User Manual
for
COP8-REF-AME

Reference design for
COP8-REF-AME

Table of Contents:

1 Brief Description of System

COP8-REF-AME Preliminary Product Requirement Spec Sep. 15, 2001

Features:

- Stand-alone multi-meter reference design with COP8AME as system processor.
- Capability to connect external voltage and current sources to measure external voltages in the range of 0 to 19.9V and current in the range of 0 to 20mA. An AGC (Automatic Gain Control) is implemented which is under software control, supporting higher accuracy.
- There is a voltage and a current source installed in order to allow standalone operation.
- PC-software with application information, source code, circuit schematics and bill of materials is provided in the package.
- Demonstrates the built in AD-converter capability of the COP8AME.
- Communication protocol is supported on the board (RS-232).
- Both in-circuit programming and in-circuit emulation supported on the reference board.
- Sensors for light and humidity measurements are on board.
- Pushbuttons are included for controlling purposes.
- Programming interface support.

2 General Description

The reference design is intended for several different purposes. The main goal is to show the built-in capabilities of the COP8AME microcontroller. The features include dual-clock, high-speed PWM timer, 6 channel AD converter, programmable gain amplifier, temperature sensor and also the built-in USART.

Another goal is to create a versatile platform for developers that want to test their own code. It is possible with Flash feature of COP8AME and can be done using PC-software to re-program the device. There is also Microwire™ bus connector with standard spacing on the card to enable the user to hook on his own devices to COP8AME.

3 Description of Firmware Functions

The user has the capability to use the reference design as a stand-alone multimeter. Below is a brief description of the different functions of the multimeter. In each paragraph there is also a description of the hardware and software used for each function. Please read the dedicated paragraph for a more detailed look into the hardware design (see p. 6).

A user can select the analogue channel to read. The possible channels are voltage input, current input, light sensor, humidity sensor, atmospheric pressure sensor and the on-chip temperature sensor. It is also possible to read voltage and current from external sources or from the voltage and current sources that are installed on board.

The firmware software is controlled via two push buttons (labeled YES and NO). To leave the startup menu the user has to push one of the buttons. The menu system has two levels, one base menu and one submenu. In the base menu the user can select if he wants to enter the mode displayed or not. By choosing NO the user will go to the next mode selection. If the user pushes YES button the next submenu will be displayed or a specific mode will be executed. The submenus and operating modes are described below. Additionally when LEDs are connected to COP8AME ports through J5 and J8, green LED flashes (once per second) when measurement is performed and also if YES button is pressed. Red LED lights when NO button is pressed. Both LEDs are not active for humidity measurements.

More information about the COP8 microcontrollers and the latest updates of tool and reference designs can be found at National Semiconductors microcontroller product site:

http://www.national.com/cop8

A lot of other useful information can also be found on the COP8 CD.
This CD can be supplied by your local distributor or National FAE.

Mode number	Mode description
1	Measure light
2	Measure temperature
3	Measure external voltage
4	Measure external current
5	Measure on-board voltage
6	Measure on-board current
7	Measure humidity
8	Measure atmospheric pressure

3.1 Light meter

The light meter mode samples the input from the photodiode (D3) and presents the result of AD conversion on the display. The value shown on the display is the ADC value without any AGC (Automatic Gain Control) implemented. The photodiode voltage is amplified to cover the range from dark to full sunshine. In order to increase or decrease gain of the amplifier it is up to the user to change the resistor R20 (default 300k Ohm). To enable light measurement a jumper must be placed on J14.

3.2 Temperature meter

The temperature meter uses CO8AME built-in temperature sensor and has a resolution of 0.5°C. The temperature to voltage conversion is given by the following equation:

$$V_{out}[V] = -8\left[\frac{mV}{^{\circ}C}\right] * T[^{\circ}C] + 1.625[V]$$

There is no calibration used for temperature sensor. For more accurate measurements of temperature it is up to the user to write complementary code (modify equation). More information about the built-in temperature sensor can also be found in COP8AME specification.

3.3 External voltage meter

An user may connect an external voltage source to connector J10 and measure it. This input voltage must be in the range of 0 - 20 Volts. To enable external voltage measurement a jumper must be placed on J18 in “Ext. Volt” position.

3.4 External current meter

An user may connect an external current source to connector J10 and measure it. This input current must be in the range of 0 - 20 mA. To enable external current measurement a jumper must be placed on J19 in “Ext. Curr” position.

3.5 On-board voltage meter

A voltage source installed on board works in the range of 0 - 5V. The voltage can be changed using VR1 potentiometer. This mode can be used to check that the AD-converter is working. To enable on-board voltage measurement a jumper must be placed on J15. The on-board voltage is also present on connector J11.

3.6 On-board current meter

A current source installed on board works in the range of 0 – 10 mA. The current can be changed using VR2 potentiometer. To use the current source an user has to put a jumper on J7 connector or connect an external circuit to J7. Otherwise the source will not be connected to ADC input.

J7 connector can also be used to connect an external current meter to check the accuracy of the AD - converter.

3.7 Humidity meter

The humidity meter is able to measure humidity from 20 to 100 RH. It uses approximation method for AD values read (described in detail in the flowcharts). Due to the fact that humidity sensor must be driven by alternating current, meter uses four pins (two for feeding and two for differential measurement) connected to J19, J18, J5 and J8 connector. Therefore these connectors must be shorted in position “Humidity”.

3.8 Pressure meter

The main part of pressure meter is MPXA4115A atmospheric pressure sensor. It could measure the pressure from 15 to 115 kPa. For AD values read a mathematical equation is applied:

$$V_{out}[V] = V_s[V] * (0.009 * P[kPa] - 0.095)$$

No calibration is used for this sensor. To make this measurement connector J6 must be shorted.

4 Description of Communications Interface

There is TIE/EIA-232 communication interface present on board. It is described below. Short description of the communication protocol implemented in the firmware is available in the following section as well.

4.1 TIE/EIA-232 (RS-232)

To be able to communicate with PC or other reference boards an RS-232 circuit is present on board. This is also the default interface for the host PC - software. The COP8AME hardware USART is capable to communicate with speeds of up to 38,400bps.

4.2 Communication protocol

A simple protocol has been created for PC - software communicating with the reference board. The communication protocol is described in the table below. Each command from the host PC has a corresponding acknowledge. For example, the “Set Clock Command” will be transmitted to the reference board as first byte 05h, second byte 05h and the actual time set in the host software window on three bytes. Then the reference board will reply with first byte 04h, second byte 05h and then the time as interpreted by the firmware sent back on three bytes (first hour then minutes and last seconds).

