MOTOROLA
intelligence everywhere™

digital dna™

Alarm Control Panel Reference Design

Designer Reference Manual

HCS12
Microcontrollers

DRM008/D
Rev. 0, 11/2002

MOTOROLA.COM/SEMICONDUCTORS

Alarm Control Panel Reference Design

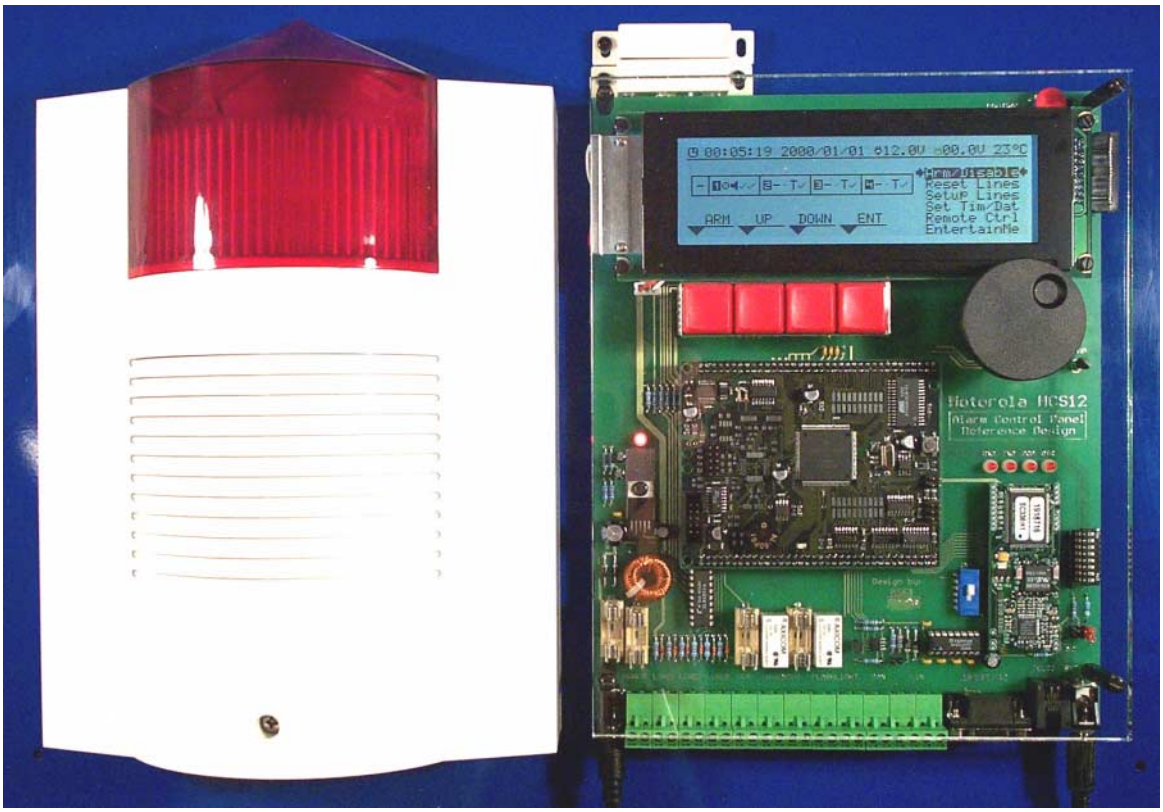
Designer Reference Manual — Rev 0

by: Oliver Thamm c/o
Elektronikladen Mikrocomputer GmbH
W.-Mellies-Str. 88
32758 Detmold, Germany
Email: othamm@hc12web.de

Elektronikladen is a member of Motorola's Design Alliance Program.

More information can be found at the author's project page:
http://hc12web.de/acprd

The Alarm Control Panel can be purchased at:
http://www.elektronikladen.de
http://www.starterGATE.com


00:05:19 2000/01/01 012.0V -00.0V 23°C
ARM
UP
DOWN
ENT
Reset Lines
SetUP Lines
Set Tim/Dat
Remote Ctrl
EntertainMe
Motorola MCS12
Alarm Control Panel
Reference Design

List of Paragraphs

List of Paragraphs

Table of Contents

List of Paragraphs

Table of Contents

List of Tables

List of Figures

Section 1. Introduction

Section 2. Getting Started

Section 3. Hardware Description

Section 4. Software Description

Section 5. Literature

Section 6. Appendix

Glossary

List of Tables

List of Tables

List of Figures

List of Figures

Section 1. Introduction

1.1 Contents

1.2 Scope of the Reference Design

The reference design was developed to demonstrate the capabilities of the HCS12 MCU in a wide range of different applications. This is done using the example of an networkable Alarm Control Panel.

The Alarm Control Panel reference design is based on a MC9S12DP256 MCU. It is a modular design, using several peripherals and features of the HCS12, enabling maximum reuse of the following:

- General HCS12 setup (including PLL and interrupts)
- SCI (for connection to sensor node, PC and modem)
- SPI (for connection with input and output devices)
- ADC (for sensing the alarm lines)
- Output Compare Channel (to generate sound and flash LED)
- Input Capture Channel (to sense the rotary encoder)
- EEPROM (to store configuration data)
- Port Pins (to drive LCD)

Furthermore several important software and hardware technologies are explained:

- Connection of an industrial standard graphical LCD to the HCS12

- Graphical user interface for the HCS12
- Connection to a sensor node using a LIN interface
- Connection of a standard modem to the HCS12

The reference design consists of hardware, software and documentation. The detailed documentation was created in a way that the functions of hardware and software are easy to understand.

1.3 Overview of the Alarm Control Panel

The Alarm Control Panel (Figure 1-1) consists of two main units: the Carrier Board and the S12compact Controller Module (Figure 1-2).

The S12compact Controller Module is a small printed circuit board that holds the microcontroller unit (MCU) as well as standard circuitry for clock and reset generation, power supply and decoupling, plus a number of additional input/output devices.

The Carrier Board provides a number of peripheral devices used to perform the specific functions of the Alarm Control Panel application. The Carrier Board is equipped with connections for alarm sensors, flashlight, siren and power supply. It also has sockets for a graphical LCDisplay, a modem and the S12compact Controller Module.

The main components of the Alarm Control Panel are shown in Figure 1-1.

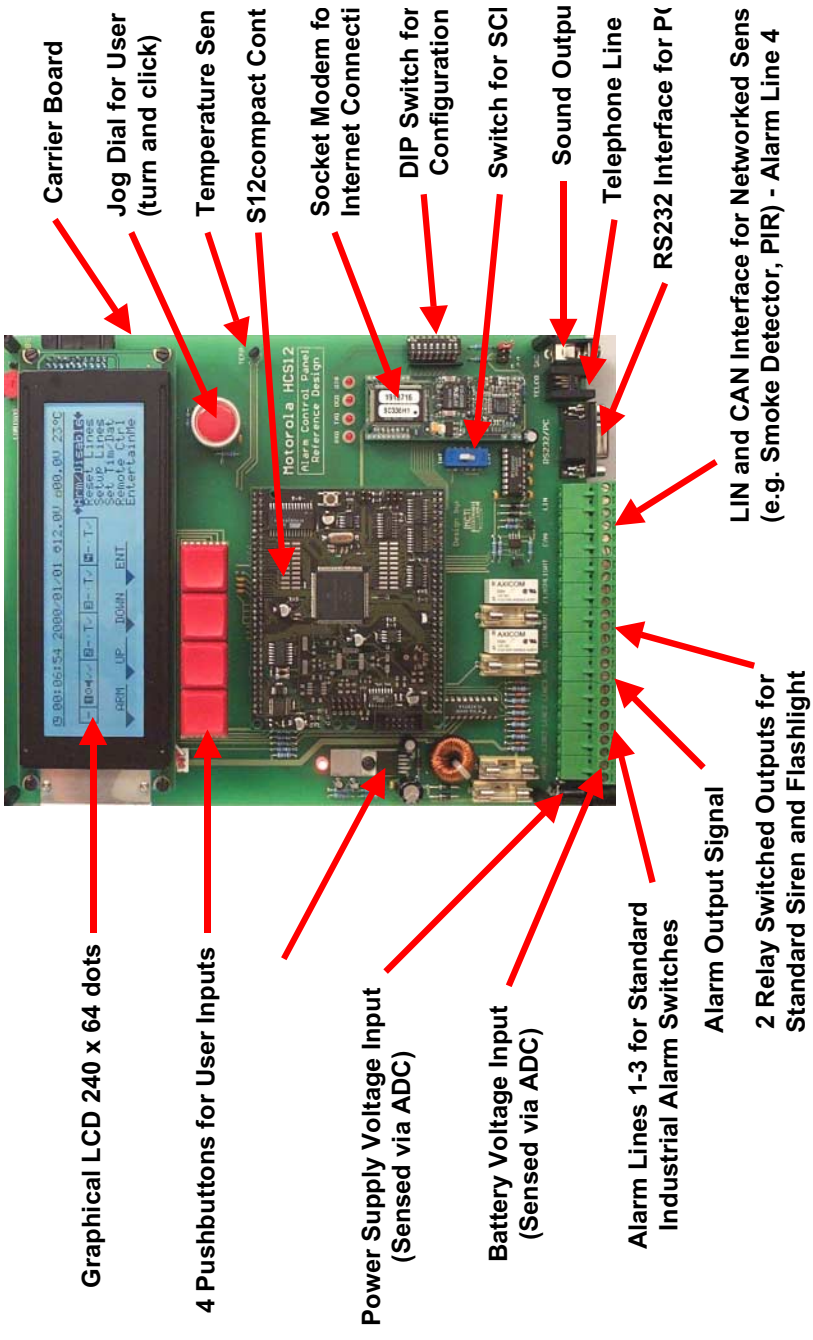


Figure 1-1. Overview of the Alarm Control Panel

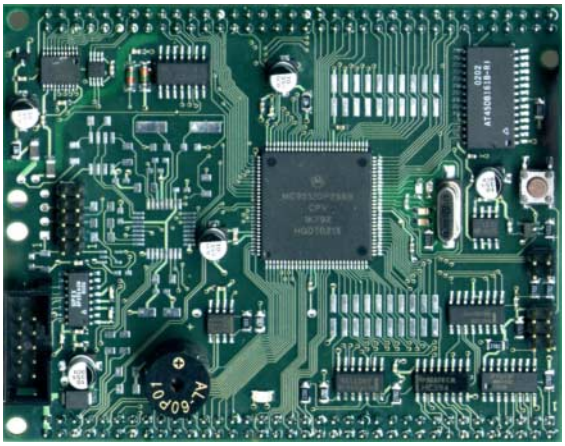


Figure 1-2. S12compact Controller Module

The Alarm Control Panel Reference Design provides a variety of Interfaces (Figure 1-3).

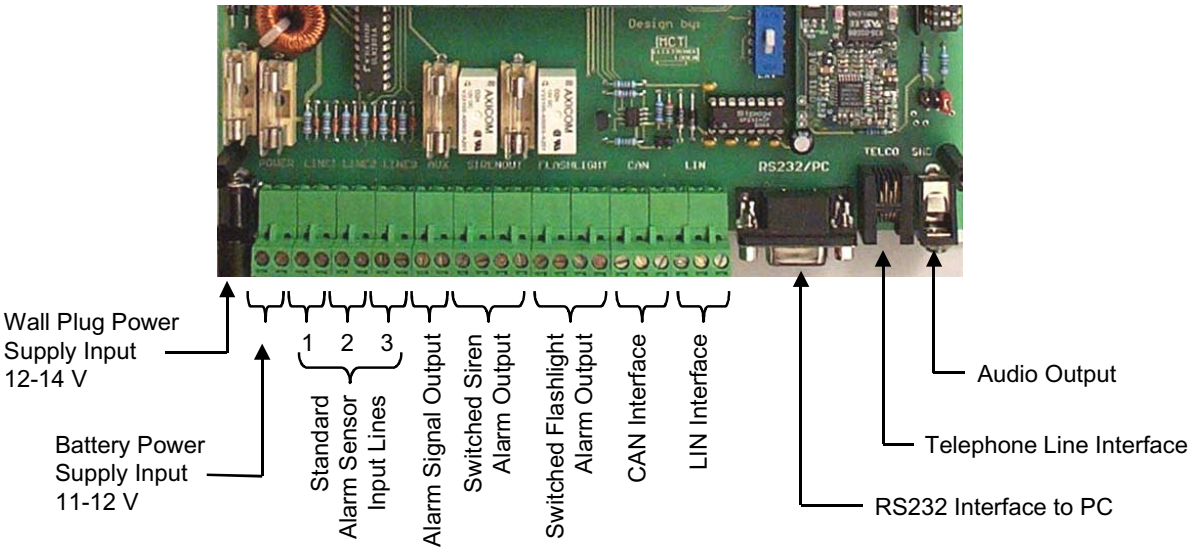


Figure 1-3. Interfaces of the Alarm Control Panel

For a detailed pin description see Section 3.

It is possible to connect industrial standard alarm sensors to the input lines 1-3. These alarm sensors can be connected in a bus structure, so one input line can serve more than one sensor. The input line 4 is used to connect a networked sensor via a LIN (Local Interconnect Network) or a CAN (Controller Area Network) interface. Those networked sensors will be described in detail in a later application note.

1.4 Contents of the Reference Design Kit

The complete Reference Design contains:

- Reference Design Board including Bill of materials (BOM) and PCB layout files (Gerber)
- Fully documented Firmware for the HCS12 in C language, including drivers, function libraries and a sample application.

