press RESET button before re-connecting! |

**Table 7: User confirmation messages**

### 6.4.3 Fatal Error Messages

If one of the following fatal errors occur, an error message is displayed and after the confirmation the program aborts:

| Number | Message | Reason |
|---|---|---|
| E100 | Error while loading neccom.dll.<br>Application will be closed! | Tasteit_GUI could not find or load NECCOM.DLL. |
| E101 | No RS232 communication interface detected.<br>Application will be closed! | No valid RS232 communication port found on current PC. |

**Table 8: Fatal error messages**

### 6.4.4 RS232 Connection Error Messages

| Number | Message | Reason |
|---|---|---|
| E200 | The target board does not answer!<br>Press RESET button and try again! | There is no response coming from F_Line - Drive it! when sending a ***Connect*** command.<br>Reasons could be<br>• no board connected<br>• board switched off<br>• board in run or connection mode<br>• wrong COM connection |
| E201 | The target board returned an unknown or incomplete Board ID! | The information received on a ***Connect*** command cannot be evaluated. The data does not identify one of the allowed monitors. Please check board type, RS232 line and connection parameters. |

**Table 9: RS232 connection error messages**

### 6.4.5 Download and File Handling Error Messages

In case of a download- or file handling error, the download aborts, but the *F_Line - Drive it!* stays in valid connection status. It is possible to select another download file.

| Number | Message | Reason |
|---|---|---|
| E300 | Cannot open download file. | The selected download file cannot be opened for reading.<br>Error comes from OS. |
| E301 | Error reading download file. | The download file cannot be read. Either EOF has been reached or file access problems have occurred.<br>Error comes from OS. |
| E302 | Mixed extended format (S). | The program download file in intel extended hex format contains a mixture of segmented and linear file format and is invalid. |
| E303 | Mixed extended format (L). | The program download file in intel extended hex format contains a mixture of segmented and linear file format and is invalid. |
| E304 | Invalid record in download file. | The program download file contains an invalid record identifier and is invalid. |
| E305 | Incomplete line in download file. | The program download file contains less data than specified in the record header and is invalid. |
| E306 | Address out of range. | The program download file contains address information exceeding the available ROM space in the target device. |
| E307 | Invalid File Format | The program download file is not of intel hex file format. |

**Table 10: Download an file handling error messages**

### 6.4.6 NECCOM.DLL Error Messages

A problem with the NEC communication DLL is shown by the following messages. No workaround is available.

| Number | Message | Reason |
|---|---|---|
| E400 | Connection failed! | Opening the communication channel failed. |
| E401 | Disconnection failed! | Closing the communication channel failed. |
| E402 | Illegal parameter selection! | Illegal communication parameters selected. |

**Table 11: NECCOM.DLL error message**

# 7. Sample programs

## 7.1 General Introduction

Each of the sample programs is located in a single directory, which will be called main-directory of the sample. This main directory of each sample contains the complete project inclusive all output files of the development tools. All sample programs use the same directory structure:

| | |
|---|---|
| c-spy File Folder | output files for C-Spy simulator (only V850/Fx2) |
| inc File Folder | C header files |
| settings File Folder | configuration files Embedded Workbench V850 |
| source File Folder | C source files |
| V850Modul File Folder | output files for V850/Fx2 CPU module |
| xcl File Folder | xcl files |
| description.txt 1 KB Text Document | example description |
| ewv850timer.pew 8 KB IAR IDE Project | project file Embedded Workbench V850 V2.30a |

**Table 12: Example directory structure**

The main directory contains only the project files for the IAR Systems Embedded Workbench V850 and a short description file.
All source files are located in the directory **source** and the **inc** directory contains the header files. Each sample project uses two targets. One target is the simulator (IAR Systems C-Spy) and the other is the demonstration kit hardware (with V850ES/Fx2 CPU module). All targets use the same source files, but they differ in the memory mapping. The different memory mapping is defined in the linker control files (*.xcl) which are located in the directory **xcl**.
The linker control file df3231.xcl define the ROM segments in the address area of the FLASH memory of the real device. This mapping must be used for the simulator. To download the sample program to the demonstration kit all ROM segments are defined in the address area of the internal RAM of the real device. This mapping is defined in the linker control file V850Modul.xcl for the V850/Fx2 CPU module.

All output files of the development tool are generated in the directory **c-spy** and **V850Module.**

To open the IAR Systems Embedded Workbench for a sample program please double-click on the corresponding project file in the main directory of the sample program. Due to a change in the project file format only an Embedded Workbench V850 version v2.30a or later can open the project files for V850ES/Fx2.
If you use an older version, a warning message appears. If you want to use a former Embedded Workbench version you have to create a new project file on your own, using the above description of the files and their location. Figures 20 and 21 on the next page show screenshots the IAR Systems Embedded Workbench.

For details of using Embedded Workbench and C-Spy Simulator please refer to the manuals.