Signal	Signal No.	Length	Data
Test Loop	0x00	0x02	
Get AD Value Cmd.	0x01	0x03	Channel
AD Value Resp.	0x02	0x07	Channel | AD Val. (HB) | (LB) | Dec Val. (HB) | (LB)
Get Clock Command	0x03	0x02	
Clock Response	0x04	0x05	Hours | Minutes | Seconds
Set Clock Command	0x05	0x05	Hours | Minutes | Seconds
Get ID Command	0x06	0x02	
ID Response	0x07	0x06	Letter(1) | … | Letter(4)
Set ID Command	0x08	0x06	Letter(1) | … | Letter(4)

The other commands are executed in the same way. Just note that all the commands have a reply and it is up to the user whether to use this information or not.

4.3 Microwire™

The RJ11 connector (J13) on board allows to program COP8AME through ISP. Please read application note AN-1154 for more information on all functions of the ISP - program.
This connector can also be used to connect to the reference board other devices that are Microwire compatible. During ISP operations it is important to short J22 connector in “Normal” position.

5 Description of firmware modules

The firmware code may be found on the CD-ROM accompanying the reference board in the *Firmware*\ folder.
This section describes the firmware in detail.
There are flowcharts for the most useful routines in chapter 7. They present some of the functions in detail. That should help the user to understand the structure of the firmware application.

5.1 main.asm

This is the main file for the application. It contains the initialization of the system and includes all other files used in the firmware. It is important to notice that the main driver is used to guarantee a part of processor time to all the modules.

There are also three subroutines (*Store_Data*, *Clear_Flex_Bit*, *Read_Data*) in the main file. They show how one can use the built-in boot-ROM subroutines to reprogram parts of the flash memory. *Store_Data* is used for storing the clock value and the system ID into the flash memory. *Read_Data* is used for reading in flash memory stored system ID and the clock value from the last set clock command from PC-software. *Clear_Flex_Bit* is used for clearing the last part of the flash memory where the OPTION register is located. The intent behind this subroutine is to show a way of getting into the boot-ROM programming mode and allowing for in-circuit programming.

5.2 memory.inc

The memory include file contains all variables and constants' declarations used in the firmware applications. It also contains declarations and setup values for all ports and pin setups.

5.3 lcd.inc

The driver (*Display_Handler*) for the multimeter application is stored in the *lcd.inc* file. The file also contains the routines for running the dot-matrix display. There are flowcharts describing the code structure in chapter 7. *Init_LCD*, *Print_LCD*, *Put_LCD_Ctrl* and *Put_LCD* are important routines of this module. They are described in detail in the code. For a quick reference *Init_LCD* is used to initiate the dot matrix on board processor (please refer to the manufactures of dot-matrix displays for more information on this). *Print_LCD* is used to write a complete string onto the display. The strings are stored in ROM and there is an example of writing a string to LCD in *lcd.inc* file. Then *Put_LCD_Ctrl* is used to send a control signal to the display controller (line selection, clear screen and cursor home). The last subroutine that is useful to know about is the *Put_LCD* that will put the content of the accumulator on to the display.

5.4 uart.inc

In *uart.inc* file the driver (*Command_Handler*) for communications handling is located. This file also includes the handling of the send and receive interrupts. The receive interrupt will put the received byte into a 64 byte long buffer and the user can use the *Get_Byte_From_Buff* subroutine to get the bytes from the buffer.

The send interrupt works in a different way. It is up to the user to load the send buffer with data to send and then activate send interrupt. This routine will empty the send buffer and then turn of the send interrupt.

5.5 ad_conv.inc

ad_conv.inc contains functions for handling the AD-converter. For instance in *ad_conv.inc* file the routine *Read_AD_ch* is located. This routine will read the AD-channel set in the *ADSET* variable and then store the results in *ADRES1* (low eight bits) and in *ADRES2* (high two bits).

5.6 timer_x.inc

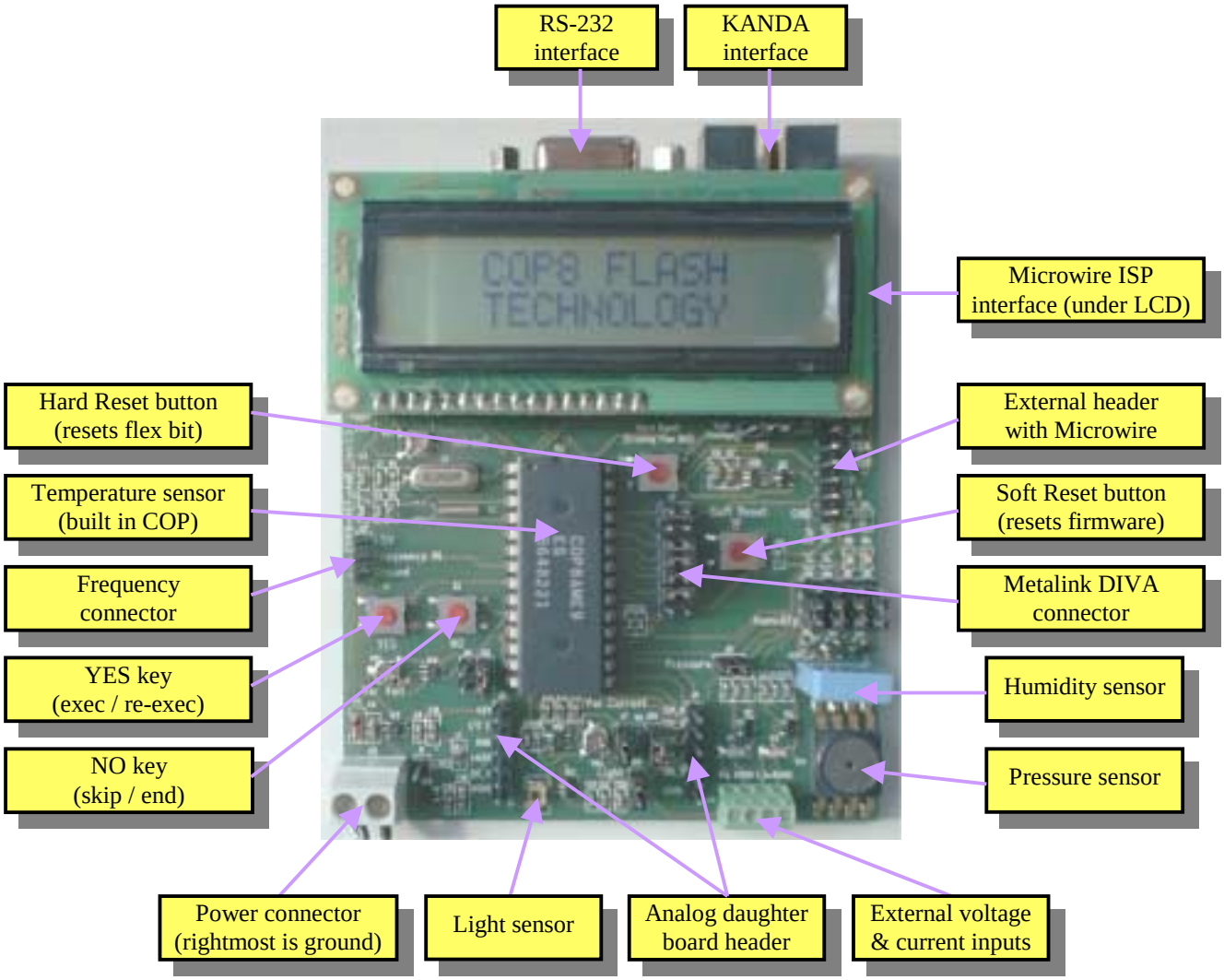
timer_x.inc file contains all functions used with the timers. First there is the timer zero interrupt handler (*Timer0_Int*). The timer zero handles the time counter (*Clock*). The control of the push buttons is also under timer zero control.
There are also functions for using timer one there. The frequency counter is implemented using timer one and the code for measuring the frequency is in *Timer1A_Int*. This function is written for user usage but not used in multimeter.
Timer two is used to generate 1 kHz signal through periodical level changing on L6 and G3 pin. This signal feeds humidity sensor. Timer T2 starts when user enters humidity measurement routine and stops immediately after exit. This simple generator code is written in the *Timer2A_Int* subroutine.