Introduction

Section 2. Getting Started

2.1 Contents

2.2 Turning On the Alarm Control Panel

In order to switch on the Reference Design the user should connect the power supply to the power supply input. The Alarm Control panel comes with an alarm sensor dummy connected to the sensor Input line 1. This sensor dummy can be used to demonstrate the features of the reference design.

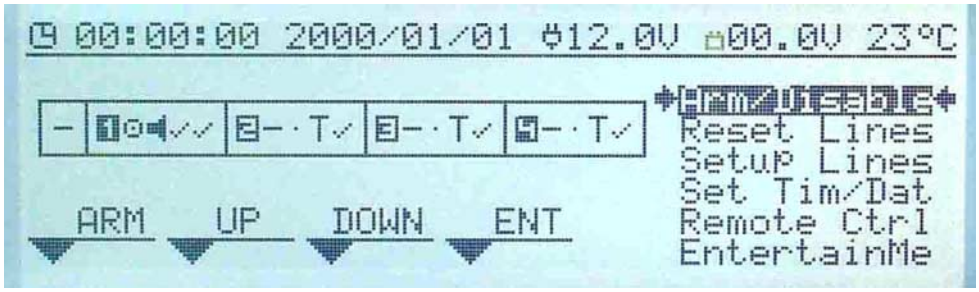


Figure 2-1. LCD after the start of the Alarm Control Panel

After the Reference design has been switched on, the graphical LC Display shows the User Interface:

On the top side of the LC Display there is a Status line which displays the current status: Time, Date, Supply Voltage, Battery Voltage, Temperature.

On the right side there is the Main Menu for the Configuration of the Alarm Control Panel using the rotary encoder (Jog Dial: turn and click).

In the middle of the LCD there is the Alarm Status Display which shows:

- The global Status of the Alarm Control Panel (Armed or disabled)
- The Configuration of each Alarm Line
- the Alarm Status of each Alarm Line

On the bottom the functions of the four push buttons for User Inputs are displayed. Note, that there can be different functions depending on the chosen main menu.

2.3 Main Menu of the Control Panel

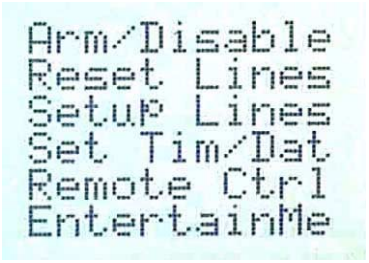


Figure 2-2. Main Menu of the Alarm Control Panel

The Main Menu of the Alarm Control Panel is displayed in Figure 2-2. The selection in the Main Menu can be made by turning the Jog Dial (Rotary Encoder) and pressing it. The following Menu functions are available:

2.3.1 Arm/Disable

This is a function to switch the Alarm On and Off globally.

2.3.2 Reset Lines

This function is used to Reset the Input Lines after an Alarm Event.

2.3.3 Setup Lines

This function configures the four Alarm Lines.

2.3.4 Set Tim/Dat

This function changes the Time and Date.

2.3.5 Remote Control

This is the function to setup the system via a PC or connect to a modem.

In the current Reference Design the modem function is not implemented. It will be described in a later Application Note.

2.3.6 EntertainMe

This function is used to switch the Special Mode for Trade Show Demonstrations on and off.

In Entertainment mode, the Reference Design LED blinks and a melody is played.

2.4 Arm the Control Panel and Disable the Alarm

In order to arm the Alarm Control Panel, go to “Arm/Disable” and click the Jog. The status symbol in the Display changes from a Minus to a Check Mark (Figure 2-3).

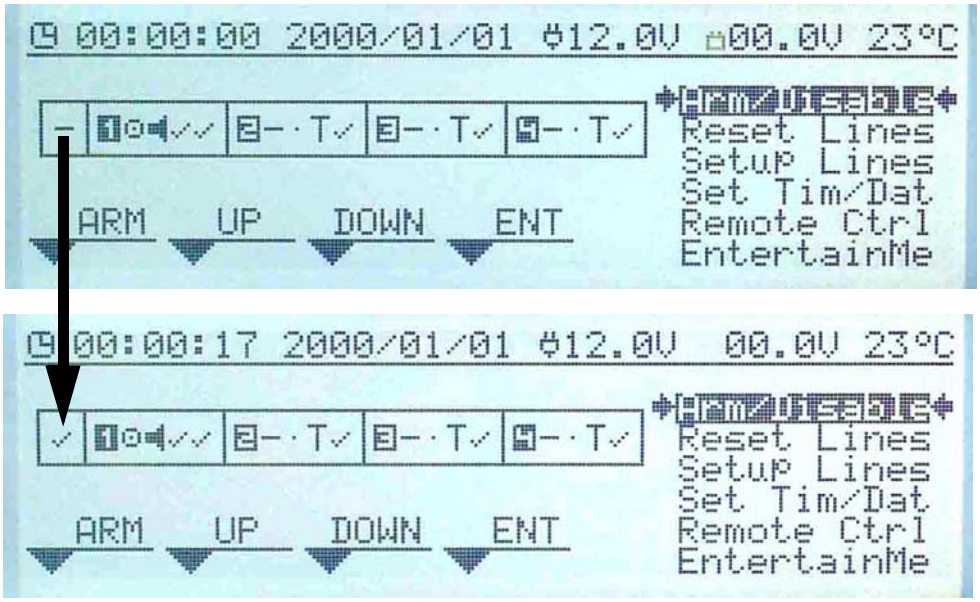


Figure 2-3. Arming the Alarm Control Panel

When an Alarm occurs, the following happens:

- The Check Mark symbol changes to a Bell
- The blue LED on the Alarm Control Panel flashes
- If enabled, a siren tone is sounded by the loudspeaker
- The output relays switch on the siren and flashlight outputs for external sirens and flashlights.

To disable the Panel or switch the Alarm off, click the Jog again.

Alternatively, the push buttons can be used. The actual functions of the push buttons are displayed on the LC Display.

In a real Alarm Control Panel application, the function of disabling the Alarm would be secured with a PIN code, so a possible intruder cannot switch off the Alarm. This function is not implemented in the reference design.

2.5 Configuration of the Alarm Lines

In order to configure the Alarm Lines, select “Setup Lines” by turning the Jog Dial and pressing it.

Then select the line to be configured by clicking the Jog Dial. The line that is currently configured will be indicated by an arrow pointing to the line.

Choose the configuration of the selected line by turning the Jog Dial. Each Alarm Line can have one of five configurations.

When you have configured one line, select the next Line by pressing the Jog Dial.

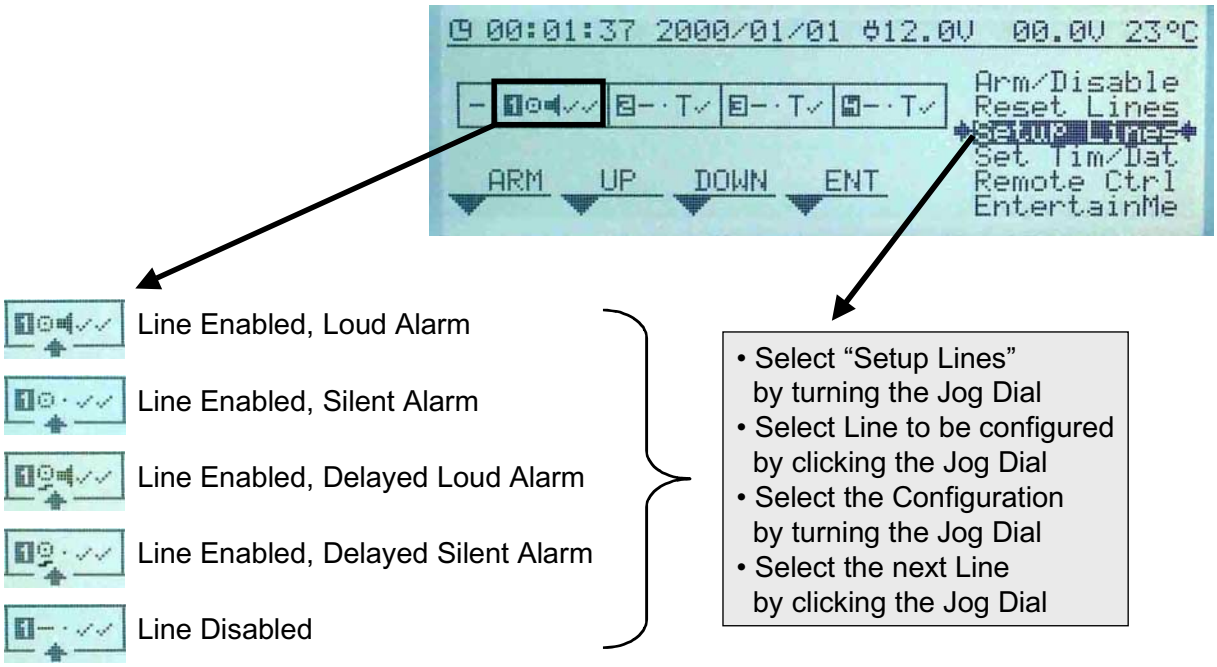


Figure 2-4. Configuration of the Alarm Lines

Each Alarm Line can have the five following Configurations:

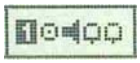
- Line Enabled, Loud Alarm: This is the default configuration of the Alarm Line 1 after a Reset.

- Line Enabled, Silent Alarm: You would choose this configuration if a possible burglar should not notice immediately that an alarm has been released.
- Line Enabled, Delayed Loud Alarm: This option delays the arming of the Alarm Control Panel by 30 seconds. You would choose this configuration if you wish to leave your house.
- Line Enabled, Delayed Silent Alarm: This configuration is a combination of the two above.
- Line Disabled: This configuration option switches off the line. This is the default configuration of Alarm Lines 2-4 after a Reset.

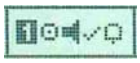
2.6 Alarm Status of the Lines

The Alarm Status of all lines is shown in figure 2.5.

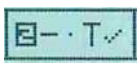
Line enabled:
If an Event on Line 1 occurs, it will trigger an Alarm.
A delayed Line would release the Alarm after 30 seconds.



Alarm Event:
An Alarm Event is sensed on Line 1.



Alarm Latched:
Even if the Alarm Event disappears, it is still latched.
To switch off the Alarm, the Line has do be reset.



Alarm Line Tampered:
It has been sensed that the Alarm Line does not work properly.
A possible reason can be the attempt to manipulate the Alarm Line (for example, to shorten or to cut the wires).
Since the standard demo version of the Alarm Control Panel does not have any sensors on the Alarm Lines 2-4, the display will show that these Alarm Lines have been tampered with.

Figure 2-5. Alarm Status of the lines

If the Line is enabled and an Event on that Line occurs, it will trigger an Alarm. A delayed Line would release the Alarm only after 30 seconds.

When an Alarm Event is sensed, the Status display will show two bells ringing. The Alarm Event is also latched, that means:

Even if the Alarm Event disappears, we still know that there was an Alarm. To switch it off, the Line has do be reset manually.

An Alarm Line can be tampered with, that means it has been sensed that the Alarm Line does not work properly. A possible reason can be the attempt to manipulate the Alarm Line (for example, to shorten or to cut the wires).

Since the standard demo version of the Alarm Control Panel does not have any sensors on the Alarm Lines 2-4, the display will show that these Alarm Lines have been tampered with. However, these Lines are disabled, so they do not trigger an Alarm.

2.7 Configuration of the Alarm Control Panel via PC

It is possible to configure the Alarm Control Panel via a standard terminal on a Personal Computer. For this purpose, the terminal should be connected to the RS232 interface of the Alarm Control Panel.

The switch for the SCI configuration should be at the position for RS232 communication.

The other position of the switch should be used for modem communication. The communication via modem is not described in this reference design but will be the subject of a later Application Note.

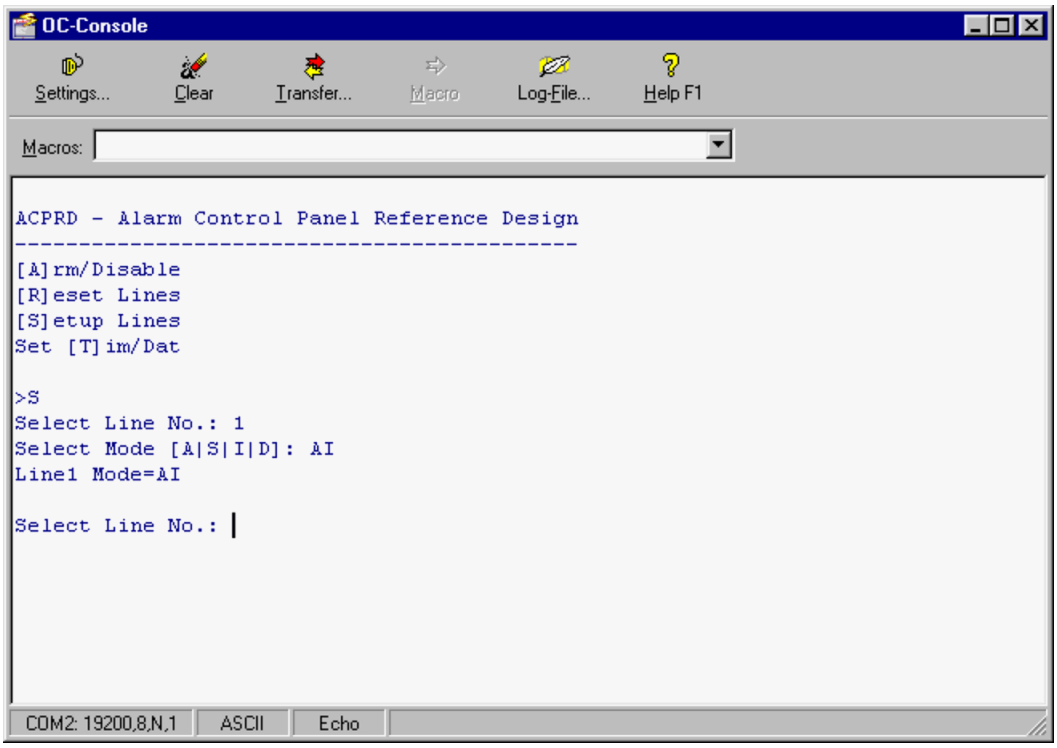


Figure 2-6. Configuration of the Alarm Control Panel via a standard terminal

Section 3. Hardware Description

3.1 Contents

3.2 Hardware Overview

The Alarm Control Panel Reference Design (ACPRD) hardware consists of two main units: the ACPRD Carrier Board and the S12compact Controller Module.