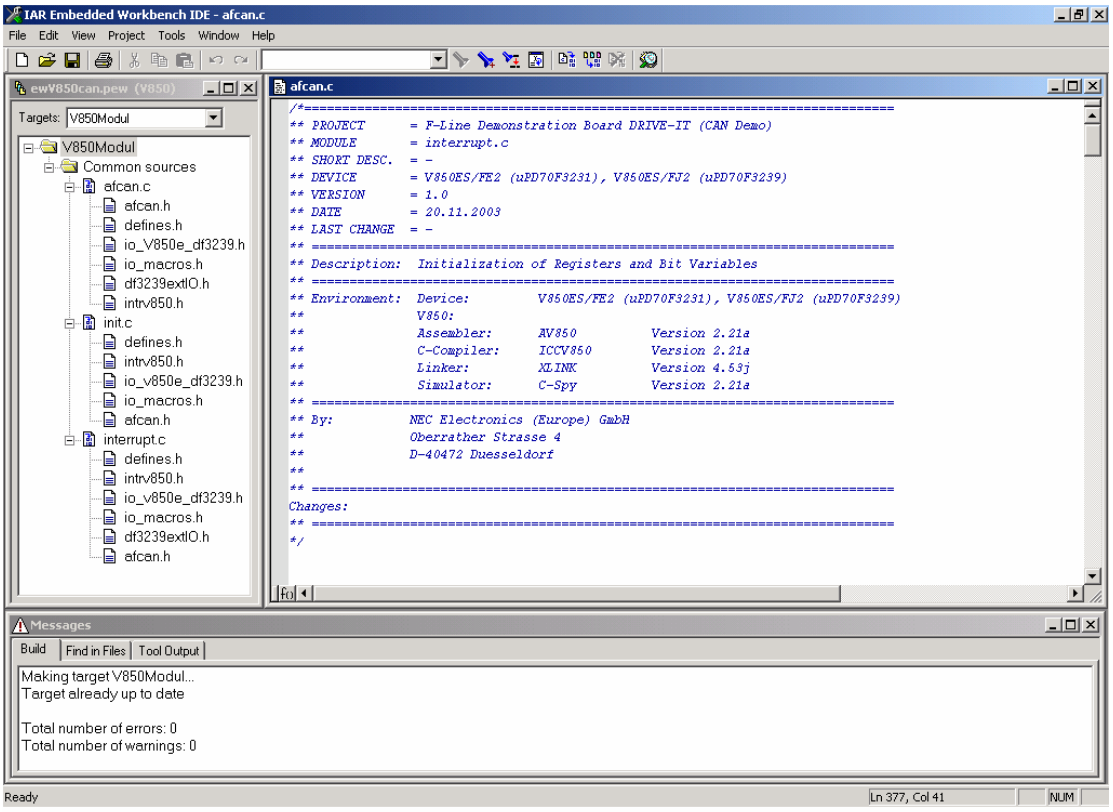



**Figure 13: IAR Systems Embedded Workbench V850**

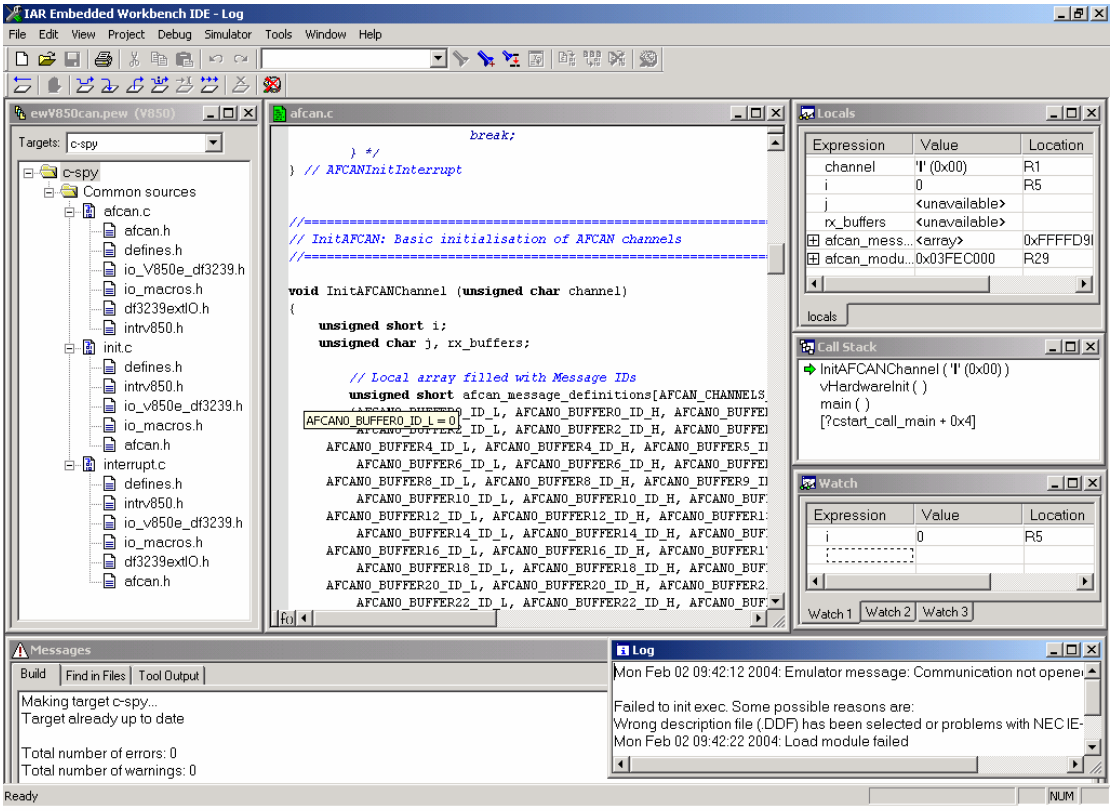



**Figure 14: IAR System V850 C-Spy Simulator**

### 7.2 Common description of all sample programs

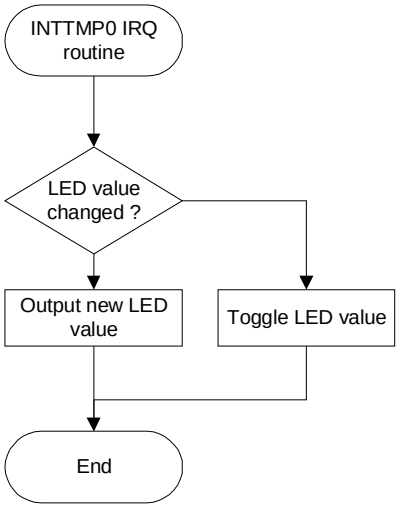
As the demonstration kit hardware is powered by a battery some power saving features are implemented in all sample programs:

- Usage of a pulsed signal (100Hz) to control the LEDs instead of a static signal. The pulsed LED control is implemented as an interrupt function of Timer P0 (TMP0).

- All hardware components (RS232 driver, reference voltage, potentiometer, etc) are disabled if they aren't used in an application.
  Therefore, if you write your own application be aware that the hardware you want to use is enabled.

To signal the program start after the download, all LEDs flash two times in a one-second interval.

Flowchart of the TMP0 interrupt function to generate the pulsed LED control:



### 7.3 Electronic dice

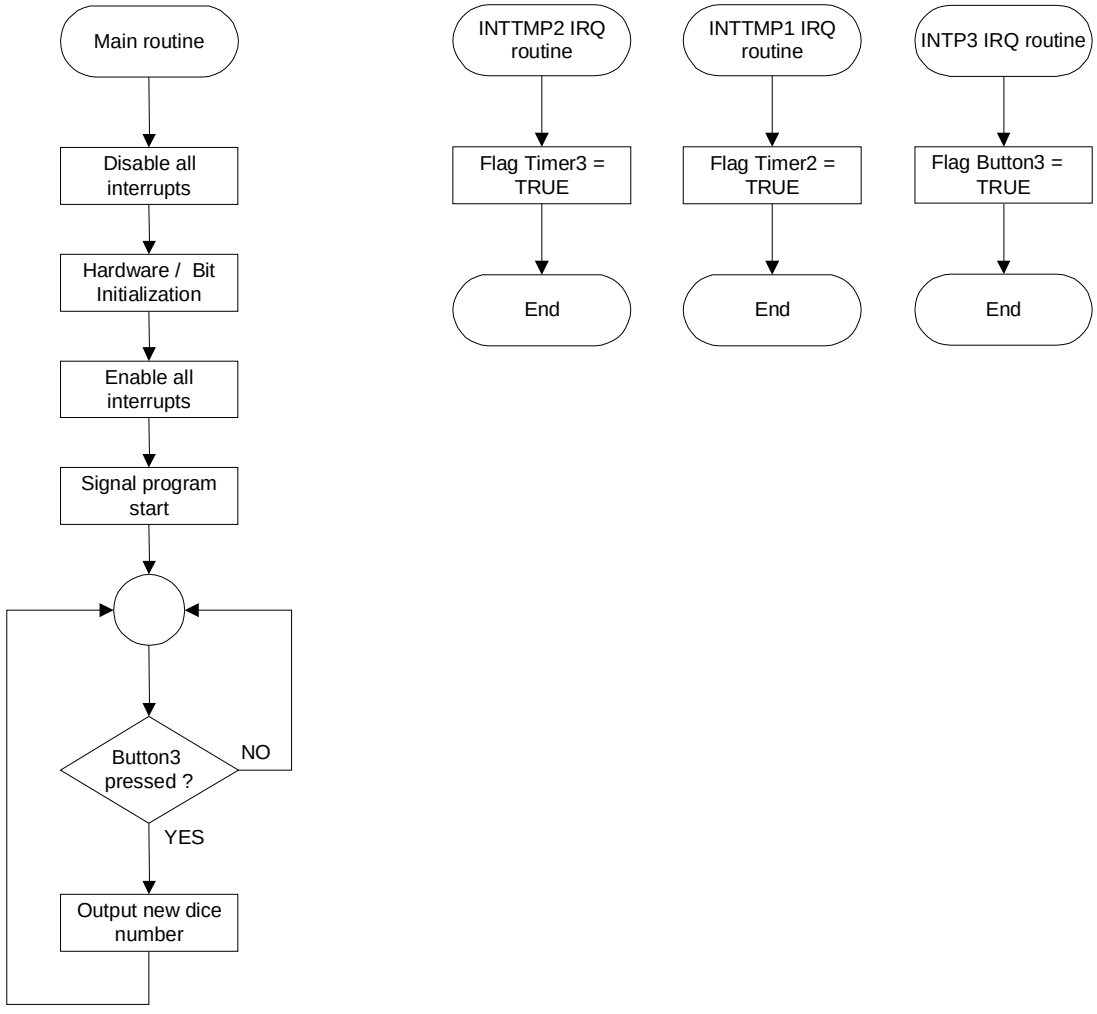
This sample program simulates a dice. After the program-start-signal, the program waits for a press of button 3 (SW3). After an animation of a rolling dice a random number between one and six is generated and shown at the LEDs.