5.7 mathpack.inc

In the *mathpack.inc* file there is a collection of useful mathematical routines. There is one routine for 16-bit multiplication (*Bin_Mul_16*) and one routine for 16-bit division (*Bin_Div_16*). There are also routines for converting 16-bit binary to decimal digit values (*Bin_Dec*).
Conversion routines especially for temperature (*Convert_Temp*), humidity (Convert_Humidity) and pressure (*Convert_Pressure*) measurements are also written.

6 Hardware Description

Here is a description of the main hardware ideas behind the COP8-REF-AME reference design. This picture illustrates functions supported by this reference design. The schematics and bill of materials are on the provided CD-ROM.



6.1 Metalink DIVA connector

A user may use Metalink DIVA connector (J9) to write his own code and emulate it trough Metalink system. The design of this connection is done according to the specifications from Metalink. When firmware runs from Flash and not from emulator the DIVA connector should be jumpered (all pin pairs).

6.2 Header for connecting an analog daughter board

This header consists of two pin strip headers (J3 and J4) with standard pin spacing 2,54 mm. The spacing between the two headers is 2,54 mm x 12, to make it possible to use a standard prototyping board. Therefore user can create an analog daughter board in an easy way to test the stand-alone amplifier and the programmable gain amplifier built into COP8AME.

6.3 ISP-programming recommendation

It is very important that the PC-parallel port cable to the RJ11 connector on the reference board has to be disconnected when resetting the board to get into ISP mode. If the cable is connected high levels on the parallel interface pins will prevent the COP8 flash from performing a correct reset.

6.4 Frequency input

There is connector (J17) on reference board that may be used to measure input frequencies using high speed timer. This is done by connecting one of the two high-speed timer capture inputs to J17.

6.5 Sensors for weather measurements

Sensors for weather measurements are installed on the reference board. They are humidity (IC2) and pressure (IC4) sensors. Their outputs are connected to two of the A/D inputs through an external connector.

7 Board configuration

The jumpers configuration is shown in a table below.

Measurement	Jumpers									
	J5	J6	J7	J8	J11	J14	J15	J18	J19	J20
Light	up	x	x	up	x	1	x	x	x	x
Temperature	up	x	x	up	x	x	x	x	x	x
External voltage	up	x	x	up	x	x	x	up	x	x
External current	up	x	x	up	x	x	x	x	up	x
On-board voltage	up	x	x	up	0	x	1	x	x	x
On-board current	up	x	1	up	x	x	x	x	x	1
Humidity	dn.	x	x	dn.	x	x	x	dn.	dn.	x
Atmospheric pressure	up	1	x	up	x	x	x	x	x	x

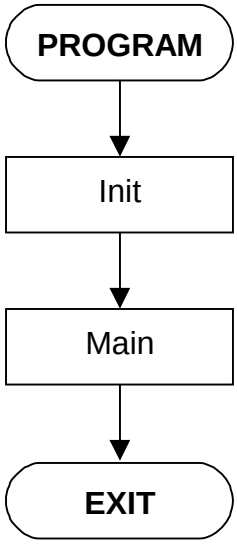
1 – closed
0 – open
x – free

up – upper position
dn. – bottom position

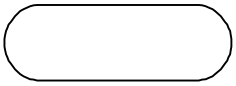
Except configuration shown above all pin pairs of connector J9 must be jumpered for correct operation of COP8-REF-AME Reference Design Board.
To use ISP, connector J22 should be shorted in rightmost position. Opposite position is used for resetting Flex bit.

8 Flowcharts

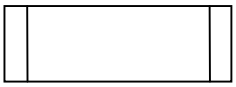
8.1 Program



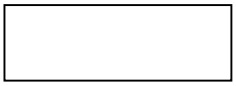
This is the whole application flow. Below the Program flow is a description of the syntax used in the flowcharts in this document. The flowchart is a complement to the comments in the code and should be used as a support while reading the code.
Location: main.asm