The ACPRD Carrier Board provides a number of peripheral devices, which are used to perform the specific functions of the Alarm Control Panel application. The Carrier Board is equipped with connections for alarm sensors, flashlight, siren and power supply. It also has sockets for a graphical LCDisplay, a modem and the S12compact Controller Module.

The S12compact Controller Module is a small printed circuit board that holds the MC9S12DP256 microcontroller unit (MCU) as well as standard circuitry for clock and reset generation, power supply and decoupling, plus a number of additional input/output devices. The S12compact Controller Module can optionally be equipped with a memory expansion, which is mainly targeted as a debugging aid.

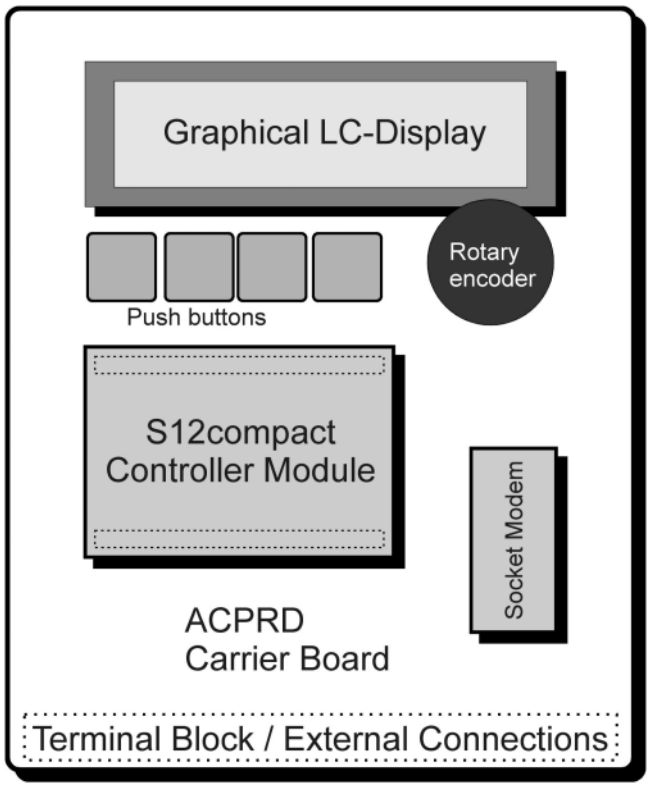


Figure 3-1. Hardware Overview

3.3 The MC9S12DP256 Microcontroller

The MC9S12DP256 microcontroller unit (MCU) contains a 16-bit central processing unit (HCS12 CPU), 256K bytes of Flash EEPROM, 12K bytes of RAM, 4K bytes of EEPROM and a large number of standard on-chip peripherals, such as SCI, SPI, CAN, IIC, Timer, PWM, ADC and Input-/Output-Channels. The MC9S12DP256 has full 16-bit data paths throughout. The inclusion of a PLL circuit allows power consumption and performance to be adjusted to suit operational requirements.

The main features of the MC9S12DP256 are listed below[1]:

- 16-bit HCS12 CPU
- Operation up to 50MHz Clock / 25MHz Bus Speed

- CRG: low current oscillator, PLL, reset, clocks, COP watchdog, real time interrupt, clock monitor
- MEBI: Multiplexed External Bus Interface
- I/O: 8-bit and 4-bit ports with interrupt functionality
- FLASH: 256K bytes
- EEPROM: 4K bytes
- RAM: 12K bytes
- ATD: two 8-channel Analog-to-Digital Converters with 10-bit resolution
- CAN: five 1M bit per second, CAN 2.0 A, B software compatible modules
- ECT: Enhanced Capture Timer with 16-bit main counter, 8 programmable input capture or output compare channels and two 8-bit or one 16-bit pulse accumulators
- PWM: eight channels, programmable period and duty cycle, 8-bit 8-channel or 16-bit 4-channel, fast emergency shutdown input
- SCI: two asynchronous Serial Communications Interfaces
- SPI: three Synchronous Serial Peripheral Interface
- BDLC: Byte Data Link Controller
- IIC: Inter-IC Bus, compatible with I2C Bus standard, multi-master operation
- 112-Pin LQFP package
- Supply: 5V operating voltage, internal voltage regulator for 2.5V MCU core voltage, I/O lines with 5V input and drive capability
- Development support: single-wire background debug™ mode (BDM), on-chip hardware breakpoints

Figure 3-2 shows the block diagram of the MC9S12DP256.

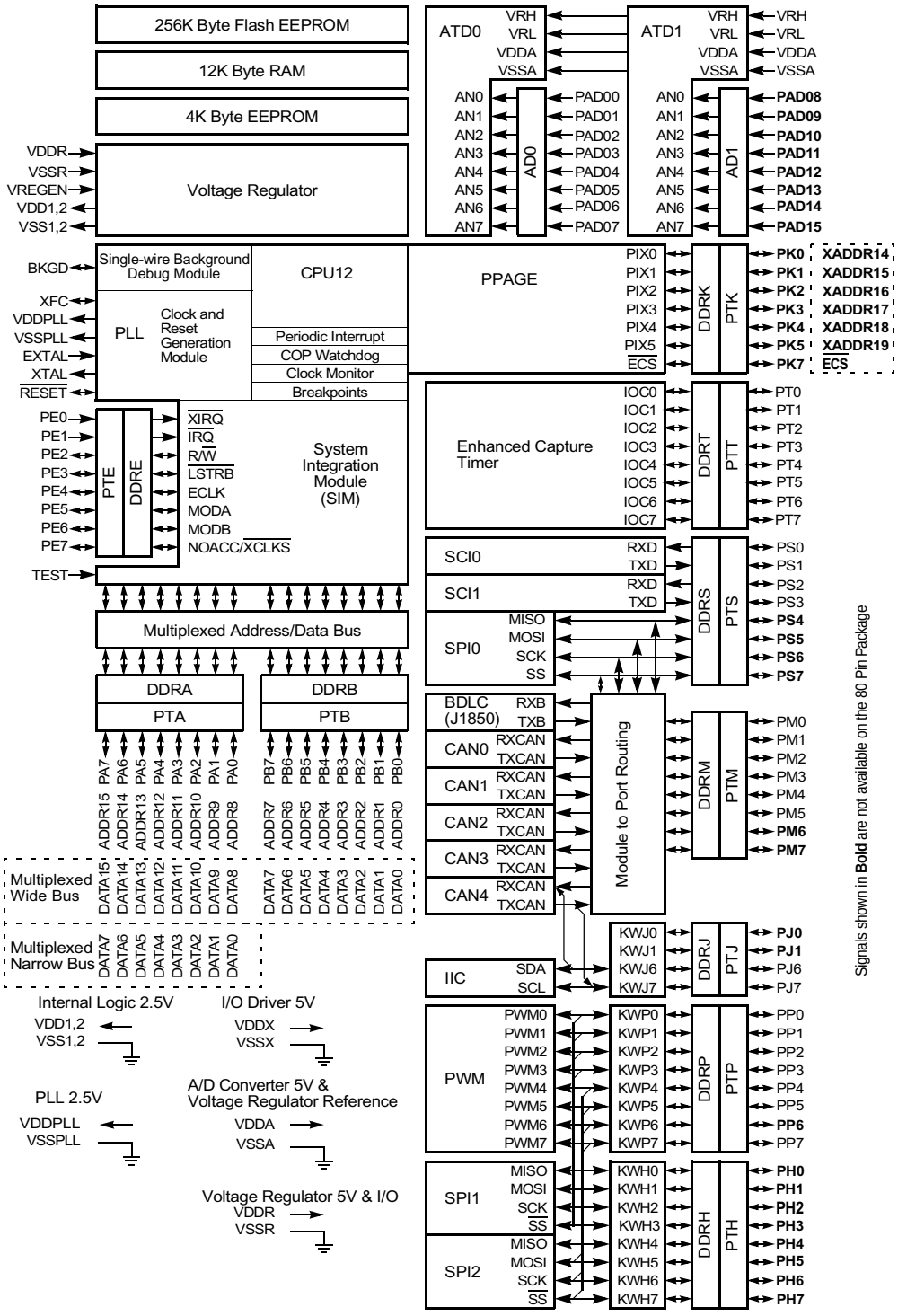


Figure 3-2. Block diagram of the MC9S12DP256 microcontroller

3.4 S12compact Controller Module

3.4.1 Controller Core, Power Supply

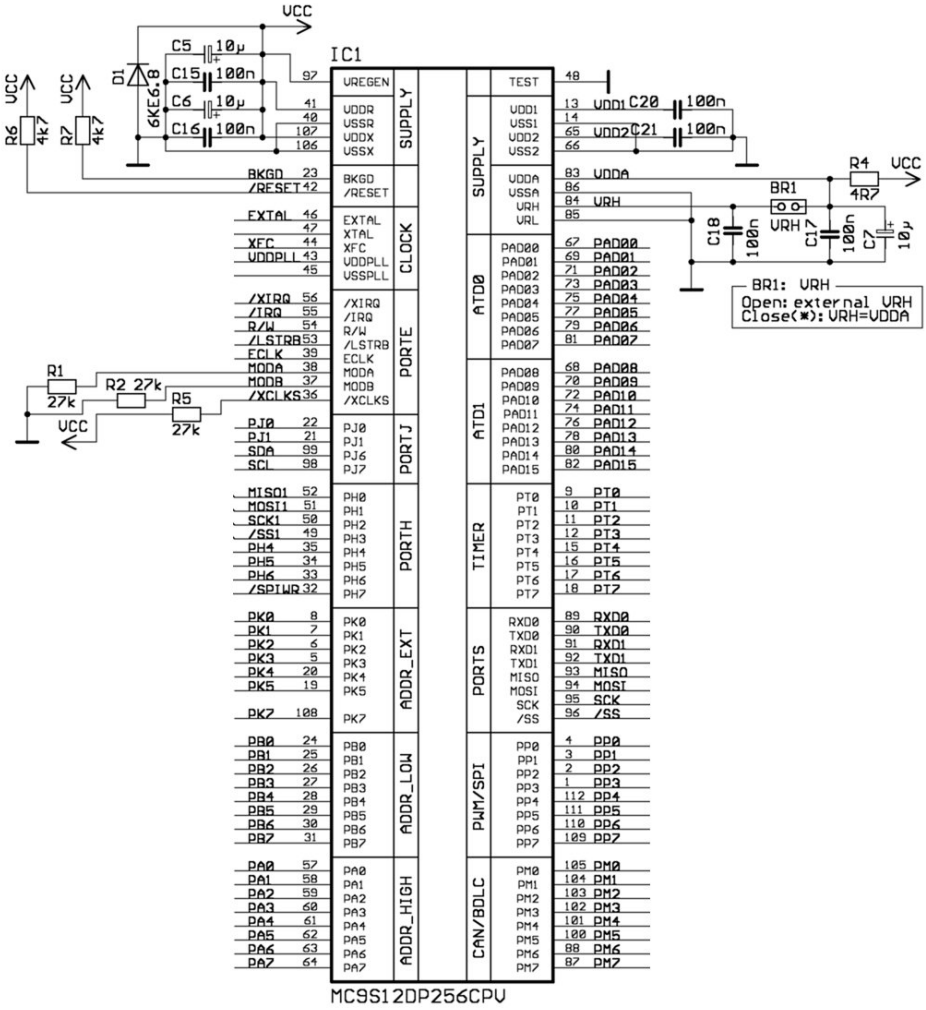


Figure 3-3. Power Supply of the MC9S12DP256 microcontroller

The nominal operating voltage of the MC9S12DP256 is 5V. The MCU (IC1) has three supply pin pairs: VDDR/VSSR, VDDX/VSSX and VDDA/VSSA. Internally, the MCU uses a core voltage of only 2.5V. The necessary voltage regulator is already included in the chip, as well as 5V I/O-buffers on all input/output pins. Therefore, the MCU behaves like a 5V device from an external point of view. There is just one exception, the

signals for oscillator and PLL are based on the core voltage and must not be driven by 5V levels. A high level on the pin VREGEN is needed to enable the internal voltage regulator.

The three terminal pairs mentioned above must be decoupled carefully. A ceramic capacitor of at least 100nF should be connected directly at each pair (C15, C16, C17). It is recommended to add a 10µF (electrolytic or tantalum) capacitor per node, especially if some MCU port pins are loaded heavily (C5, C6, C7). Special care must be taken with VDDA, since this is the reference point (VDDA/2) for the internal voltage regulator.

The internal core voltage appears at the pin pairs VDD1/VSS1, VDD2/VSS2 and VDDPLL/VSSPLL, which also have to be decoupled (C19, C20, C21). A static current draw from these terminals is not allowed. This is especially true for VDDPLL, which serves as the reference point for the external PLL loop filter combination (R3, C3, C4).