The ‘random’ number is generated by a transformation of the timer value of Timer P2 to a number between one and six. Timer P2 is working at 2.5 MHz using a compare value of 80 in the modus ‘clear and start on match between TMP2 and TP2CCR0’. The result is an interrupt repetition time of 31,25 KHz. Timer P1 is working in the same modus to generate a 25ms timebase.

| Used Internal Peripherals | Used External Parts | Source Modules | |
|---|---|---|---|
| Timer P0 (TMP0) | LEDs | dice.c | main function |
| Timer P1 (TMP1) | Button3 (SW3) | init.c | Hardware and Bit initialization |
| Timer P2 (TMP2) | | interrupt.c | interrupt functions |
| | | | |

To end the program and return to the download program again please press the START button.

Flowcharts of the main routine and the interrupt functions:



### 7.4 Lightshow

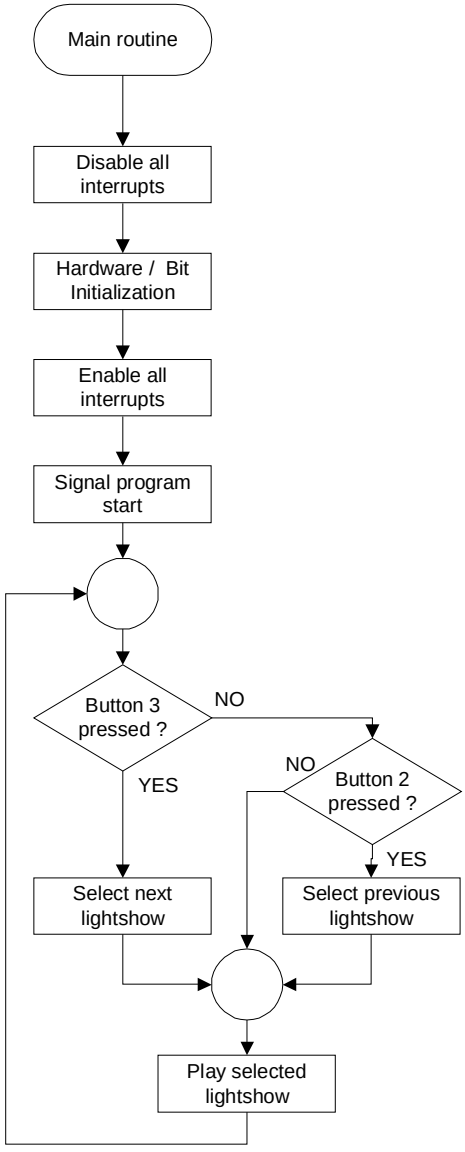
This sample programs plays one of three predefined lightshows. After the program-start-signal, the program plays the first lightshow. By pressing button3 (SW3) the next show is selected. Button2 (SW2) selects the previous show. Every show is completed before the next show starts.

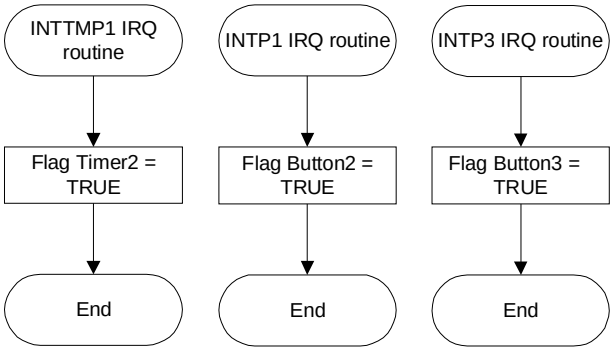
Timer P1 (TMP1) is working in the modus ‘clear and start on match between TMP1 and TP1CCR0’. According to selected operating frequency and the value of the compare register CCR0 Timer P2 generates a 25ms timebase.

| Used Internal Peripherals | Used External Parts | Source Modules | |
|---|---|---|---|
| Timer P0 (TMP0) | LEDs | lightshow.c | Main function |
| Timer P1 (TMP1) | Button2 | init.c | Hardware and bit initialization |
| | Button3 | interrupt.c | Interrupt functions |
| | | sequences.c | Lightshow sequence definitions |
| | | | |

To end the program and return to the download program again please press the START button.

Flowcharts of the main routine and the interrupt functions:



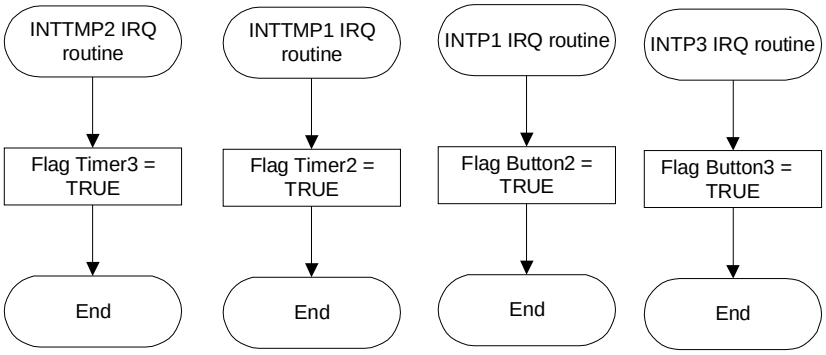
### 7.5 Reaction time measurement

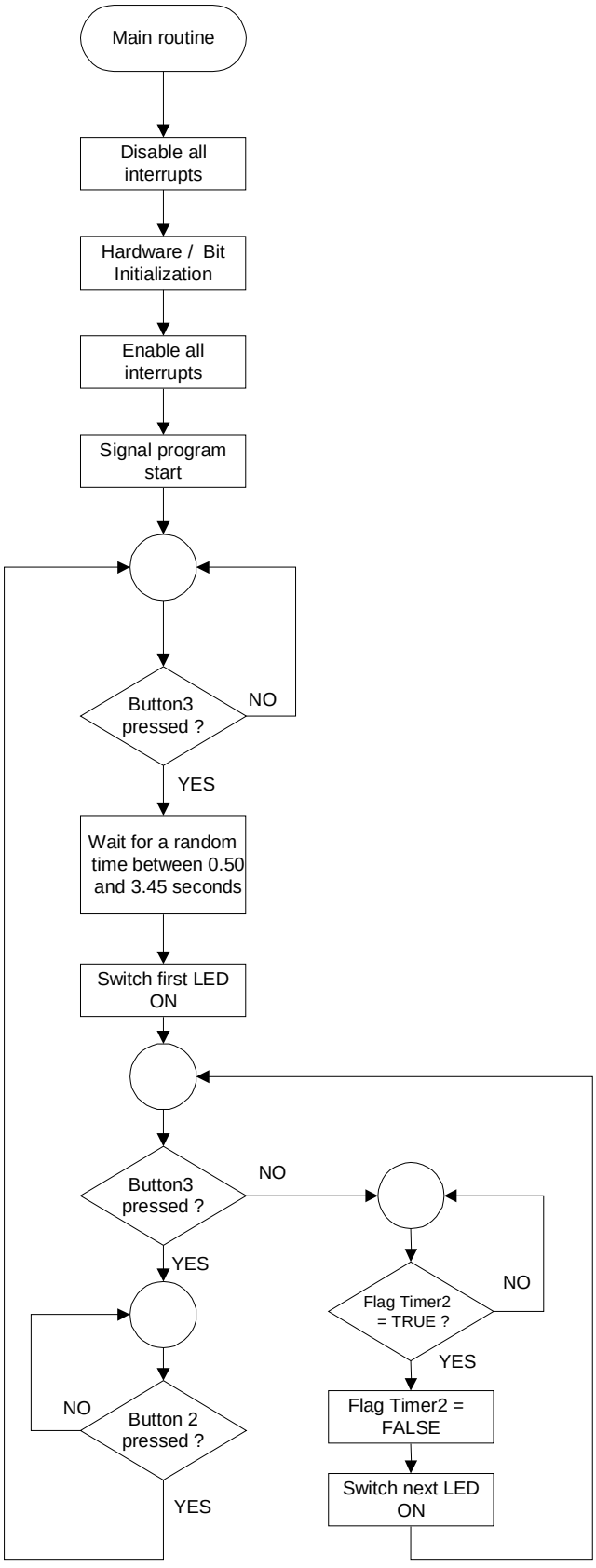
This sample program demonstrates a reaction time measurement. After a press of button3 (SW3) the application waits for a random time between 0.50 and 3.45 seconds. Then the first LED is switched on and measurement starts. Until the next keystrokes of button3 every 50 ms a new LED is switched on. Pressing button2 (SW2) starts a new measuring cycle.
Timer P3 is working at 2.5 MHz using a compare value of 80 in the modus 'clear and start on match between TMP2 and TP2CCR0''. The result is an interrupt repetition time of 31.25 kHz. Timer P2 is working in the same modus to generate a 25ms timebase.

| Used Internal Peripherals | Used External Parts | Source Modules | |
|---|---|---|---|
| Timer P0 (TMP0) | LEDs | reactime.c | Main function |
| Timer P1 (TMP1) | Button2 | init.c | Hardware and bit initialization |
| Timer P3 (TMP2) | Button3 | interrupt.c | Interrupt functions |
| | | | |

To end the program and return to the download program again please press the START button.

Flowcharts of the main routine and the interrupt functions:




Main routine
Disable all interrupts
Hardware / Bit Initialization
Enable all interrupts
Signal program start
Button3 pressed ?
NO
YES
Wait for a random time between 0.50 and 3.45 seconds
Switch first LED ON
Button3 pressed ?
NO
YES
Flag Timer2 = TRUE ?
NO
YES
Flag Timer2 = FALSE
Switch next LED ON
Button 2 pressed ?
NO
YES

### 7.6 Entrance code checker

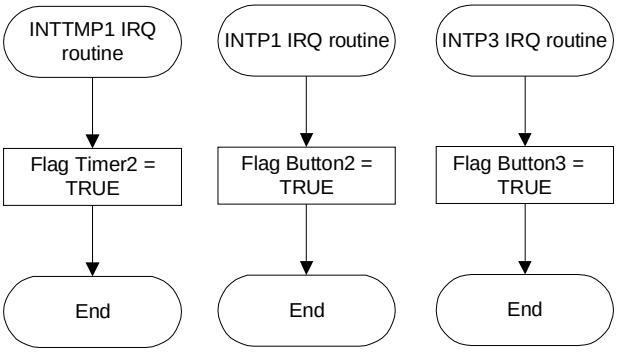
This sample program waits for a sequence of five keystrokes. If the input sequence matches the predefined sequence LED9 and one Port are switched on for 5 seconds. Otherwise, all LED flash five times.
Then the program waits for a new input sequence. The output port is P4.1.
The code sequence is stored in the array ucCode. The sequence is Button 3 (SW3) - Button 3 (SW3) – Button 2 (SW2) - Button 3 (SW3) – Button 2 (SW2). A flash of LED D9 signals an accepted key press.

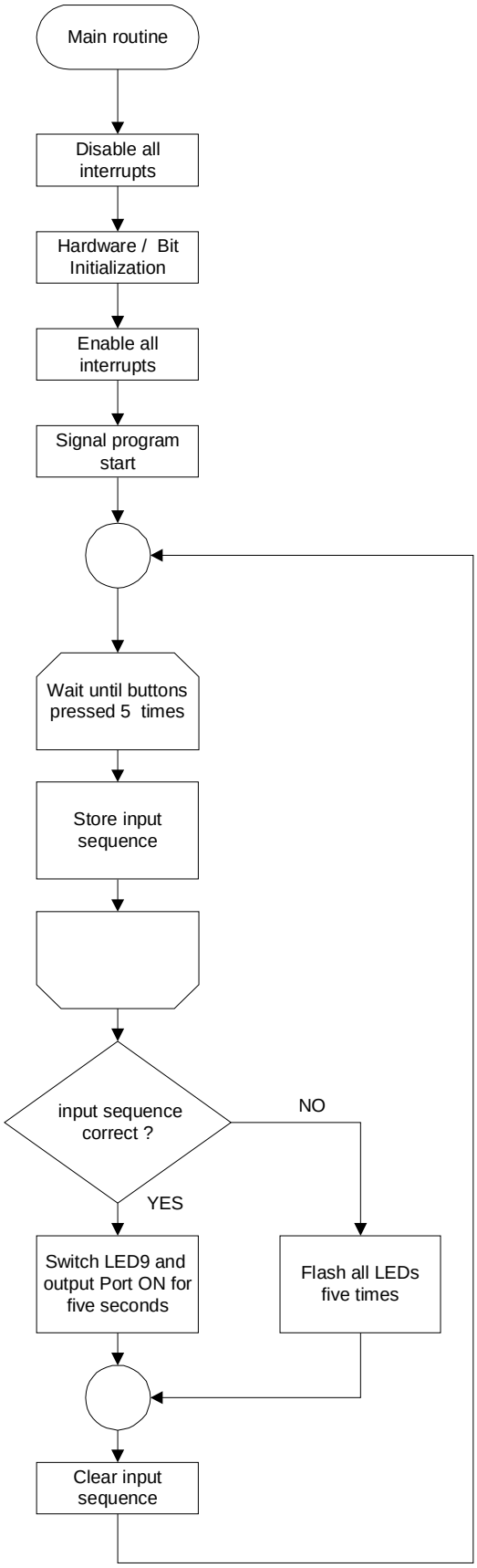
Timer P1 (TMP1) is working in the modus 'clear and start on match between TMP1 and TP1CCR0'. According to selected operating frequency and the value of the compare register TP1CCR0 TMP1 generates a 25ms timebase.