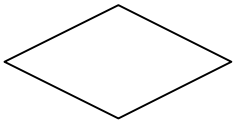
= START / STOP



= SUBROUTINE

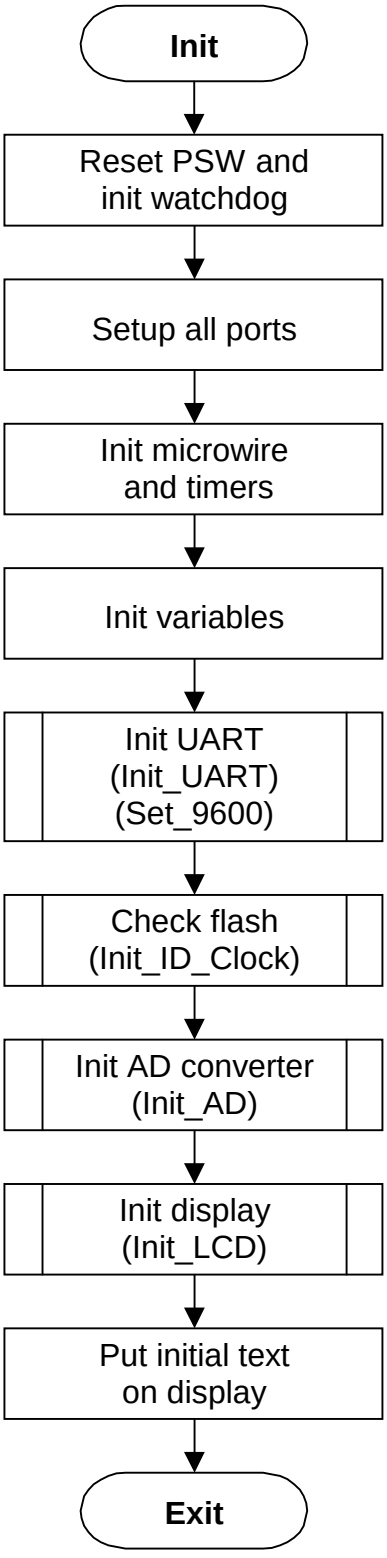


= ACTION



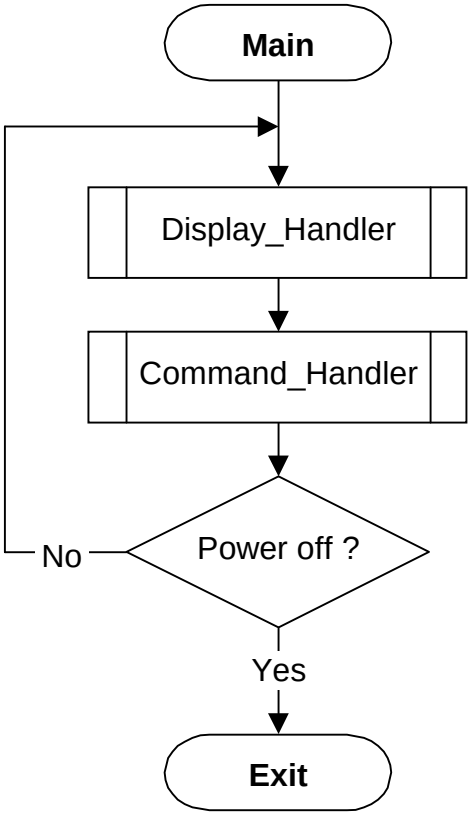
= QUESTION

8.2 Init



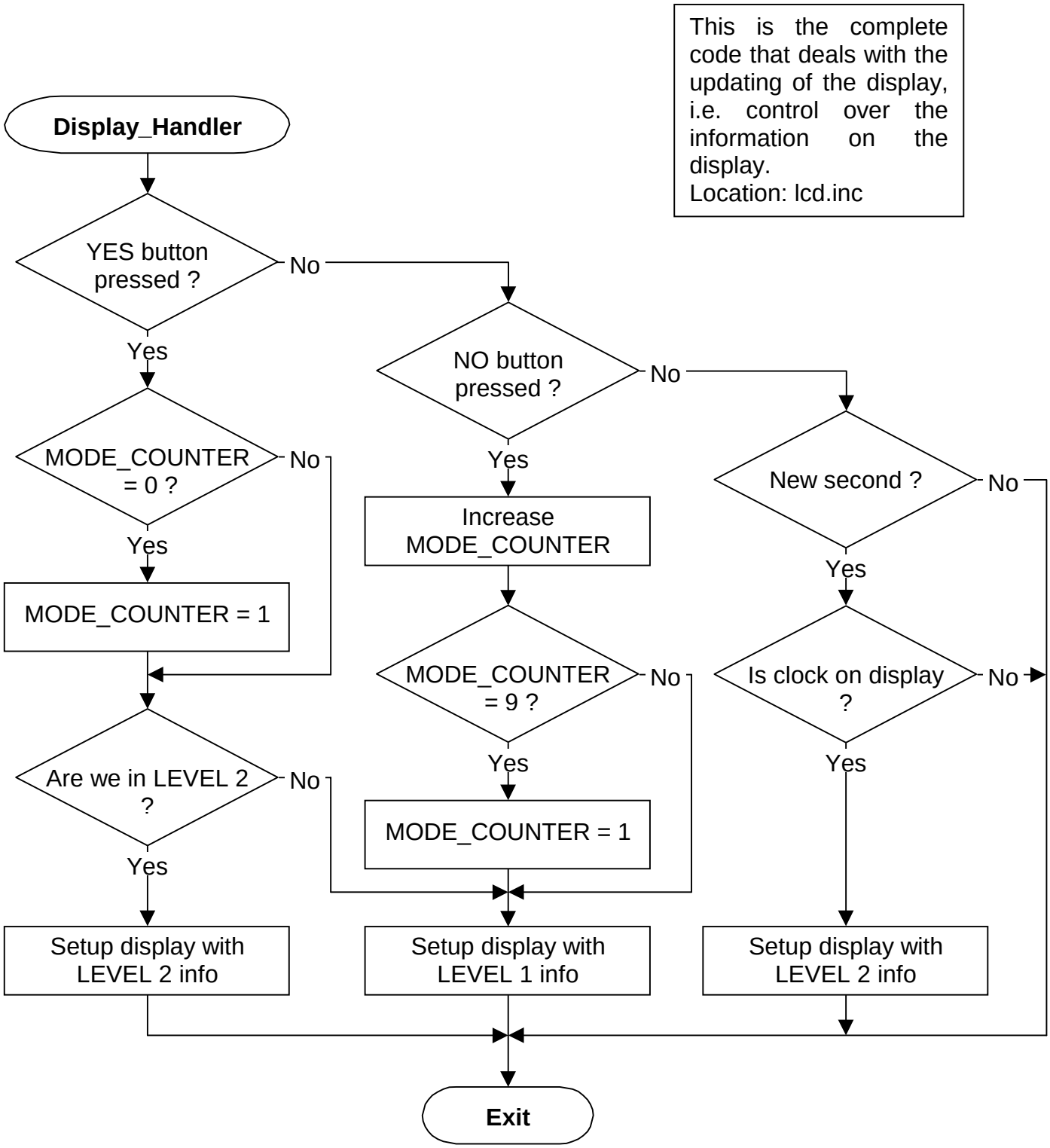
This is the way the system is initialized. This is in our opinion a good way to start up your own software.
Location: main.asm