There are two MCU pins (VRH/VRL) to define the upper and lower voltage limits for the internal analog to digital (ATD) converter. While VRL is grounded, VRH is usually tied to VDDA. C18 is used for decoupling. VRH can be supplied externally after opening the solder bridge BR1. This can be useful if the main supply is not in the desired tolerance band or if the ATD should work with a reference value lower than 5V. VRH must not exceed VDDA, regardless of the selected supply mode.

The TEST pin is used for factory testing only, in an application circuit this pin always has to be grounded.

3.4.2 Reset Generation

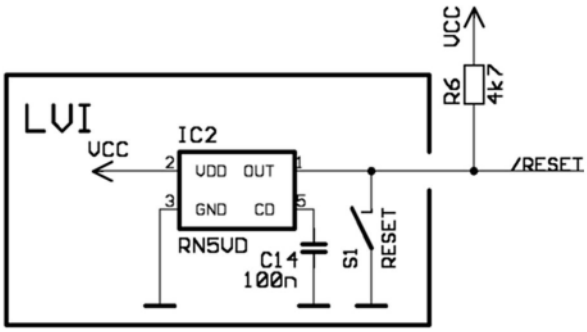


Figure 3-4. Reset generation of the MC9S12DP256 microcontroller

$\overline{\text{RESET}}$ is the MCU's active low bidirectional reset pin. As an input it initializes the MCU asynchronously to a known start-up state. As an open-drain output it indicates that a system reset (internal to MCU) has been triggered. The HCS12 MCU already contains on-chip reset generation circuitry including power-on reset, COP watchdog timer and clock monitor. It is, however, necessary to add an external Low Voltage Inhibit (LVI) circuit, also referred to as “reset controller”. The task of this reset controller is to issue a stable reset condition if the power supply falls below the level required for proper MCU operation.

To prevent collisions with the bidirectional $\overline{\text{RESET}}$ pin of the MCU, the LVI circuit IC2 has an open-drain output. In the inactive state it is pulled-up high by the resistor R6. The detector threshold of IC2 is typically 4.6V, which is slightly higher than the required minimum MCU operating voltage of 4.5V.

Furthermore, IC2 is capable of stretching the reset output to filter out short pulses on the power supply effectively. The duration of that delay can be selected using the capacitor C14. A value of 100nF results in a delay of approx. 50–80ms.

It is important to note, that this delay will only be applied during a power cycle event. IC2 will not stretch pulses coming from the MCU's internal reset sources. This is important, since otherwise the MCU would not be able to detect the source of a reset. This would finally lead to a wrong reset vector fetch and could crash the whole system. Please be aware,

that also a capacitor on the reset line would cause the same fatal effect, therefore external circuitry connected to the $\overline{\text{RESET}}$ pin of a HC12/HCS12 MCU should never include a large capacitance!

3.4.3 Clock Generation and PLL

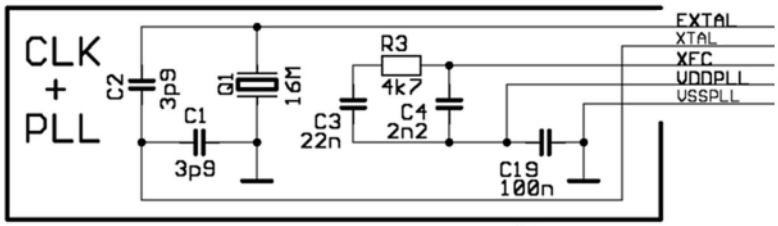


Figure 3-5. Clock generation of the MC9S12DP256 microcontroller

The on-chip oscillator of the MC9S12DP256 can generate the primary clock (OSCCLK) using a quartz crystal (Q1) connected between the pins EXTAL and XTAL. The allowed frequency range is 0.5–16MHz. As usual, two load capacitors are part of the oscillator circuit (C1, C2). However, this circuit is modified compared to the standard Pierce oscillator that was used for the HC11 or most HC12 derivatives.

The MC9S12DP256 uses a Colpitts oscillator with translated ground scheme. The main advantage is a very low current consumption, although the component selection is more critical. The Reference Design circuit uses a 16 MHz automotive quartz from NDK together with two load capacitors of only 3.9pF. Furthermore, special care was taken for the PCB design to introduce as little stray capacitance as possible in respect to XTAL and XTAL.

With an OSCCLK of 16MHz, the internal bus speed (ECLK) becomes 8MHz by default. To realize higher bus clock rates, the PLL has to be engaged. The MC9S12DP256 can be operated with a bus speed of up to 25MHz, though most designs use 24MHz because this value is a better basis to generate a wide range of SCI baud rates.

A passive external loop filter must be placed on the XFC pin. The filter (R3, C3, C4) is a second-order, low-pass filter to eliminate the VCO input

ripple. The value of the external filter network and the reference frequency determines the speed of the corrections and the stability of the PLL. If PLL usage is not required, the XFC pin can be tied to VDDPLL.

The choice of filter component values is always a compromise over lock time and stability of the loop. 5 to 10kHz loop bandwidth and a damping factor of 0.9 are a good starting point for the calculations. With a quartz frequency of 16MHz and a desired bus clock of 24MHz, a possible choice is R3 = 4.7k and C3 = 22nF. C4 should be approximately (1/20–1/10) x C3, e.g. 2.2nF in our case. Please refer to the chapter “XFC Component Selection” in the MC9S12DP256B Device User Guide [1] for details on how to calculate loop filter values for other system configurations.

An alternative, external clock source can be used for the MC9S12DP256 if the internal oscillator and PLL are disabled by applying a low level to the $\overline{\text{XCLKS}}$ pin during reset. Since this option is not used on the S12compact Controller Module, R5 is used to pull $\overline{\text{XCLKS}}$ high. Please note, that other HCS12 derivatives will have different features associated with the $\overline{\text{XCLKS}}$ pin.

3.4.4 Operating Modes, BDM Support

Three pins of the HCS12 are used to select the MCU operating mode: MODA, MODB and BKGD (=MODC). While MODA and MODB are pulled low (R1, R2) to select Single Chip Mode, BKGD is pulled high (R7) by default. As a consequence, the MCU will start in Normal Single Chip Mode, which is the most common operating mode for application code running on the HCS12.

The HCS12 operating mode used for download and debugging is called Background Debug Mode (BDM). BDM is active immediately out of reset if the mode pins MODA/MODB/BKGD are configured for Special Single Chip Mode. This is done by pulling the BKGD pin low during reset, while MODA and MODB are also pulled-down.

Because only the BKGD level is different for the two modes, it is quite easy to change over. However, there is no need to switch the BKGD line manually via a jumper or solder bridge because this can be done by a

BDM-Pod attached to connector X1A. Such a BDM-Pod is required for BDM-based download and/or debugging anyway, so it can handle this task automatically, usually controlled by a PC-based debugging program.

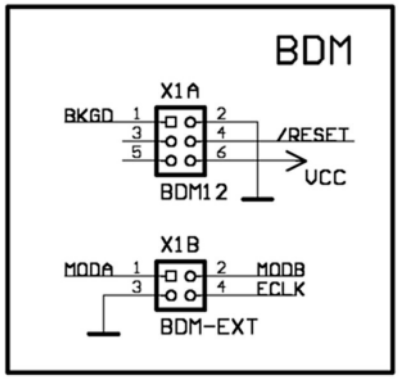


Figure 3-6. BDM interface

The 6-pin header X1A uses the suggested standard BDM12 connector layout. Connector X1B carries additional MCU signals, which are normally not needed for BDM12 debugging. Some debuggers, however, provide additional features, which rely on the presence of these supplemental signals.

3.4.5 SPI Subsystem

While the MC9S12DP256 provides a large number of versatile general-purpose I/O-ports, it may be desirable for some applications to keep a number of ports unused. This is especially the case if the external bus interface is intended to be used.

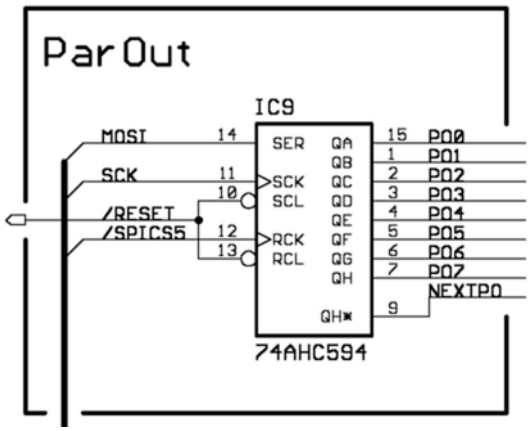


Figure 3-7. SPI Parallel Output

A preferred way to provide additional input or output ports is to use one of the Serial Peripheral Interface (SPI) modules of the MC9S12DP256. It is quite easy to create eight additional binary outputs by adding a shift register like IC9. In addition to the SPI signals, MOSI and SCK, only a chip select signal ($\overline{\text{SPICS5}}$) is needed to operate IC9. This shift register has an asynchronous reset input for both the shift register and the output latch, so all output lines PO0–PO7 will drive L-level out of reset automatically.

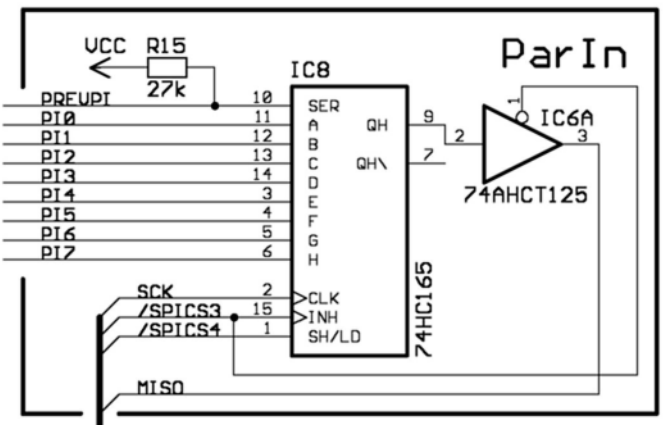


Figure 3-8. SPI Parallel Input

Using the same SPI port, IC8 provides eight additional input pins. Two chip selects ($\overline{\text{SPICS3}}$ and $\overline{\text{SPICS4}}$) are needed to latch the input

information coming from PI0–PI7 and to shift out data via MISO respectively. IC6A is needed to decouple the push/pull-output QH of IC8 from MISO, in order to allow multiple slaves on the same SPI port.

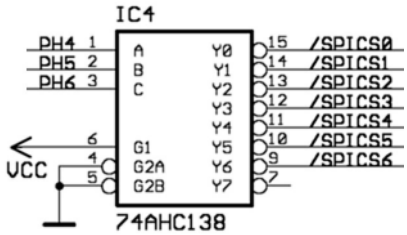


Figure 3-9. SPI Chip Select decoding

The SPI chip select signals are derived from the MCU port pins PH[4–6]. The decoder chip IC4 activates one of it's eight low-active outputs depending on the Port H pattern. This provides an economical way to provide up to eight chip selects with a small number of MCU resources.

3.4.6 Physical Interface Drivers

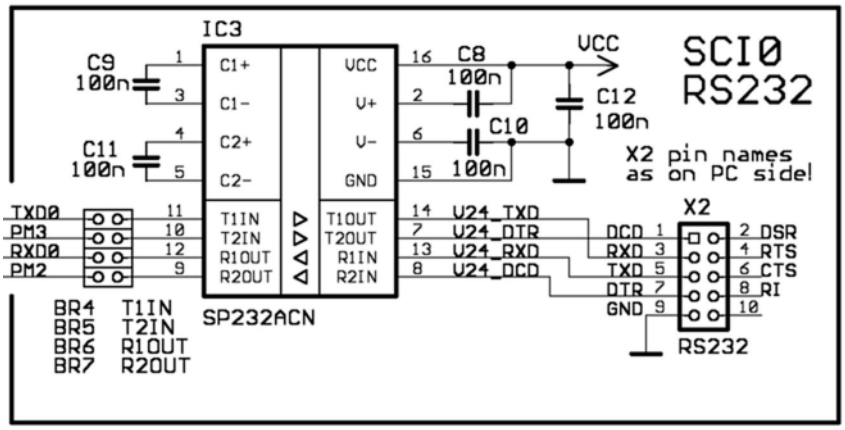


Figure 3-10. Serial Communication Interface of the MC9S12DP256

IC3 is an industry standard RS232 line transceiver circuit. It is connected to SCI0, which is the first asynchronous serial communications channel of the MC9S12DP256. In addition to the receive and transmit lines

(RXD0, TXD0), two general purpose I/Os (PM2, PM3) can be used as hardware handshake lines. If the RS232 transceiver IC3 is not needed, it can easily be uncoupled by opening the four solder bridges BR4–BR7.

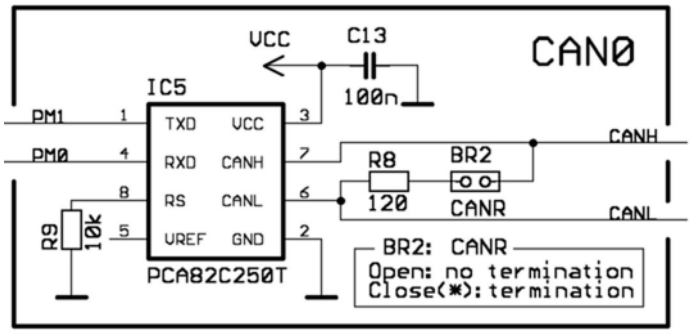


Figure 3-11. CAN Interface of the MC9S12DP256

IC5 serves as a CAN physical bus interface, connected to the first CAN module (CAN0) of the MC9S12DP256. It is a high-speed interface chip commonly used in industry applications. R9 determines the slope control setting. R8 is a termination resistor, required if the S12compact is the last node in a CAN bus chain. BR2 should be closed in this case, otherwise opened.