| Used Internal Peripherals | Used External Parts | Source Modules | |
|---|---|---|---|
| Timer P0 (TMP0) | LEDs | codechk.c | Main function |
| Timer P1 (TMP1) | Button2 (SW2) | init.c | Hardware and bit initialization |
| | Button3 (SW3) | interrupt.c | interrupt functions |
| | | | |

To end the program and return to the download program again please press the START button.

Flowcharts of the main routine and the interrupt functions:




Main routine
Disable all interrupts
Hardware / Bit Initialization
Enable all interrupts
Signal program start
Wait until buttons pressed 5 times
Store input sequence
input sequence correct ?
NO
YES
Switch LED9 and output Port ON for five seconds
Flash all LEDs five times
Clear input sequence

### 7.7 Count down timer

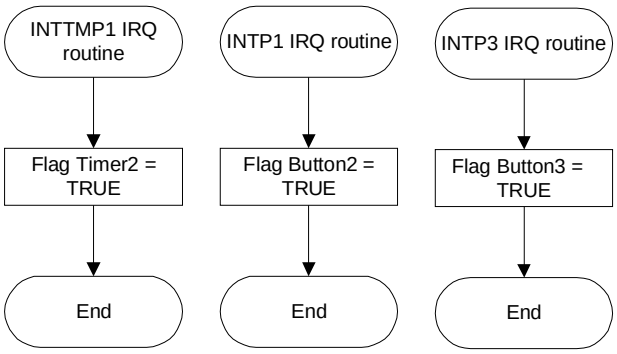
With this sample programs the board can be used as a count down timer. After the signal for program start the program waits for the input of the timer time, which can be set by button 3 (SW3). Each press increases the time by 60 seconds. The current time is shown by the LEDs in units of minutes. A press of button 2 (SW2) starts the timer, which is shown by a flashing LED D9.
After the selected time is finished, all LEDs flash for 3 minutes or a key press of the user. Then the program goes to stand by modus. A key press during countdown time or stand-by-modus sets the program to input mode again.

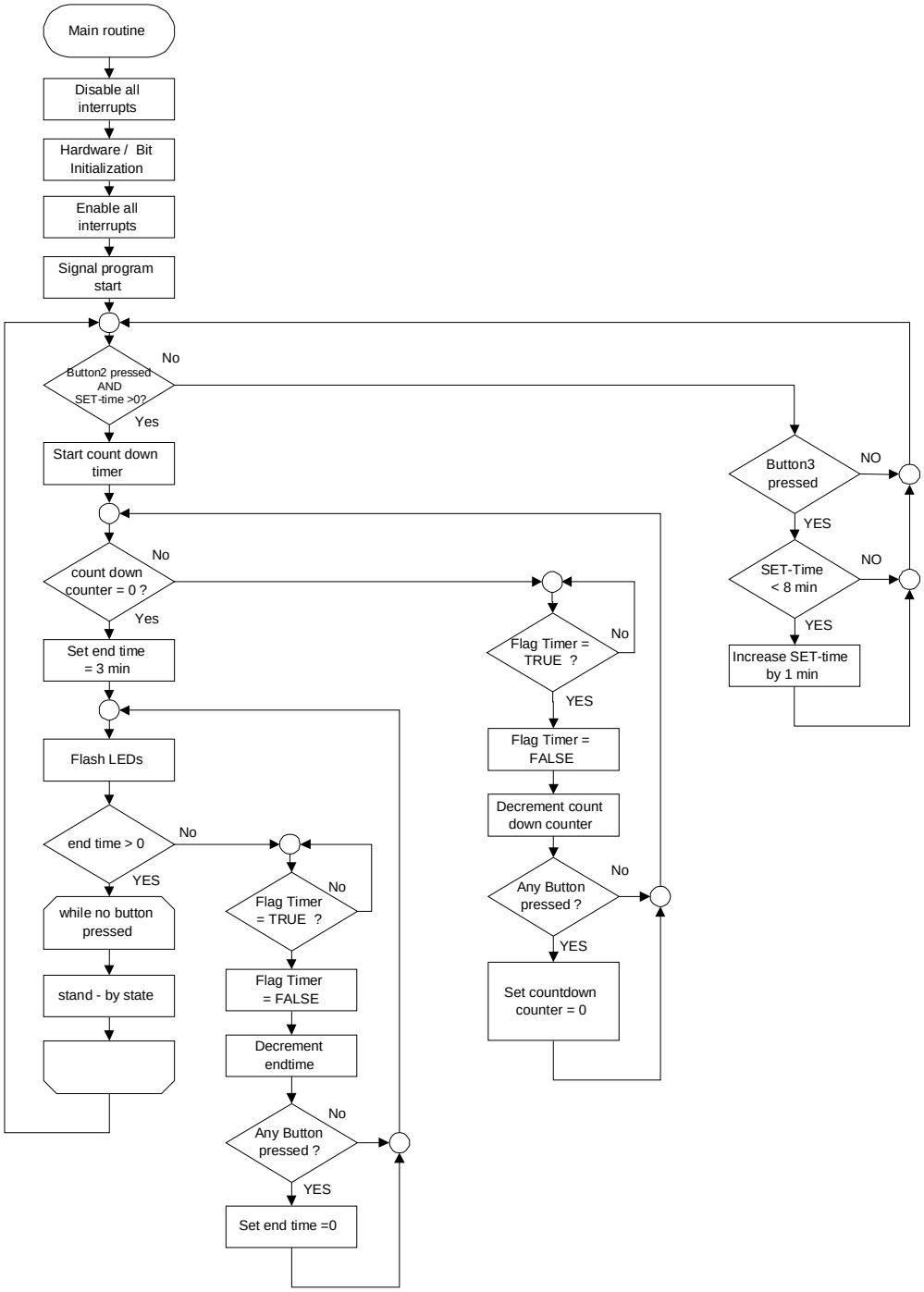
Timer P1 is working in the modus 'clear and start on match between Timer P1 (TMP1) and TP1CCR0'. According to selected operating frequency and the value of the compare register TP1CCR0 Timer P1 generates a 25ms timebase.

| Used Internal Peripherals | Used External Parts | Source Modules | |
|---|---|---|---|
| Timer P0 (TMP0) | LEDs | timer.c | Main function |
| Timer P1 (TMP1) | Button 2 (SW2) | init.c | Hardware and bit initialization |
| | Button 3 (SW3) | interrupt.c | interrupt functions |
| | | | |

To end the program and return to the download program again please press the START button.