8.3 Main

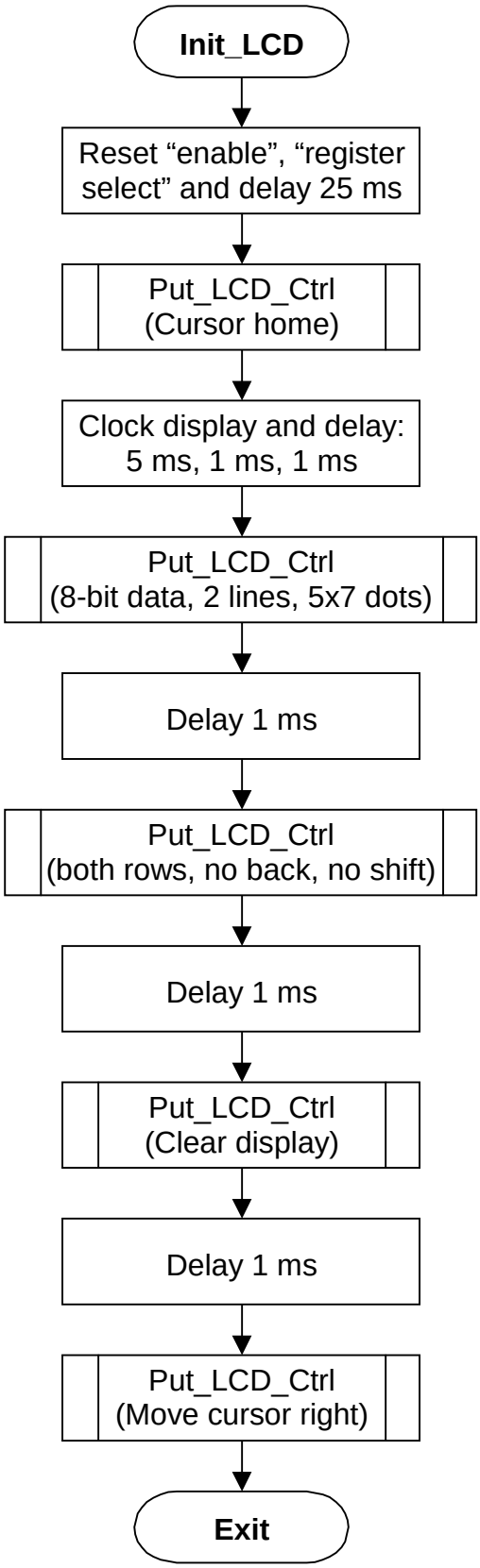


This is the forever loop that secures that all the resources get their shares of processing time.
Location: main.asm

8.4 Display handler

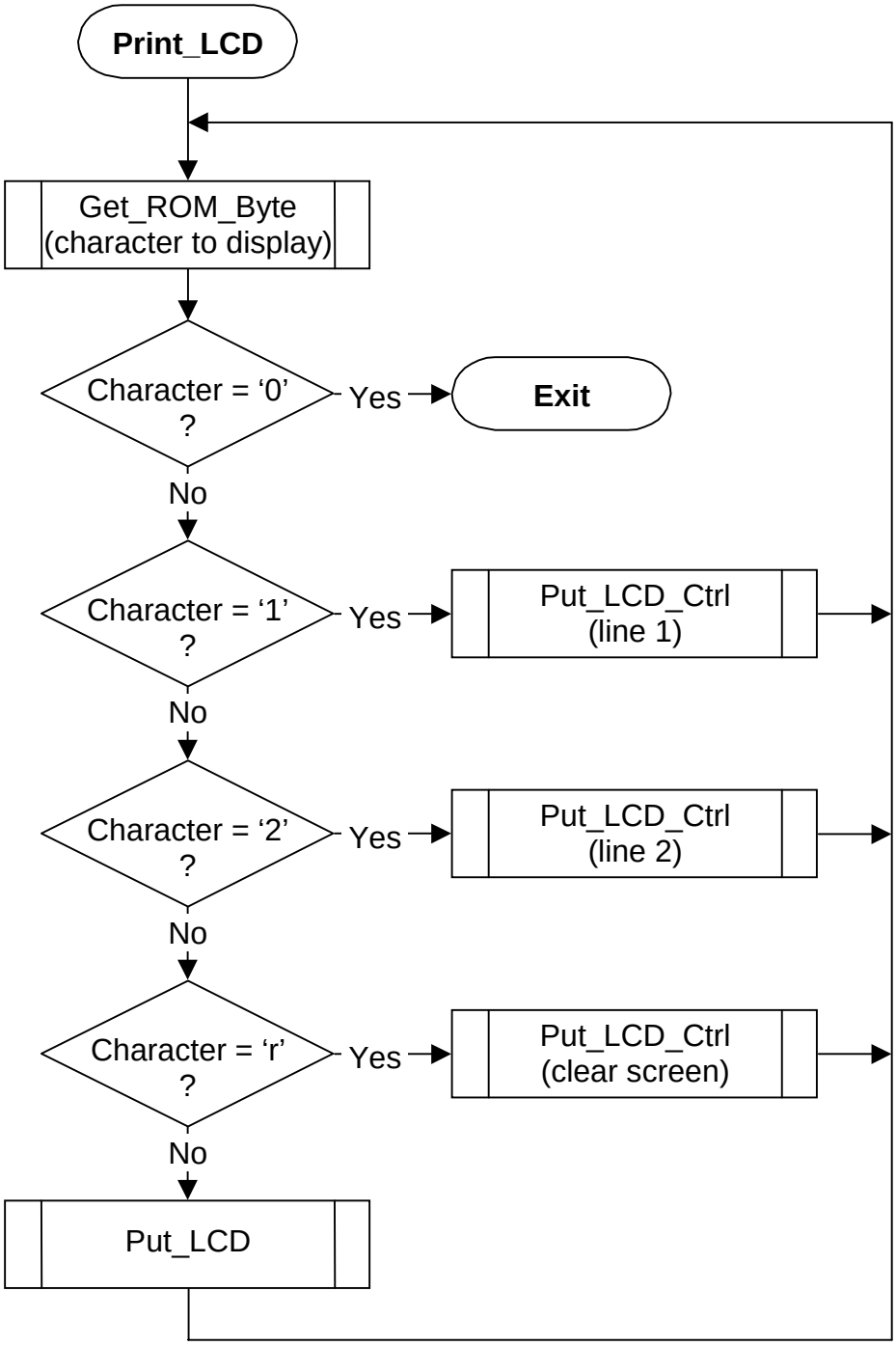


8.5 Init_LCD



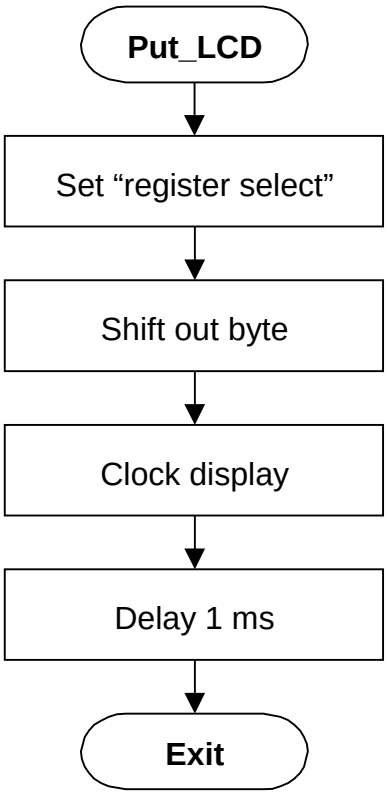
This is a description of how initiation of the display controller is handled.
Location: lcd.inc