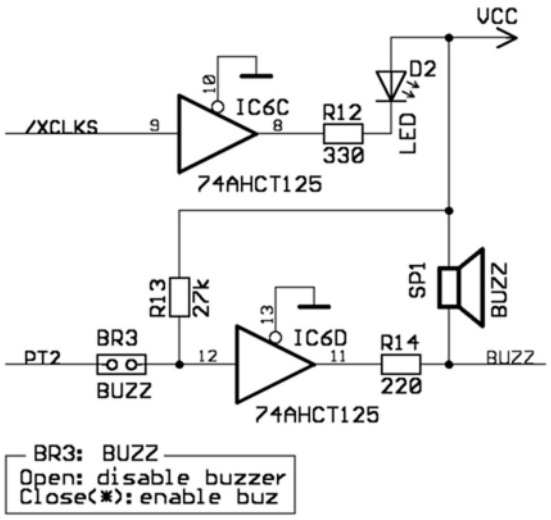


Figure 3-12. Buzzer Interface

There is a LED on the S12compact Controller Module which is driven by buffer IC6C. The signal XCLKS is used to select one of the clock options for the MCU during reset. It can be freely used as I/O-signal after reset.

IC6D is another buffer gate, it is connected to port pin PT2 and drives a small sound transducer (buzzer). BR3 must be closed in order to enable the buzzer function.

3.4.7 External Bus Interface

The MCU ports A, B, K and E (partly) are related to the Multiplexed External Bus Interface (MEBI). All bus signals are accessible via two header connectors (X7, X8).

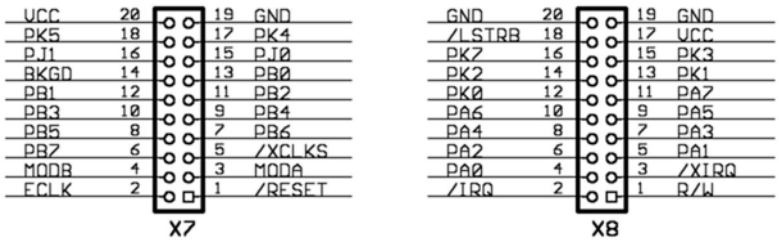


Figure 3-13. Connectors of the Multiplexed External Bus Interface

A small memory expansion PCB can be plugged onto these two connectors, which is especially useful for debugging purposes (Flash emulation).

If the Multiplexed External Bus Interface is not used (default) the ports A, B, K can be used as general-purpose I/O-ports.

3.4.8 Circuit Options

If desired, the S12compact Controller Module can be equipped with additional peripheral function blocks:

- RTC with clock/calendar
- 16 bit ADC with 8 channels

- 16 bit DAC with 2 channels
- 2 MByte Serial Data Flash
- Full-Speed USB2.0 interface

None of these circuit options is currently necessary for the Alarm Control Panel Reference Design. However, they are provided to demonstrate the simplicity of adding various peripheral hardware blocks to the HCS12 using simple serial connections via SPI or SCI.

3.5 The ACPRD Carrier Board

3.5.1 S12compact Module Socket

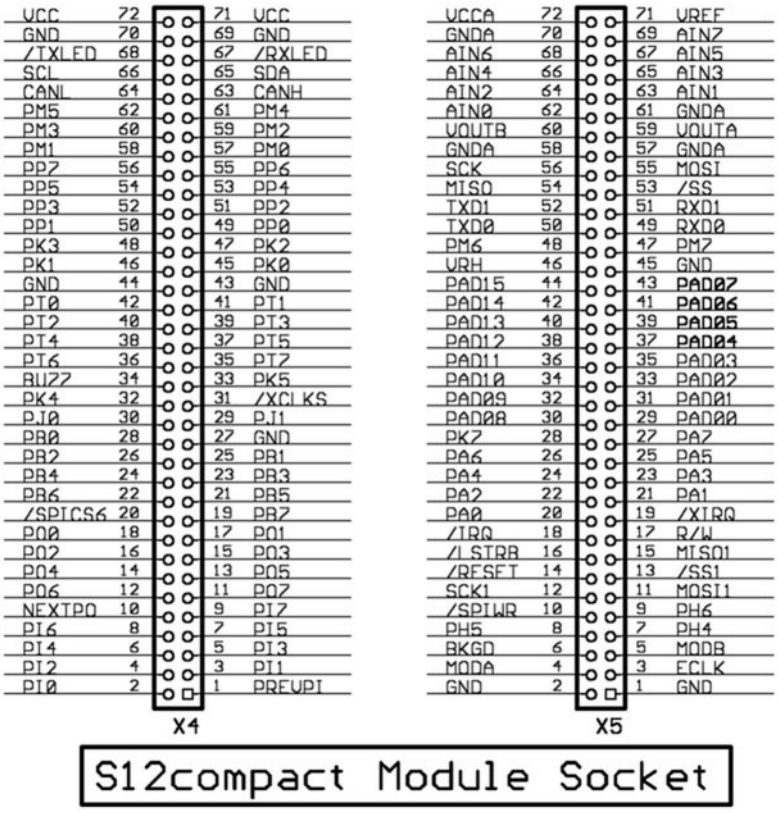


Figure 3-14. Connectors of the S12compact Module

The S12compact Controller Module is equipped with a double row of header connectors along both edges. On the ACPRD Carrier Board, corresponding socket strips (X4, X5) are available to easily connect both boards.

3.5.2 Voltage Regulator, Power LED

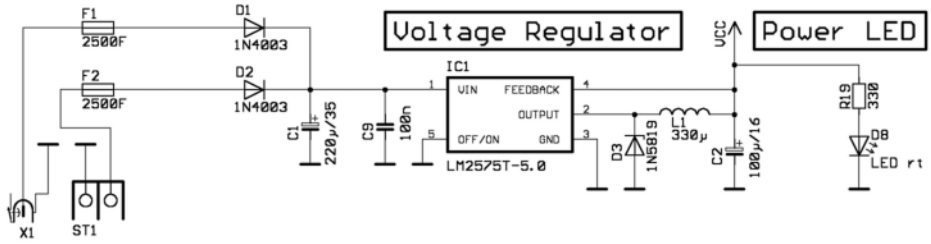


Figure 3-15. Voltage Regulator of the Alarm Control Panel

IC1 is a step-down switching voltage regulator, capable of driving a 1A load with very good line and load regulation. It has a fixed output voltage of 5V and requires only four external components. Compared to traditional, linear (3-terminal) regulators, the switched device offers high efficiency, therefore the heat sink can be very small.

The presence of 5V power output is indicated by LED D8. The input voltage may be as high as 20VDC, but the nominal value is 12–16VDC for the main power input at X1 and 10–12VDC for the backup battery input at ST1. Under normal conditions, the main voltage should always be higher than the battery voltage.

3.5.3 ATD Signal Conditioning, Power Line and Battery Sensing, Alarm Line Sensing

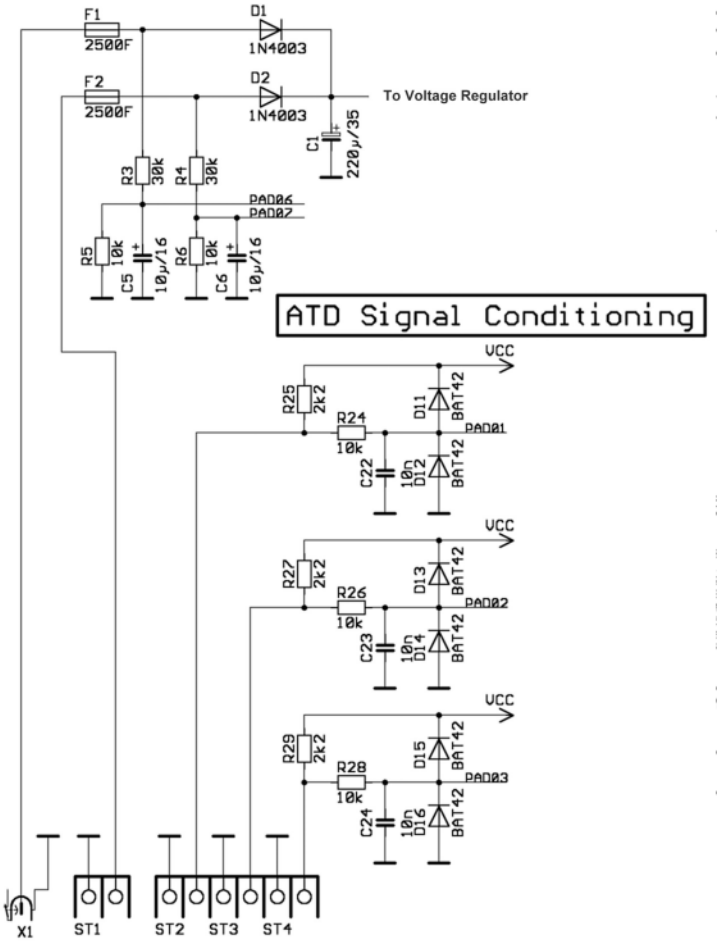


Figure 3-16. Analog to Digital Converter Signal Conditioning

The voltage level of main power input and battery backup input is measured by PAD06 and PAD07, respectively. These two signals are inputs of the HCS12 on-chip analog to digital (ATD) converter. The input voltage levels are decreased to one fourth using the resistor divider R3/R5 and R4/R6 in order to fit the input voltage range of the ATD which is 0–5V. C5 and C6 are filter capacitors to remove ripple from the unregulated input.

The ACPRD has three independent Alarm Line channels. Each Alarm line consists of a resistor divider, which is partly formed from the exterior

alarm loop and partly from the components on the ACPRD Carrier Board. For the first Alarm Line channel, R24, C22, D11 and D12 provide some (basic) noise and overvoltage protection. R25 is the upper resistor of the divider chain. Figure 3-17 shows the principle of operation of such an alarm line.

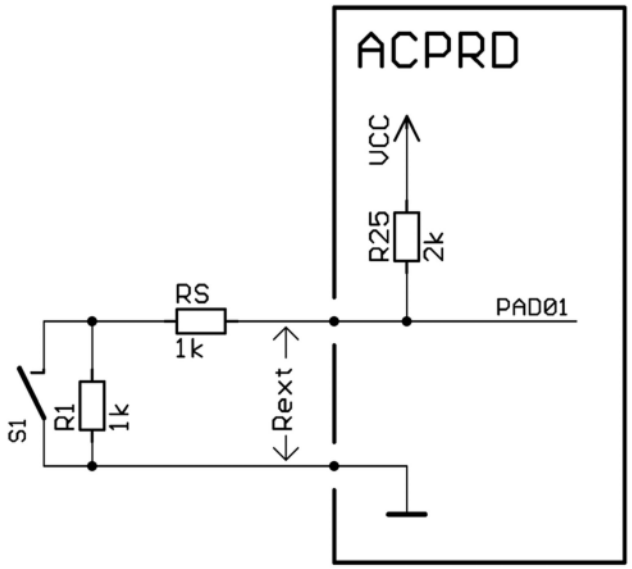


Figure 3-17. Alarm line scheme

The alarm contact S1 is normally closed. If this is the case, only resistor Rs contributes to the external resistor value Rext. S1 will open if an alarm situation is detected. As a result, Rs and R1 together form the lower part (Rext) of the resistor divider chain. If the alarm line is tampered with, either the alarm loop is shortened (Rext=0) or it is cut off (Rext=00). All in all, there are four possible ATD input values, as the following table summarizes:

Status	Rext	R2x	PAD0x
okay	1k	2k	1/3 VCC
alarm	2k	2k	1/2 VCC
short	0k	2k	0
cut	-	2k	VCC

Table 3-1. Alarm Line input values

This kind of alarm contact configuration is known as NC (normally closed) alarm loop with EOL (end-of-line) termination.

3.5.4 Temperature Sensor

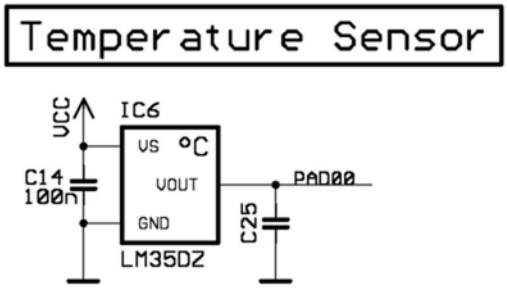


Figure 3-18. Temperature Sensor

IC6 is a precision Celsius temperature sensor which delivers an output voltage linearly proportional to the ambient temperature. The scale factor is 10mV/°C, therefore 0°C delivers an output of 0V and 100°C delivers 1.00V nominally. The output if IC6 is connected to the ATD channel PAD00.