Flowcharts of the main routine and the interrupt functions:




Main routine
Disable all interrupts
Hardware / Bit Initialization
Enable all interrupts
Signal program start
Button2 pressed AND SET-time >0?
No
Yes
Start count down timer
count down counter = 0 ?
No
Yes
Set end time = 3 min
Flash LEDs
end time > 0
No
YES
while no button pressed
stand - by state
Flag Timer = TRUE ?
No
Flag Timer = FALSE
Decrement endtime
Any Button pressed ?
No
YES
Set end time =0
Flag Timer = TRUE ?
No
YES
Flag Timer = FALSE
Decrement count down counter
Any Button pressed ?
No
YES
Set countdown counter = 0
Button3 pressed
NO
YES
SET-Time < 8 min
NO
YES
Increase SET-time by 1 min

### 7.8 Melody maker

This sample programs plays one of eight predefined melodies using an external piezo buzzer. By pressing button3 (SW3) the next melody is selected. Button2 (SW2) selects the previous melody. Every melody is completed before the next melody starts.

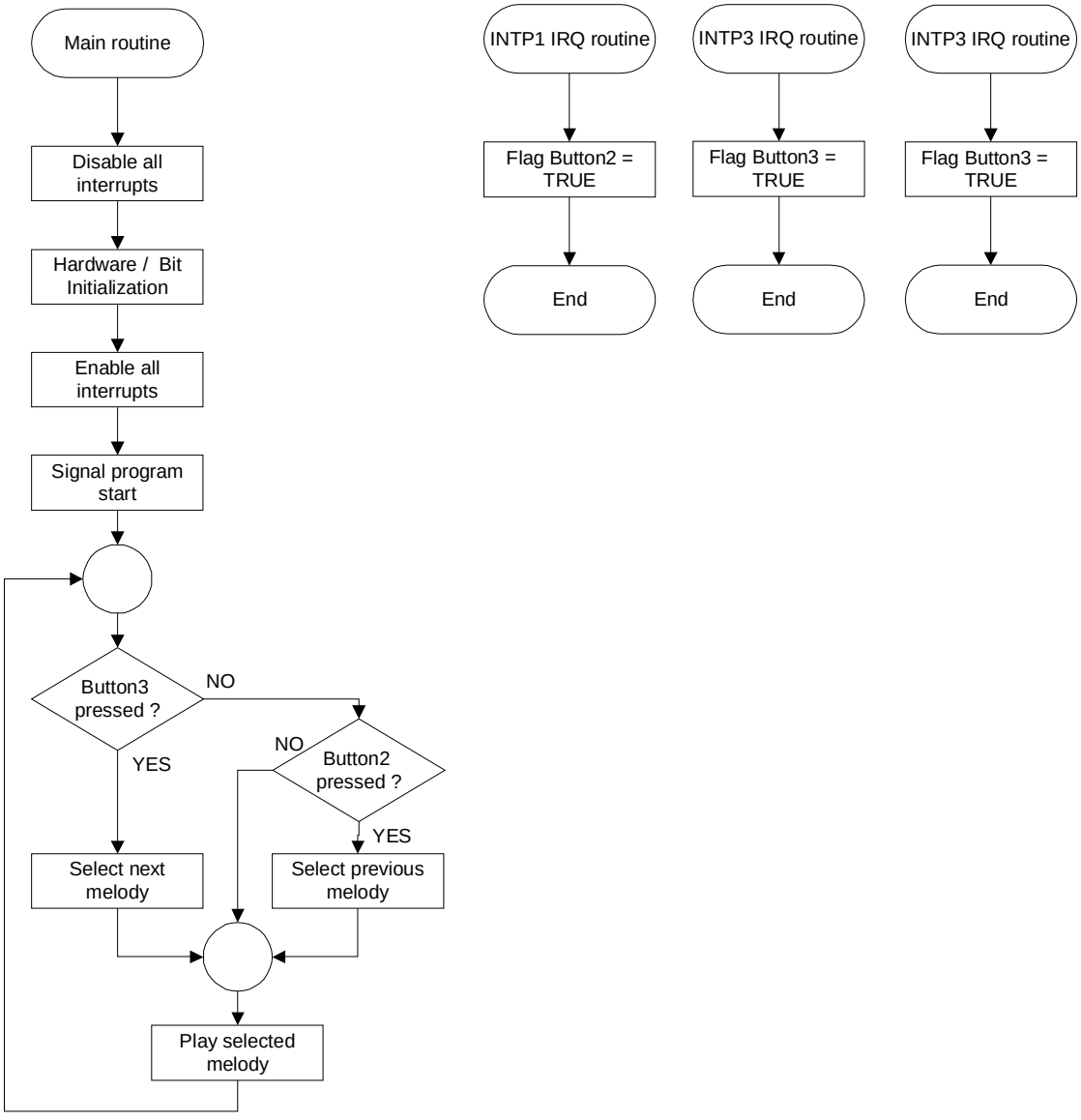
**Note:** The external buzzer must not exceed the limit of 10 mA current consumption. It must be connected between the port pin P4.1 and $V_{CC}$.

Timer P1 (TMP1) is working in the modus 'clear and start on match between Timer P1 (TMP1) and TP1CCR0'. According to selected operating frequency and the value of the compare register CCR0 Timer P1 generates a 27.5ms timebase.

| Used Internal Peripherals | Used External Parts | Source Modules | |
|---|---|---|---|
| Timer P0 (TMP0) | LEDs | melody.c | Main function |
| Timer P1 (TMP1) | Button2 | init.c | Hardware and bit initialization |
| | Button3 | interrupt.c | interrupt functions |
| | | | |

To end the program and return to the download program again please press the START button.

Flowcharts of the main routine and the interrupt functions



### 7.9 CAN Example

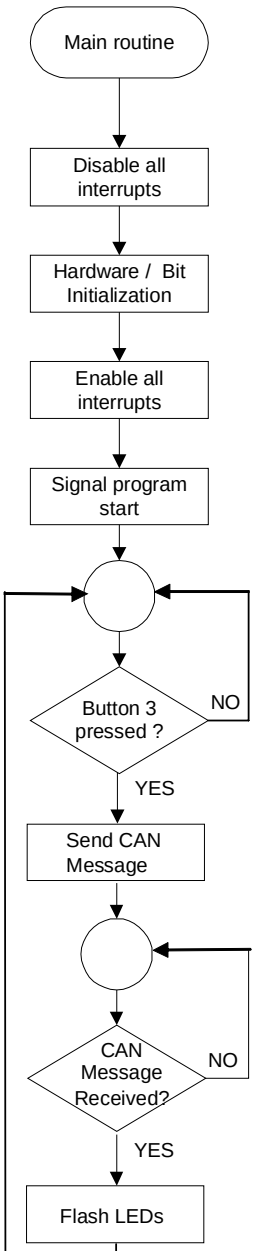
This sample programs sends a CAN frame and waits for another. It can be used together with another Drive it or together with the CANoe program supplied on the CD. After the program-start-signal, the program waits until Button3 is pressed. By pressing button3 a CAN message is sent. When this message is received by the remote node (e.g CANoe or another Drive it) this node sends a confirmation frame. When this confirmation message has been received by Drive it the LEDs will flash 5 times.