8.6 Print_LCD



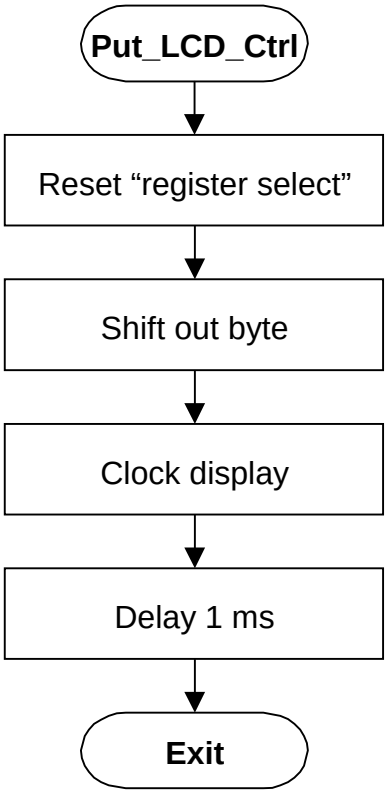
This is the flow for writing a string stored in ROM onto the display. Please not that the string must be null terminated (/0) otherwise the routine won't stop. Note also that digit 1 and 2 is used for determining which line to print on. Location: lcd.inc

8.7 Put_LCD



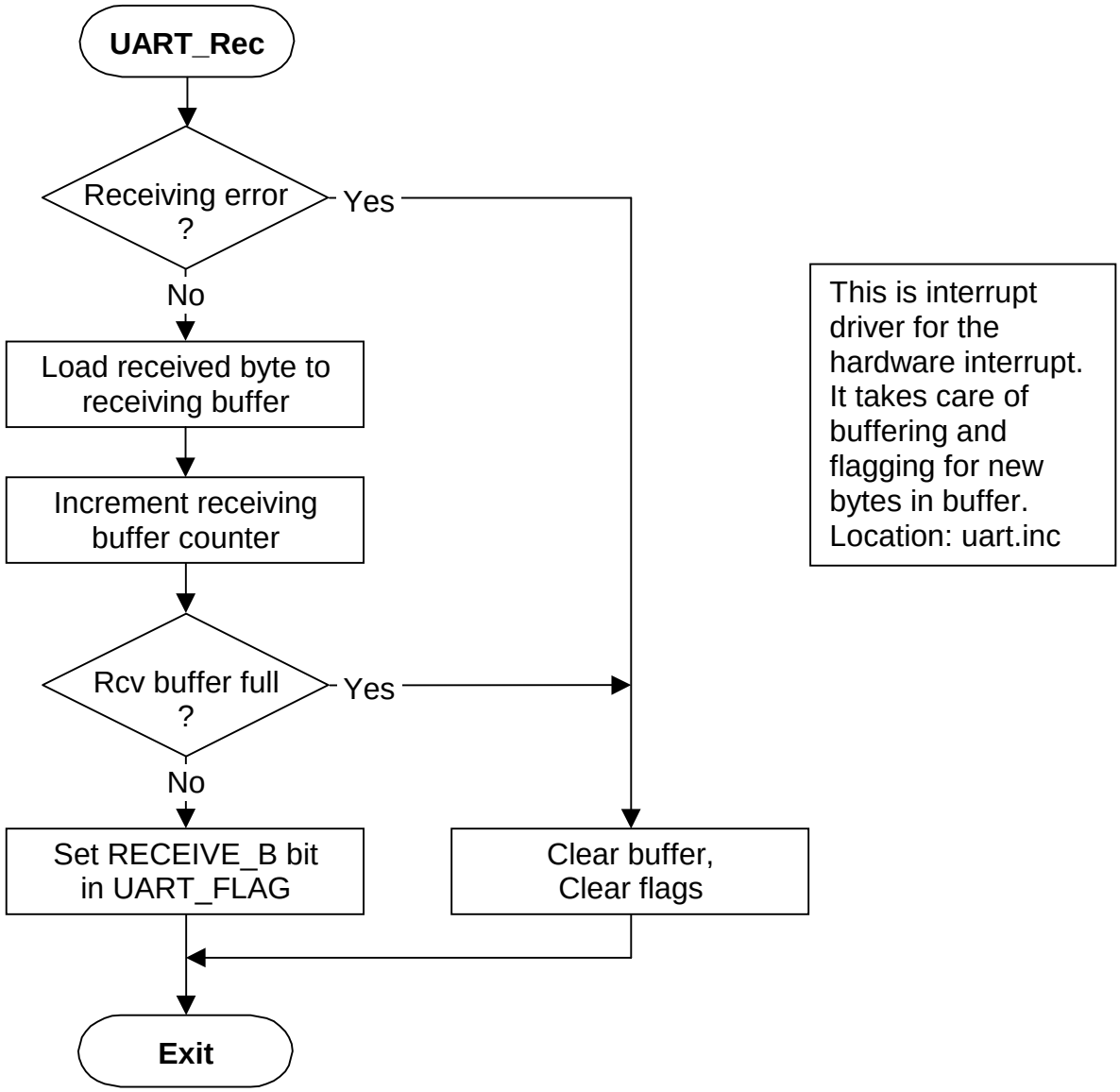
This routine is used for shifting out one byte of display data.
Location: lcd.inc