3.5.5 RS232 Transceiver

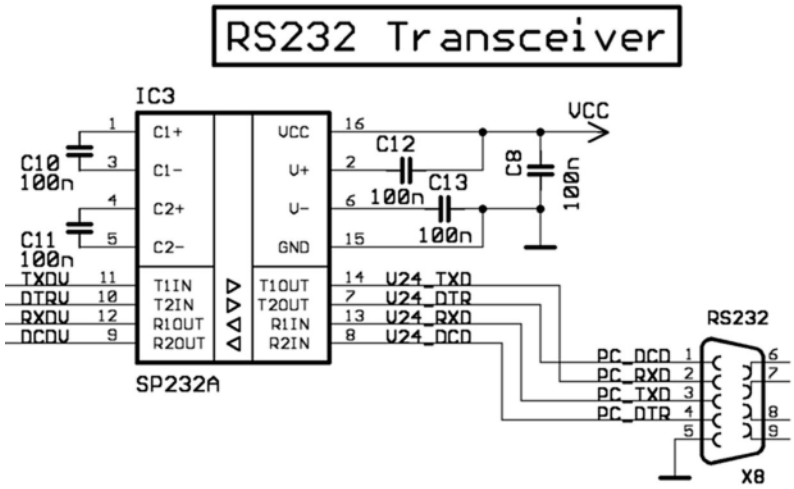


Figure 3-19. RS232 transceiver on SCI0

IC3 is an industry standard RS232 line transceiver circuit. It is connected to SCI0, which is the first asynchronous serial communications channel of the MC9S12DP256. In addition to the receive and transmit lines (RXD0, TXD0), two general purpose I/Os (PM2, PM3) can be used as hardware handshake lines.

IC3 on the ACPRD Carrier Port duplicates the RS232 transceiver circuit on the S12compact Controller Module. This was implemented in order to be able to switch between two modes: either the TTL-level RS232-signals are used to feed the level shifter IC3, or they can be used to drive the socket modem IC2, which can be directly connected to a telephone line. In both cases, the solder bridges BR4–BR7 on the S12compact Module must be opened.

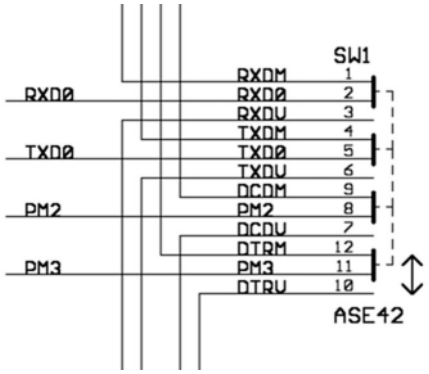


Figure 3-20. Switch for the SCO0 mode

The slide switch SW1 can be used to switch between the two modes on SCI0.

3.5.6 Socket Modem

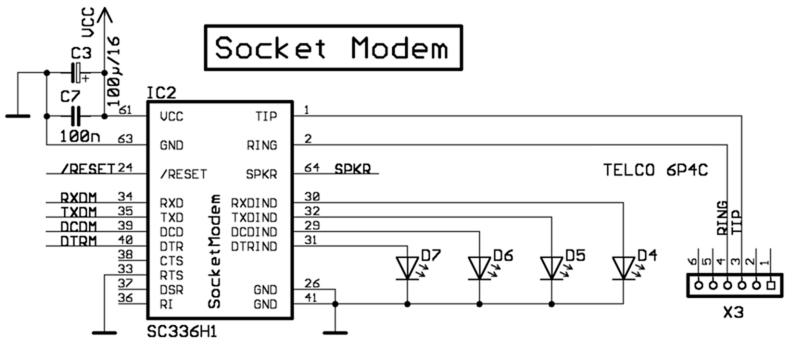


Figure 3-21. Socket Modem on SCI0

To provide additional communication capabilities, the ACPRD can be equipped with a Socket Modem. A Socket Modem is small OEM module with pins along the edges of the module, comparable with a 64-pin DIP-IC. IC2 on the ACPRD Carrier Board is actually socketed, so different types of Socket Modems can be plugged in.

For the ACP Reference Design a versatile V.34 Socket Modem from Conexant/Multitech was chosen. It supports line speeds up to 33.6kbps,

industry standard AT-commands, data compression and error detection. It provides country profiles for virtually all types of telephone standards worldwide. The TIP and RING pins of the Socket Modem can be connected directly to the telephone line (X3).

Other communication standards, like ISDN or GSM, can be supported by simply changing the type of Socket Modem.

While the Socket Modem was implemented as a hardware feature of the current Alarm Control Panel, the implementation of associated software was not part of the original Reference Design task. It is planned to be used and described in a future Application Note.

3.5.7 LIN Driver

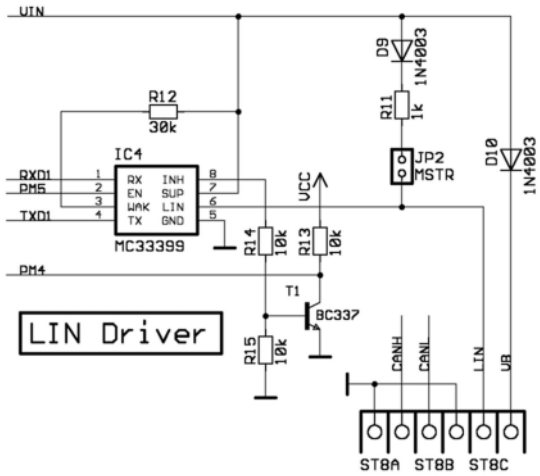


Figure 3-22. LIN driver on SCI1

The MC33399 (IC4) is a physical layer component dedicated to Local Interconnect Network (LIN) sub bus applications. Although LIN is mainly used in the automotive industry, other fields of application are also possible. Since LIN combines high robustness, low standby current and very low cost, it is a good candidate to allow the addition of networked sensors to the Alarm Control Panel Reference Design.

From the view of the microcontroller, LIN is an asynchronous serial protocol, much the same as RS232, even if LIN is using only one wire

(apart from GND). Therefore, IC4 simply can be operated via SCI1 of the MC9S12DP256. Two additional general-purpose I/O-signals (PM4, PM5) are used to enable the LIN driver and to report wake-up events.

A LIN configuration can always have several slaves, but only one master. To make the LIN node on the ACPRD Carrier Board a master node, JP2 must be closed. The terminal block ST8 carries the LIN data signal, GND, and, on a third wire, VB (ca. 12V), which is a possible power supply source for the connected alarm sensors. ST8 also carries the CAN bus signals CANH and CANL, which are not related to the LIN bus functions.

While the LIN driver was implemented as a hardware feature of the current Alarm Control Panel, the implementation of associated software was not part of the original Reference Design task. It is planned to be used and described in a future Application Note.

For more information about the LIN sub bus standard, please refer to [2] and [3].

3.5.8 Graphical LCD

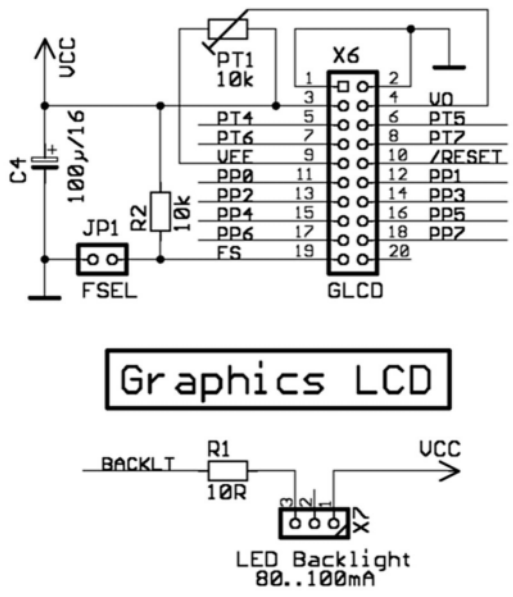


Figure 3-23. Interface to the Graphical LCD

A dot-matrix liquid crystal display (LCD) is connected to X6 as part of the Alarm Control Panel's user interface. The black and white display has full graphics capabilities, the resolution is 240 x 64 pixels. The display module contains controller, character generator, LC-drivers and display RAM.

The display controller is a Toshiba T6963, which has been used by a large number of display manufacturers for their graphical LCD modules for many years. The existence of a large number of compatible modules was one of the main reasons to choose this kind of display. Another reason is the generic electrical interface of the T6963-based LCDs, which is quite easy to design-in. There are eight data lines and four control signals. The following table shows the complete pin configuration:

Pin no.	Symbol	Function	Connected to
1	FG	Frame ground	GND
2	VSS	Power supply (GND)	GND
3	VDD	Power supply (+5V)	VCC
4	VO	Contrast adjust	
5	$\overline{WR}$	Data write	PT4
6	$\overline{RD}$	Data read	PT5
7	$\overline{CE}$	Chip enable	PT6
8	C/D	Command/data select	PT7
9	VEE	Negative voltage output	
10	$\overline{RST}$	Reset	$\overline{RESET}$
11-18	D0–7	Data bus	PP0–7
19	FS	Font selection	
20	n.c.	No connection	

Table 3-2. Interface Specification for to the Graphical LCD

The display contains a white LED backlight, which is supplied separately via connector X7. The white LED has a forward voltage of approx. 4–4.2V. The resistor R1 should have a value of 10 ohms to achieve the desired LED current of 80–100mA. Other backlight technologies may require other setups.

For electrical specifications and programming details, please consult data sheet [4].

3.5.9 Rotary Encoder & Switch

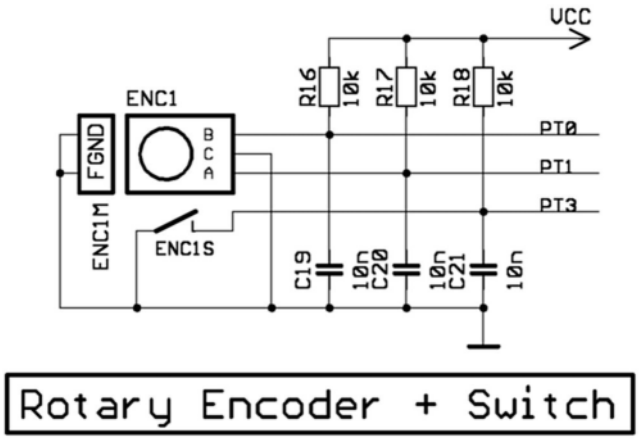


Figure 3-24. Interface to the Rotary Encoder

Another key component of the ACPRD user interface is the rotary encoder ENC1. Today, rotary encoders are used as input devices for a large number of electronic equipment, e.g. mobile phones, PC monitors or home entertainment equipment.

Rotary encoders have at least two outputs that deliver a square wave signal with identical period time, but a shifted phase relation. While the frequency of the output(s) represents the rotation speed, the phase relation of the two signals can be used to determine the direction of rotation. The MCU inputs PT0 and PT1 are used to sense the encoder pulses. For more details on direction recognition, please refer to the corresponding software section.

ENC1 is also equipped with a momentary switch (ENC1S), which can be triggered by pressing the encoder's knob. This switch is connected to input PT3 of the MCU.

The symbol ENC1M in the schematic represents the GND-connection of the encoder body.

3.5.10 DIP Switch

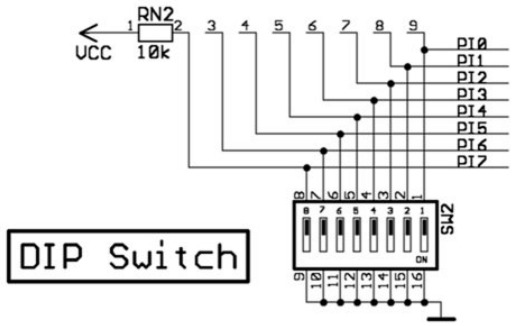


Figure 3-25. Interface to the DIP Switch

SW2 is an eight bit DIP switch. It allows the user to select between different operating modes of the Alarm Control Panel (for details see software description).

The switches have pull-up resistors (resistor network RN2), so an open position reads high, while a closed switch reads low. The switch positions can be detected via the Parallel Input port PI of the S12compact Controller Module. This input port is implemented as a parallel/serial converter and is operated via SPI.

3.5.11 High Power Driver

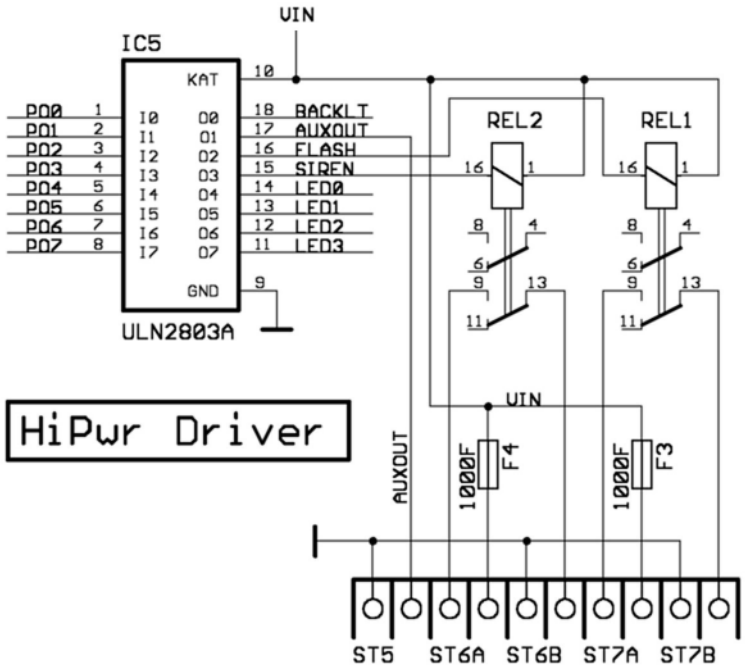


Figure 3-26. High Power Driver

The power driver circuit IC5 consists of eight Darlington stages. It is capable of driving up to 500mA per channel at 50V max. All stages are equipped with integral clamp diodes, making it possible to drive resistive loads as well as inductive loads, such as relays.

The input signals of IC5 come from the serial/parallel converter on the S12compact Controller Module. This is a shift register operated via SPI. During a reset, the outputs PO0–7 of the shift register are reset, therefore all stages of IC5 are turned off out of reset.