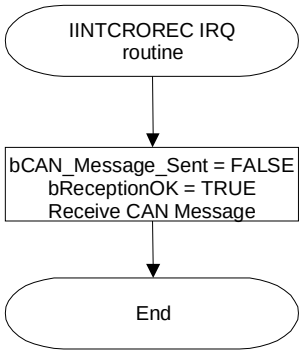
Timer01 is working in the modus 'clear and start on match between TM01 and CR001'. According to selected operating frequency and the value of the compare register CR001 Timer01 generates a 25ms timebase.

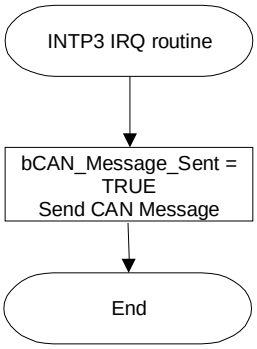
| Used Internal Peripherals | Used External Parts | Source Modules | |
|---|---|---|---|
| TimerP00 | LEDs | afcan.c | Main and afcan functions |
| AFCAN0 | Button3 | init.c | Hardware and bit initialization |
| | | interrupt.c | Interrupt functions |
| | | | |
| | | | |

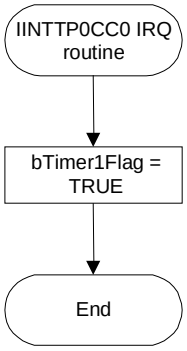
To end the program and return to the download program again please press the START button.

Flowcharts of the main routine and the interrupt functions:



# 8. Connectors and Cables

## 8.1 Serial host connector


**Figure 15: Target cable 2**

| Pin | 78K0/Fx2 | V850ES/Fx2 | RS232 Signal on *F_Line - Drive it!* |
|---|---|---|---|
| 1 | NC | NC | --- |
| 2 | RXD6 | RXD0 | RxD |
| 3 | TXD6 | TXD0 | TxD |
| 4 | NC | NC | DTR |
| 5 | NC | NC | GND |
| 6 | NC | NC | DSR |
| 7 | P17 | P33 | CTS |
| 8 | P16 | P32 | RTS |
| 9 | NC | NC | ---- |

| Pin | RS232 Signal on PC side |
|---|---|
| 1 | DTR |
| 2 | RxD |
| 3 | TxD |
| 4 | DSR |
| 5 | GND |
| 6 | --- |
| 7 | CTS |
| 8 | RTS |
| 9 | --- |