8.8 Put_LCD_Ctrl

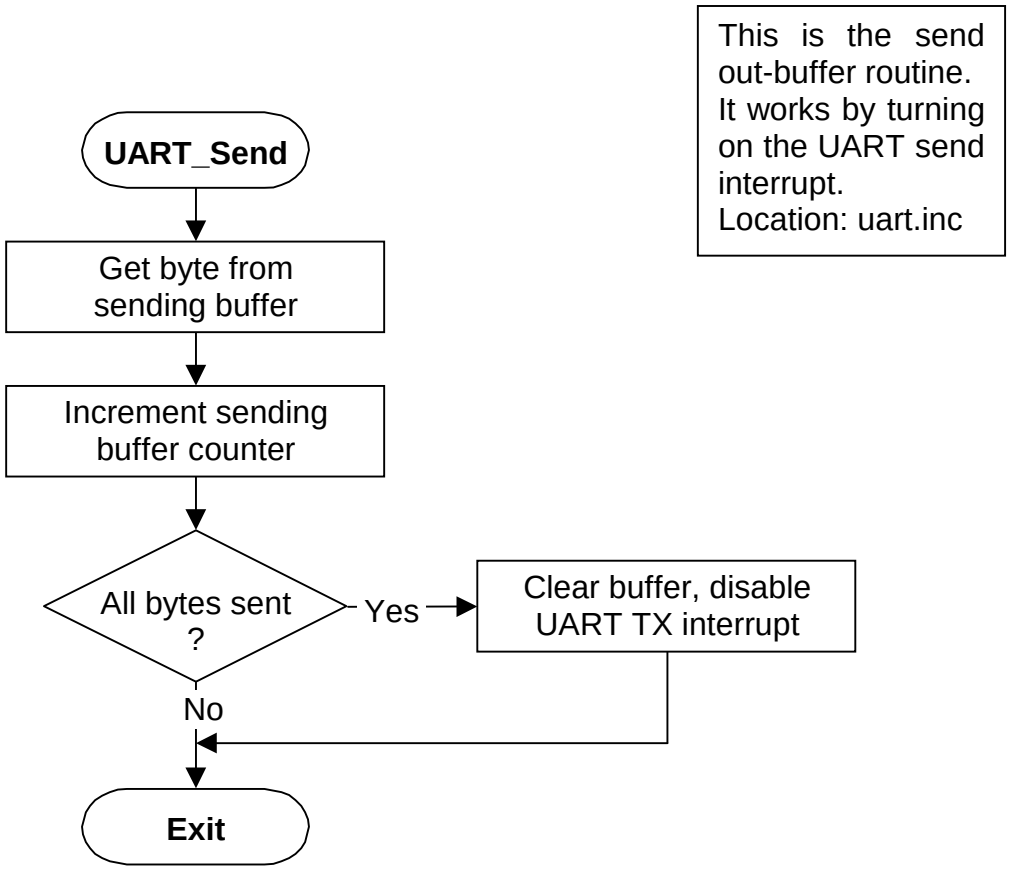


This is the way control signals are sent to the display.

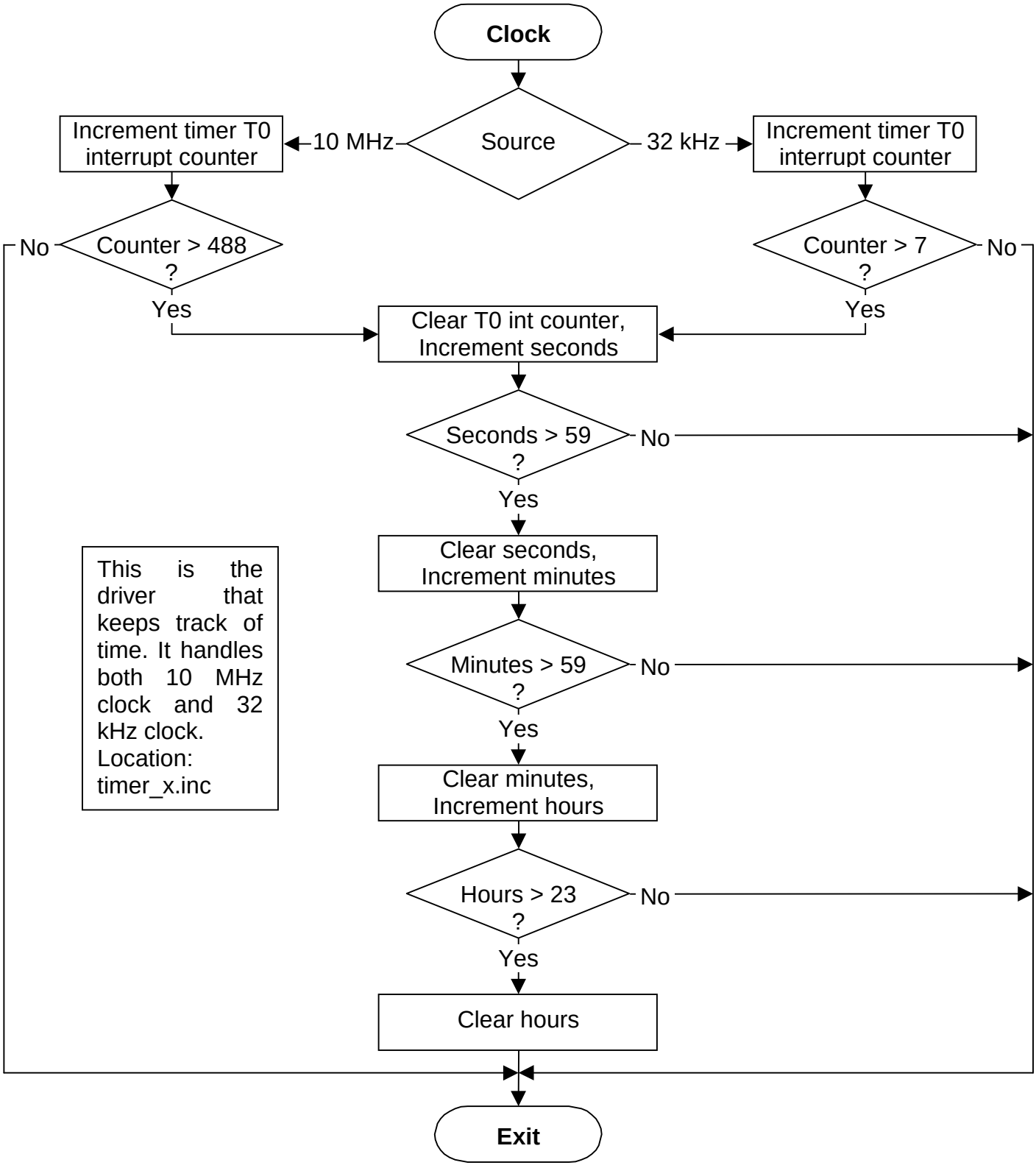
8.9 UART_Rec


UART_Rec
Receiving error ?
Yes
No
Load received byte to receiving buffer
Increment receiving buffer counter
Rcv buffer full ?
Yes
No
Set RECEIVE_B bit in UART_FLAG
Clear buffer, Clear flags
Exit
This is interrupt driver for the hardware interrupt. It takes care of buffering and flagging for new bytes in buffer. Location: uart.inc