The following table shows the function of each channel:

Channel	Function	Reference
0	LCD Backlight	X7
1	Auxiliary Output	ST5
2	Flashlight Relay	ST7/REL1
3	Siren Relay	ST6/REL2
4	Push Button LED0	K0
5	Push Button LED1	K1
6	Push Button LED2	K2
7	Push Button LED3	K3

Table 3-3. Function of each power channel

3.5.12 Push Buttons & LEDs

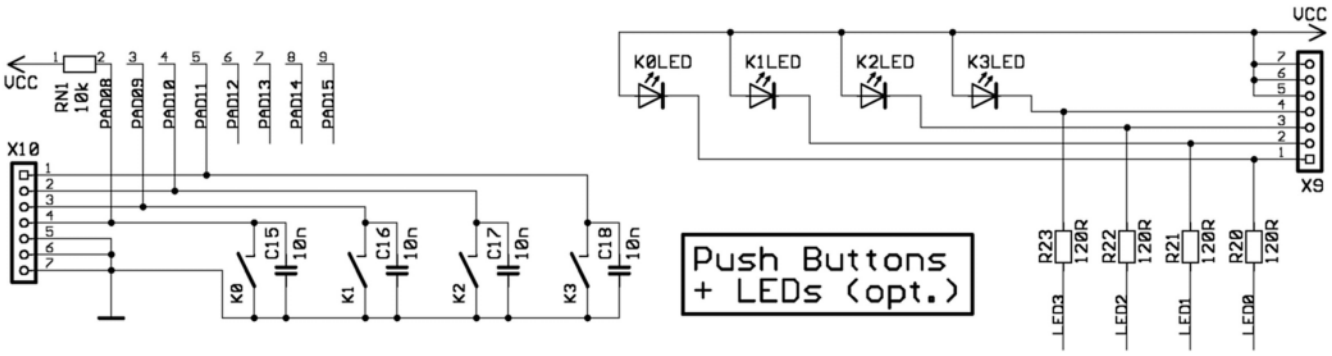


Figure 3-27. Push button switches and optional LEDs

There are four push button switches (K0–3) connected to inputs PAD08–11. These inputs belong to the 2nd ATD module of the MC9S12DP256, which are operated as digital logic-level inputs within the Alarm Control Panel Reference Design. The switches are normally open and have a pull-up resistor (RN1), thus delivering an H-signal on PAD08–11. C15–C18 may be added for debouncing, but normally they are not necessary. The Push Button Switches are read by software cyclically (polling mode), so no interrupt generation is necessary in this case.

As an option, Push Buttons with an integrated LED may be used, which can be driven by stages 4–7 of the High Power Driver circuit. By default, the Push Buttons on the ACPRD have no such LED.

3.5.13 Audio Option Select

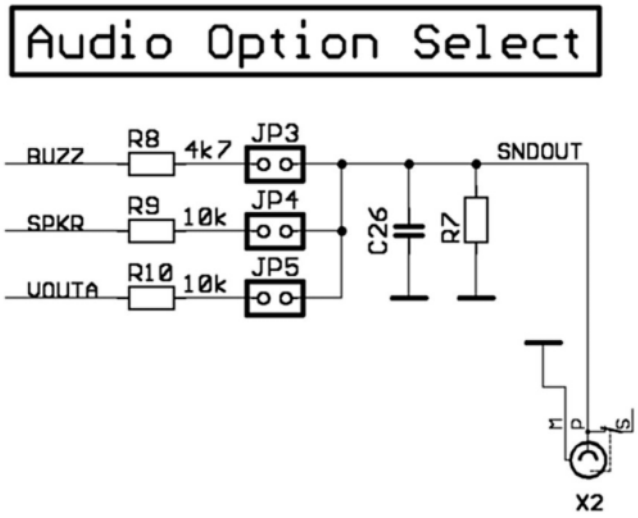


Figure 3-28. Selection of Audio Outputs

The jumpers JP3, JP4 and JP5 select the source(s) for an external audio amplifier. The following table summarizes the options available:

Jumper	Signal	close to enable...
JP3	BUZZ	Buzzer output (Timer Channel PT2) BR3 on the S12compact must be closed as well!
JP4	SPKR	Speaker output of Socket Modem
JP5	VOUTA	16-bit D/A-Converter output of S12compact (Hi-quality audio), optional

Table 3-4. Audio Output Options

3.5.14 External Connections

Figure 3-29 shows the external connections for the Alarm Control Panel:

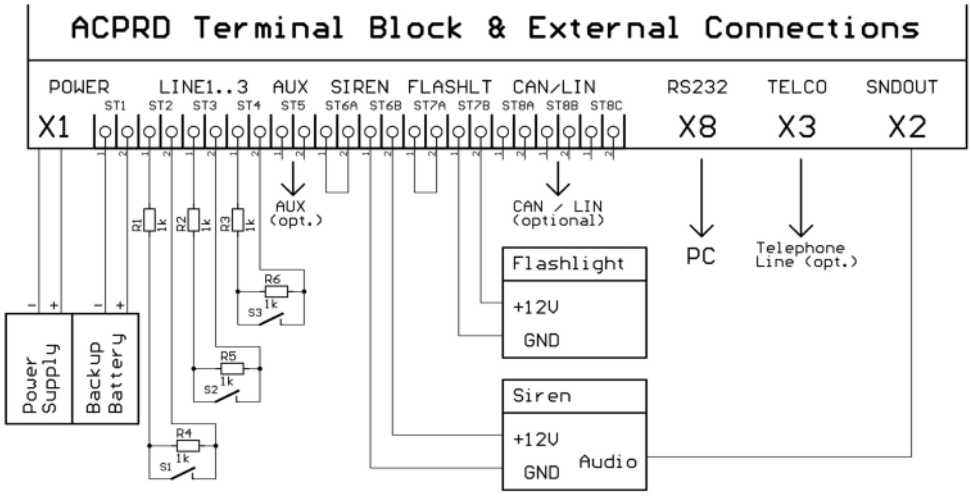


Figure 3-29. External connections for the Alarm Control Panel

Section 4. Software Description

4.1 Contents

4.2 Software Overview

4.2.1 Module Structure

The software for the Alarm Control Panel consists of three layers. The lowest layer deals with the on-chip peripherals of the MC9S12DP256 MCU. These functions could be re-used for other HCS12 projects very easily. The names of such modules start with “S12_”. The T6963 module is related hardware. It contains a low-level driver for the T6963 LCD controller.

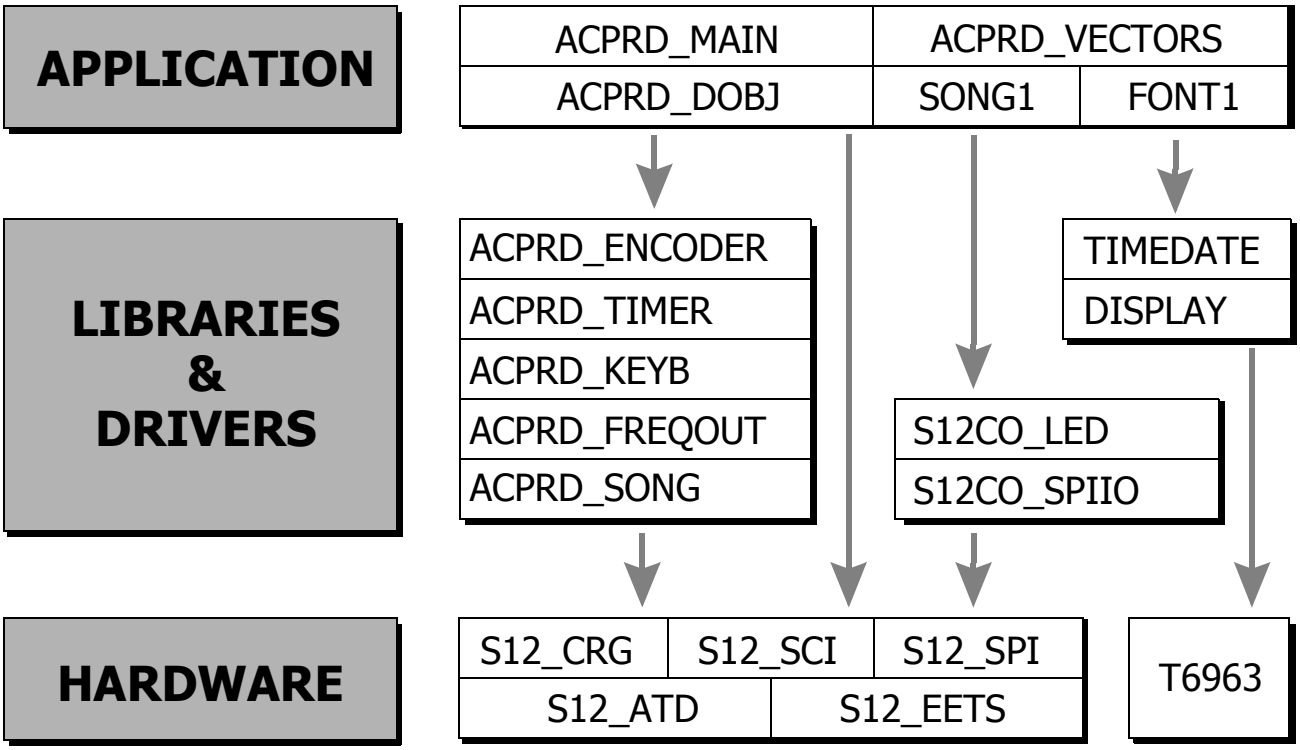


Figure 4-1. Module Structure of the Alarm Control Panel software

The next layer contains library functions specific to the S12compact Controller Module hardware. Names of modules belonging to this layer start with “S12CO_”. The modules TIMEDATE and DISPLAY can also be considered library modules, although they are not based on any specific hardware.

The top layer is the application layer. It contains the main() function as well as all other parts of the application framework. The display objects for the graphical user interface are concentrated in a separate file, as are the interrupt and reset vectors. The names of application layer modules normally start with “ACPRD_”, except for some files containing general purpose constant data definitions.

The implementation part of each module is contained in a “.C” file, while a header file contains declarations such as defines and function prototypes. The header file has the extension “.H” and the same base name as the C-file.

4.2.2 Additional Header Files

The address assignments for the MCU control registers are contained in the header file HCS12DP256.H and the bit masks used to access individual features of these control registers are defined in the module header files.

DATATYPES.H contains miscellaneous definitions concerning data types and also some compiler dependent macros.

4.2.3 Module Initialization

Every module provides an initialization function, which should be called once in the initialization phase of the application. The name of such a function always starts with init...(), e.g. initSPI() or initEETS().

4.3 Software Module Details

4.3.1 S12 Analog-to-Digital Converter Module (S12_ATD)

This module contains functions for driving the Analog-to-Digital converter of the HCS12. The MC9S12DP256 contains two such blocks, but this software module only deals with the first block (ATD0), which covers the input signals PAD00–PAD07.

initATD0() should be called once in the initialization phase of the application. It enables the ATD hardware module and selects 10-bit resolution mode. getATD0() performs a single ATD conversion on a single channel. The channel number argument must be in the range 0–7. The function will wait until the conversion is finished which will cause an additional delay of about 10 microseconds.

The following two functions provide another method to handle the A/D converter. Calling startATD0() will just start the ATD sequence for all eight input channels. The function will return immediately but will not deliver any result. To read the conversion results, readATD0() must be called in a second step. The argument “results” is a pointer to a memory area intended to store the eight conversion values. readATD0() must not

be called before the conversion sequence is complete. Finally, handleATD0() demonstrates how to use startATD0() and readATD0() to cyclically perform ATD conversions and the calculation of arithmetic mean values for each input channel.

4.3.2 S12 Clocks and Reset Generator Module (S12_CRG)

This module deals with the Phase Locked Loop (PLL), which is part of the CRG hardware module of the HCS12. There is only one function, initPLL(), which is responsible for setting up the PLL, waiting until a stable clock output is available and finally switching over from oscillator clock to PLL clock. As a result, the internal system clock (ECLK) will change from 8 MHz to 24 MHz.

4.3.3 S12 Enhanced Capture Timer Module (S12_ECT)

For this module, there is just a header file containing bit mask definitions for the ECT hardware module of the HCS12. This peripheral block is rather complex, so there is no general setup which would cover all available functions. Rather than providing a generic ECT module, timer related functions are split into “themes” according to the features needed for the Alarm Control Panel application. For details, please see ACPRD_TIMER, ACPRD_ENCODER and ACPRD_FREQOUT.

4.3.4 S12 EEPROM Module (S12_EETS)

The EEPROM of the HCS12 is partitioned into 4-byte sectors. The functions of this module can access every EEPROM sector on an individual basis. In order to simplify the handling of data that is smaller than a sector, the module functions consider any information to be a 32-bit “item”. If mostly smaller data portions were used by the application (e.g. bytes), EEPROM space would be wasted. However, since the total size of the EEPROM block of the MC9S12DP256 is 4K bytes, there is sufficient space to store a limited amount of configuration data (as in the ACPRD).

Before using the EETS module it is important to write to the EEPROM Clock Divider Register, which is done in the initEETS() function. The proper value depends on the oscillator clock, which is defined in the CRG header file S12_CRG.H.

4.3.5 S12 Serial Communications Interface Module (S12_SCI)

The SCI module provides transmit and receive functions for both asynchronous serial interfaces, SCI0 and SCI1. SCI0 is connected either to a host (PC) or to a modem. SCI1 is connected to the on-board LIN driver. The functions use polling mode for both, receive and transmit.

Please note, that the routines of this module are generic data transfer functions, i.e. protocol or handshaking features are not implemented. This means in particular, there are no LIN-specific software drivers contained in this initial release of the ACPRD.