**Figure 16: Serial host connector**

## 9. Schematics

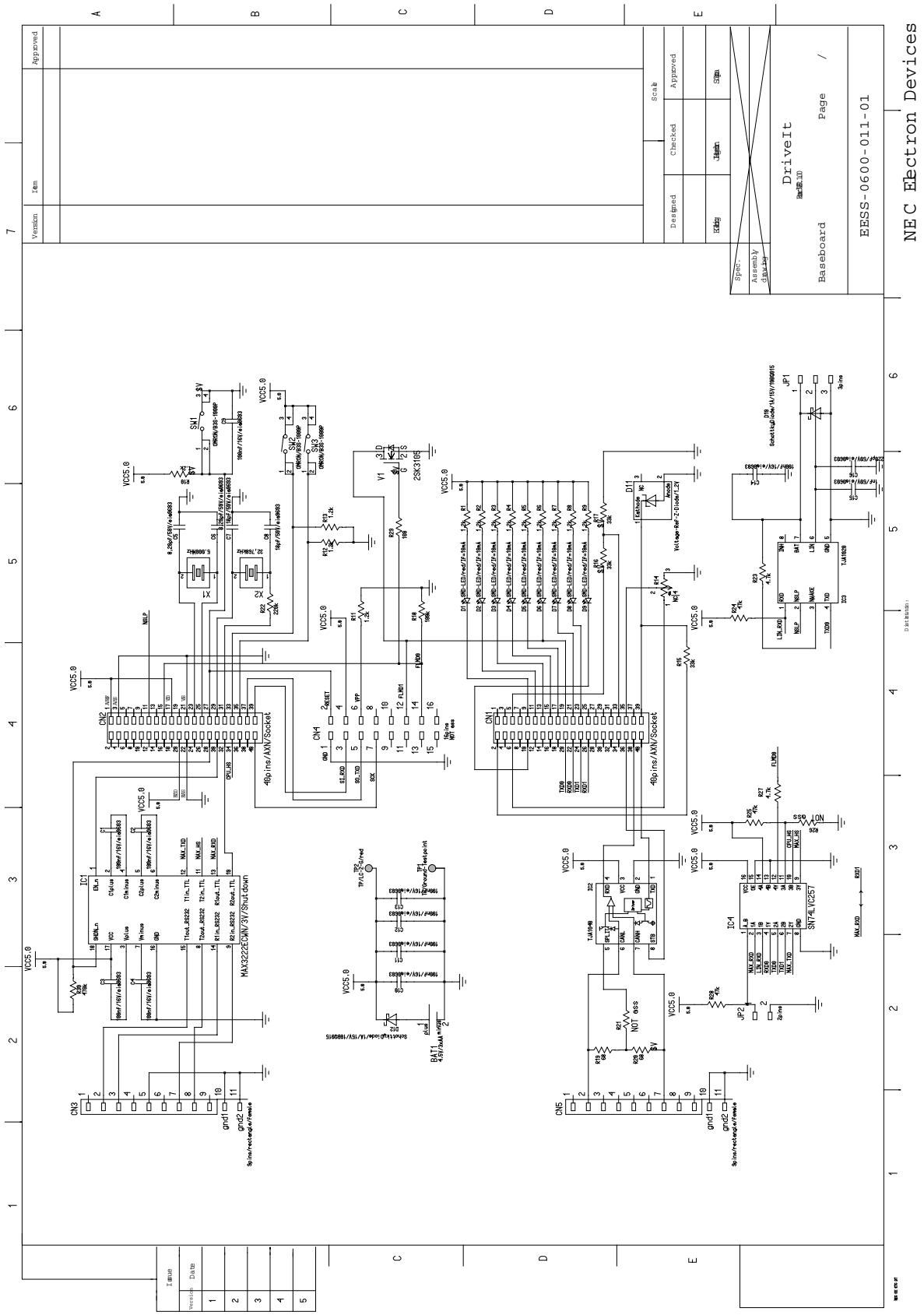



Figure 17: *F_Line - Drive it!* baseboard schematics

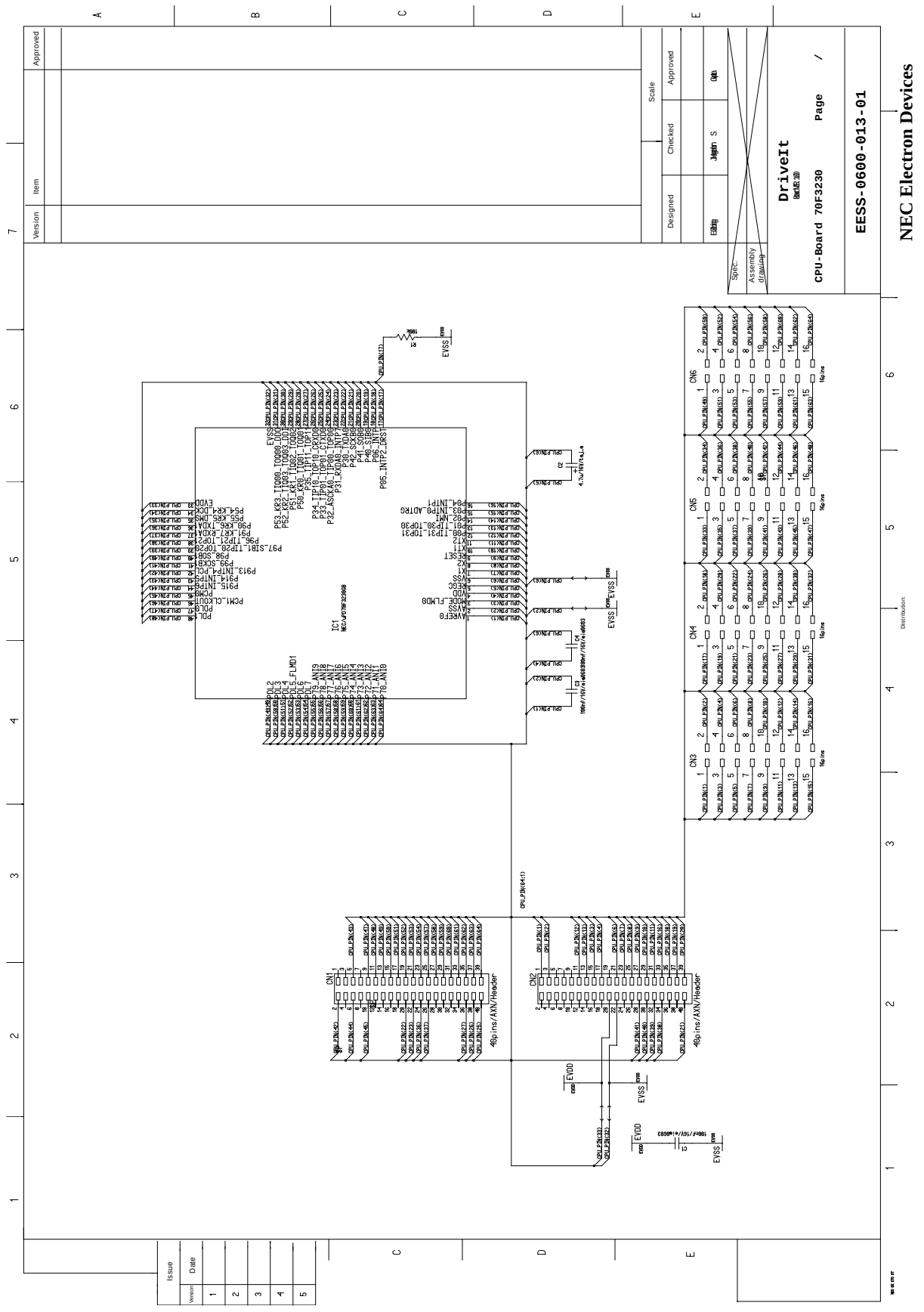



**Figure 18: *F_Line - Drive it!* V850ES/FE2 CPU module schematics**

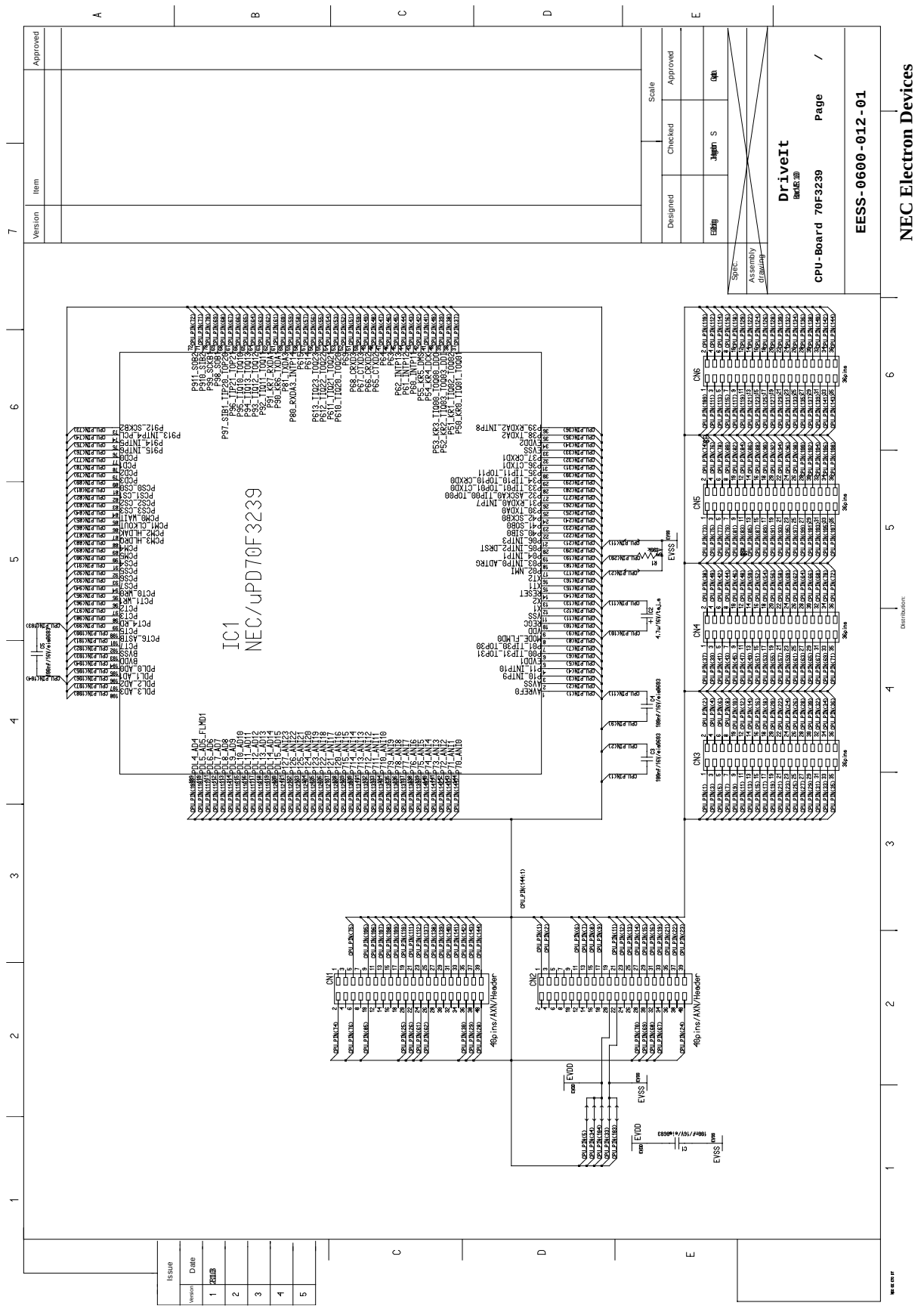



**Figure 19: *F_Line - Drive it!* V850ES/FJ2 CPU module schematics**