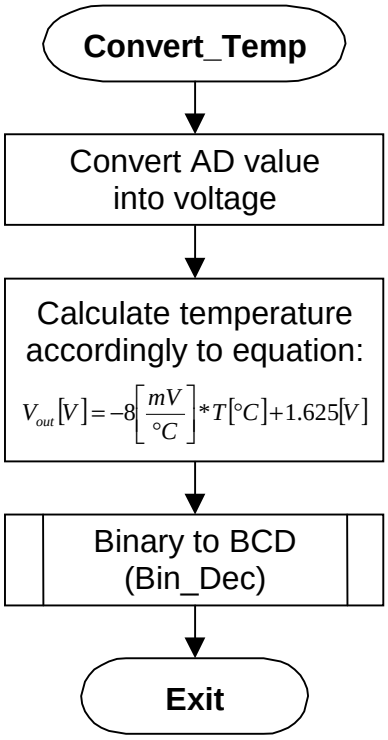
8.10 UART_Send



8.11 Clock

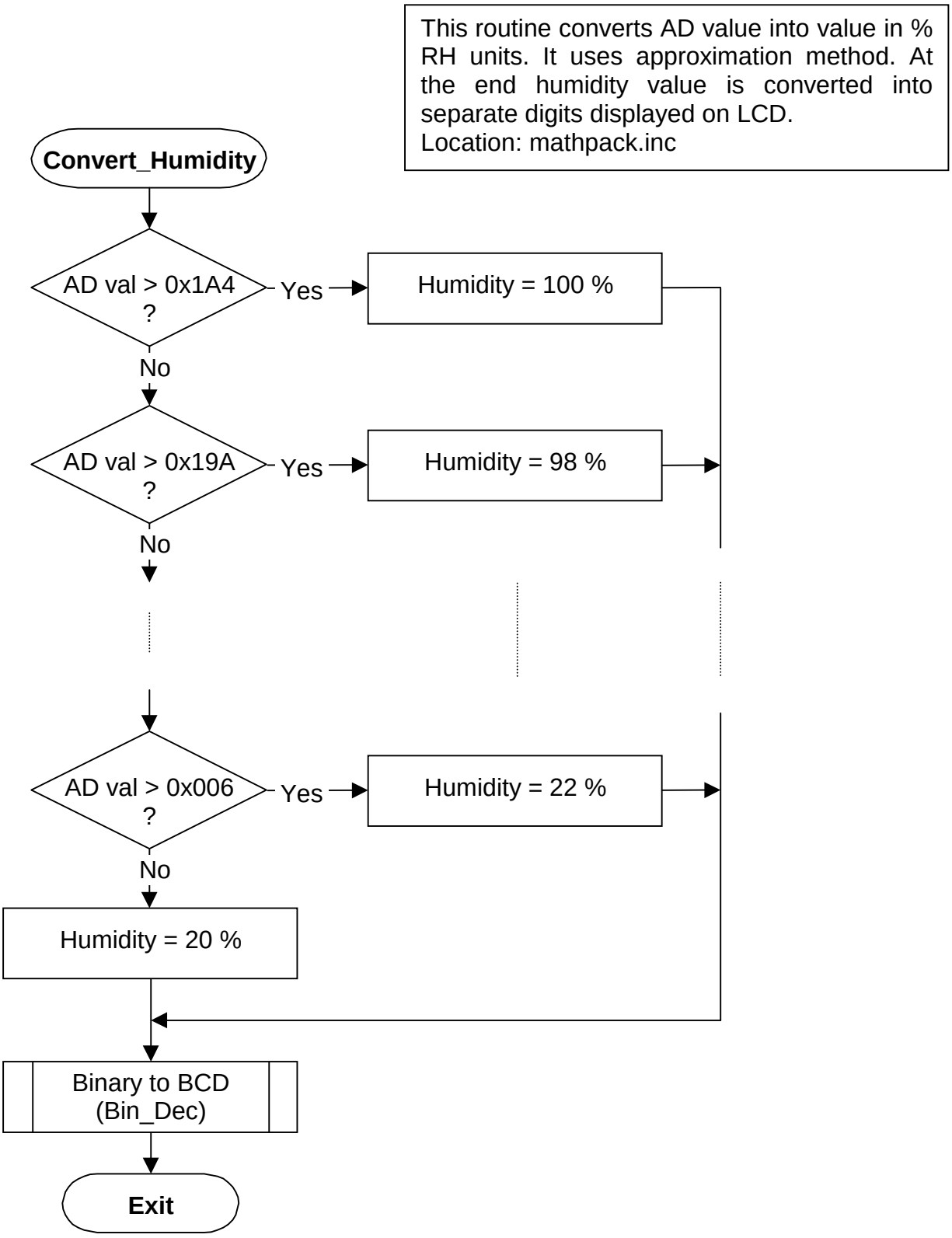

Clock
Source
10 MHz
32 kHz
Increment timer T0 interrupt counter
Increment timer T0 interrupt counter
Counter > 488 ?
No
Yes
Counter > 7 ?
No
Yes
Clear T0 int counter, Increment seconds
Seconds > 59 ?
No
Yes
Clear seconds, Increment minutes
Minutes > 59 ?
No
Yes
Clear minutes, Increment hours
Hours > 23 ?
No
Yes
Clear hours
Exit
This is the driver that keeps track of time. It handles both 10 MHz clock and 32 kHz clock.
Location: timer_x.inc

8.12 Convert_Temp

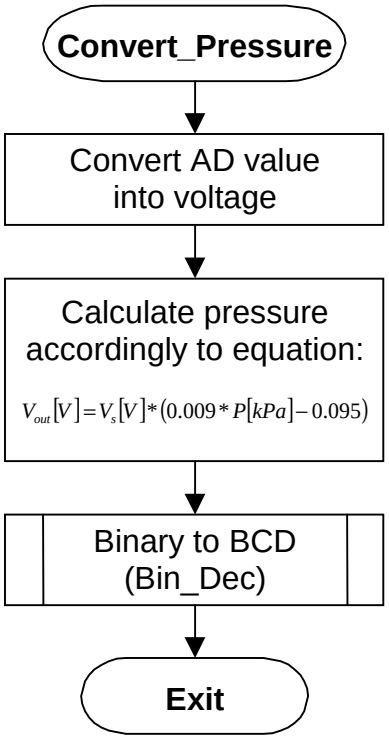


This routine converts value read from AD converter into value in °C units. It is up to user to write code that will handle appropriate accuracy. At the end temperature value is converted into separate digits displayed on LCD. Location: mathpack.inc

8.13 Convert_Humidity



8.14 Convert_Pressure



This routine converts value read from AD converter into value in kPa units using appropriate equation. At the end pressure value is converted into separate digits displayed on LCD. Location: mathpack.inc