4.3.6 S12 Serial Peripheral Interface Module (S12_SPI)

This module provides routines for the first of three SPI hardware modules of the MC9S12DP256, which is designated SPI0. The module is setup as an SPI master.

SPI transfers are always bidirectional, therefore only a single transfer function is needed instead of separate transmit and receive functions. This function xferSPI0() will wait for the completion of the transfer, thus a delay will occur, the length of which will depend on the SPI clock speed, but not on the SPI peripherals. Since the SPI clock can be as high as 12 MHz (ECLK/2), the delay is not a problem in most cases.

4.3.7 S12compact SPI Input Output Module (S12CO_SPIIO)

On the S12compact Controller Module, SPI0 of the MCU can be used to access several peripheral chips. The chip select signals needed for this purpose are generated by an 1-out-of-8 decoder (IC4), which is driven by the MCU signals PH4, PH5 and PH6. If all of these signals are high then output Y7 of IC4 will become low. This is the default, inactive state

of the decoder (Y7 is not used). Any other bit combination on PH4–6 will select one of the chip select signals $\overline{\text{SPICS0}}$–$\overline{\text{6}}$.

putSPIPO() activates $\overline{\text{SPICS5}}$ and transfers a byte via SPI0 to the shift register IC9 on the S12compact Controller Module. The shift register will latch the value with the rising edge of the chip select signal. There is no feedback path to recognize the values present on the shift register outputs, therefore a software shadow register should be used to track the current status in the application code.

IC8 is a parallel-in/serial-out shift register. The input latch can be triggered by pulling $\overline{\text{SPICS4}}$ low for a short time. getSPIPI() will do that, followed by activating $\overline{\text{SPICS3}}$ in order to shift out the latched input value. The function then will return the byte read.

Both shift registers can be daisy-chained outside the S12compact board by connecting additional shift registers to NEXTPO or PREVPI, respectively.

4.3.8 S12compact LED Module (S12CO_LED)

This module simply switches the on-board LED of the S12compact. The LED is driven by /XCLKS, which is identical to PortE[7] of the MCU.

There is no code file for this module, since all functions are implemented as macro definitions in the header file.

4.3.9 T6963 Module (T6963)

The Toshiba T6963 is a LCD controller which is used on a great variety of LC-Display modules from different manufacturers. Although not the latest device, it is very popular on the market.

The T6963 software module implements low-level functions to access this display controller. This module implements low level functions to write single bytes, read the status and transfer a block of (display) data to the LCD. Please refer to [4] for details on setup and programming of the T6963.

There are two functions (refreshT6963c() and refreshT6963a()) which are used to update the display information. The two functions are called cyclically from the periodic system interrupt. As a result, all display information is transferred from the display buffer to the display RAM of the LCD.

4.3.10 Display Module (DISPLAY)

The Display Module implements a virtual display screen and character based access functions (text editor). It provides a screen buffer and maintains the position of a cursor. This module can be used for different kinds of physical display devices, for instance (but not limited to) LC-Displays.

The display size is defined in the header file. Every display position is defined by a word in the display buffer disp_buf[]. The LSB of the word is the character code, while the MSB is reserved to hold attribute information (reverse, blink).

The function putcDisp() performs the editor function, i.e. control code handling and cursor movement. _putDisp() is then called (if necessary), which is the only function which actually writes to the display buffer.

The functions starting with prDisp...() are helper functions intended to format the output. The ANSI-C standard library function printf() could perform most of the desired formatting, but the code size of this function is substantial and, at the same time, some desirable features are not covered. So, implementing some small - but specific - functions is often better than using printf().

4.3.11 Time/Date Module (TIMEDATE)

A traditional approach to express time and date information on UNIX-based machines is the number of seconds since January 1 1970. The TIMEDATE module implements calculations based on this model.

This module contains no hardware dependencies, except the functions time() and stime() which access the second counter variable systime of the ACPRD.

asctime() is implemented using sprintf() which consumes a considerable amount of code space. Therefore, asctime() must be activated individually by defining the preprocessor macro __USE_ASCTIME.

This module is Y2K compliant but will fail after 2099.

4.3.12 ACPRD Keyboard Module (ACPRD_KEYB)

The four push buttons of the Alarm Control Panel are handled by this module. Also, the momentary switch of the rotary encoder is treated as a (fifth) key.

The key press events are buffered. hasKeyb() checks if there are any key codes in the buffer. This function returns immediately. Conversely, getKeyb() will wait for a key press, if the buffer is empty. After that, the function will remove the oldest key code from the buffer and return it. putKeyb() is only used internally to place a key code into the buffer.

The array keyb_hist is used to maintain a history over time of the recent states of each key. This history is updated every time scanKeyb() is called, which is done as part of a cyclic interrupt routine. The keys are connected to binary port pins of the MCU (PORTAD1 is used as a digital input port here as well). The states (High or Low) of these pins are shifted into the keyb_hist variable. The detection of a “HLLL” pattern means a high to low transition followed by a stable low reading and therefore is considered a valid key press event. This has been proven to be an easy method for debouncing mechanical push buttons.

4.3.13 ACPRD Encoder Module (ACPRD_ENCODER)

The encoder module utilizes the Enhanced Capture Timer to handle a rotary encoder connected to port pins PT0 and PT1. These two inputs are capable of generating an Input Capture interrupt, which is used to detect if the rotary encoder was moved.

Most of the work is done during initialization. The interrupt routine for encoder channel A isrIC1() just needs to check whether the two encoder channels have the same level or not. Depending on the result of this

investigation, the encoder variable encoder_count will be incremented or decremented.

The pulses on PT0 (coming from encoder channel B) will also generate an interrupt, but the related interrupt service routine will do nothing, except clear the interrupt flag.

Calling hasEncoder() will return the current encoder counter value. getEncoder() will not only return the current encoder_count value, but also reset this variable.

4.3.14 ACPRD Timer Module (ACPRD_TIMER)

This module uses the Modulus Down Counter within the ECT to generate a periodic interrupt every 10 milliseconds. The interrupt routine isrMDCU() just clears the local interrupt flag and then calls the user supplied interrupt function, which is implemented in ACPRD_MAIN.

4.3.15 ACPRD Frequency Output Module (ACPRD_FREQOUT)

To generate a frequency output, one of the Output Compare channels of the ECT in the MC9S12DP256 is used. initFreqOut() establishes a prescaler value of 16 for the main timer clock. This value is favourable to generate frequencies in the audible range. The lowest possible frequency becomes 22.9Hz (24MHz / (16 * 65536)).

setFreqOut() establishes a frequency according to the function argument, which is the period time in microseconds. For this purpose, it calculates the number of timer ticks for the period time given and stores the result in the variable freqout_tticks. If the function argument is zero, the frequency output will be disabled by disconnecting the output pin from the OC2 logic. isrOC2() is the interrupt handler for the Output Compare 2 channel. It is invoked cyclically with the interval value stored in freqout_tticks.

4.3.16 ACPRD Song Module (ACPRD_SONG)

ACPRD_SONG can play a melody based on the frequency generation capabilities of ACPRD_FREQOUT. A simple pseudo code is used to define the frequency and duration of music notes (an example is given in SONG1.C). The argument for startSong() is a pointer to a zero-terminated string, consisting of such music pseudo codes.

handleSong() has to be called cyclically to proceed playing the song. This is done (as a background task) every 30ms as part of the periodic ACPRD system interrupt.

4.3.17 ACPRD Main Module (ACPRD_MAIN)

The ACPRD Main Module contains the main program loop that handles the user interface (display menu and other visual objects, check push buttons and rotary encoder) and a background task, based on the periodic interrupt every10ms().

Displayable objects are contained in an extra source file (ACPRD_DOBJ.C) which is included in ACPRD_MAIN.C. Also, the display font definition file FONT1.C and the music definition file SONG1.C are included and therefore become part of the main module.

Keyboard scanning, time/date calculation, sound generation, refreshing the display and (of course) checking the alarm lines is all done in the “background” task, which is the 10ms periodic interrupt.

4.3.18 ACPRD Vectors Module (ACPRD_VECTORS)

This is the table of interrupt and reset vectors for the MC9S12DP256. All unused vectors point to the dummy interrupt service routine isrErrorHandler(), which can be used to identify any spurious interrupts during debugging.

4.4 How to build the project

To build the ACPRD, add the following source files to a compiler project:

- acprd_main.c
- acprd_timer.c
- acprd_encoder.c
- acprd_keyb.c
- acprd_freqout.c
- acprd_song.c
- acprd_vectors.c
- s12co_spiio.c
- s12_atd.c
- s12_crg.c
- s12_eets.c
- s12_sci.c
- s12_spi.c
- t6963.c
- display.c
- timedate.c

The linker options should be set as follows:

- text segment start at 0xC000
- data segment start at 0x3800
- stack pointer set to 0x4000

The source code is ANSI-C compliant with the exception of using // for comments. Some compiler specific issues are covered in the header file DATATYPES.H. The project was implemented using ImageCraft's ICC12 compiler and tested under Metrowerks CodeWarrior and Cosmic Cx12, though it should be possible to build the project with any other HC(S)12 compiler.

Software Description

Section 5. Literature

5.1 Contents

5.2 Documentation

[1] Motorola, Inc.: MC9S12DP256B Device User Guide; Document Number: 9S12DP256BDGV2/D (also consult the various HCS12 Block User Guides)

[2] Homepage of the LIN organization: http://www.lin-subbus.org

[3] Motorola, Inc.: MC33399 Advance Information; Document Number: MC33399/D

[4] Powertip Technology Corp.: PG24064 Data Sheet (also containing programming instructions for the Toshiba T6963 LCD controller)

5.3 ACPRD on the Web

There is a Web project page, dedicated to (possible) future updates and enhancements of the Alarm Control Panel Reference Design project:

http://hc12web.de/acprd

Literature

Section 6. Appendix

6.1 Contents

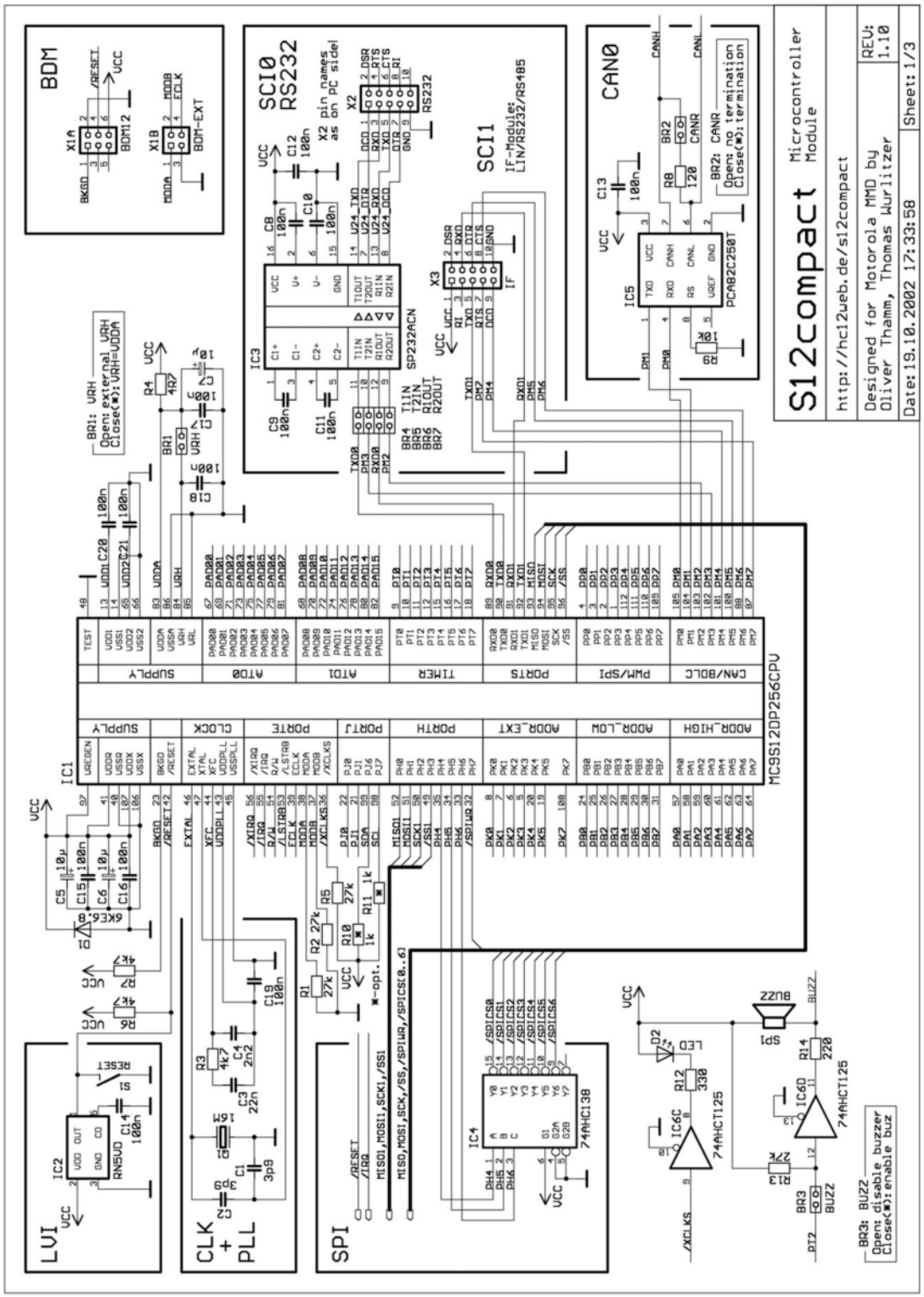


Figure 6-1. Schematics of the S12compact module (Part 1)

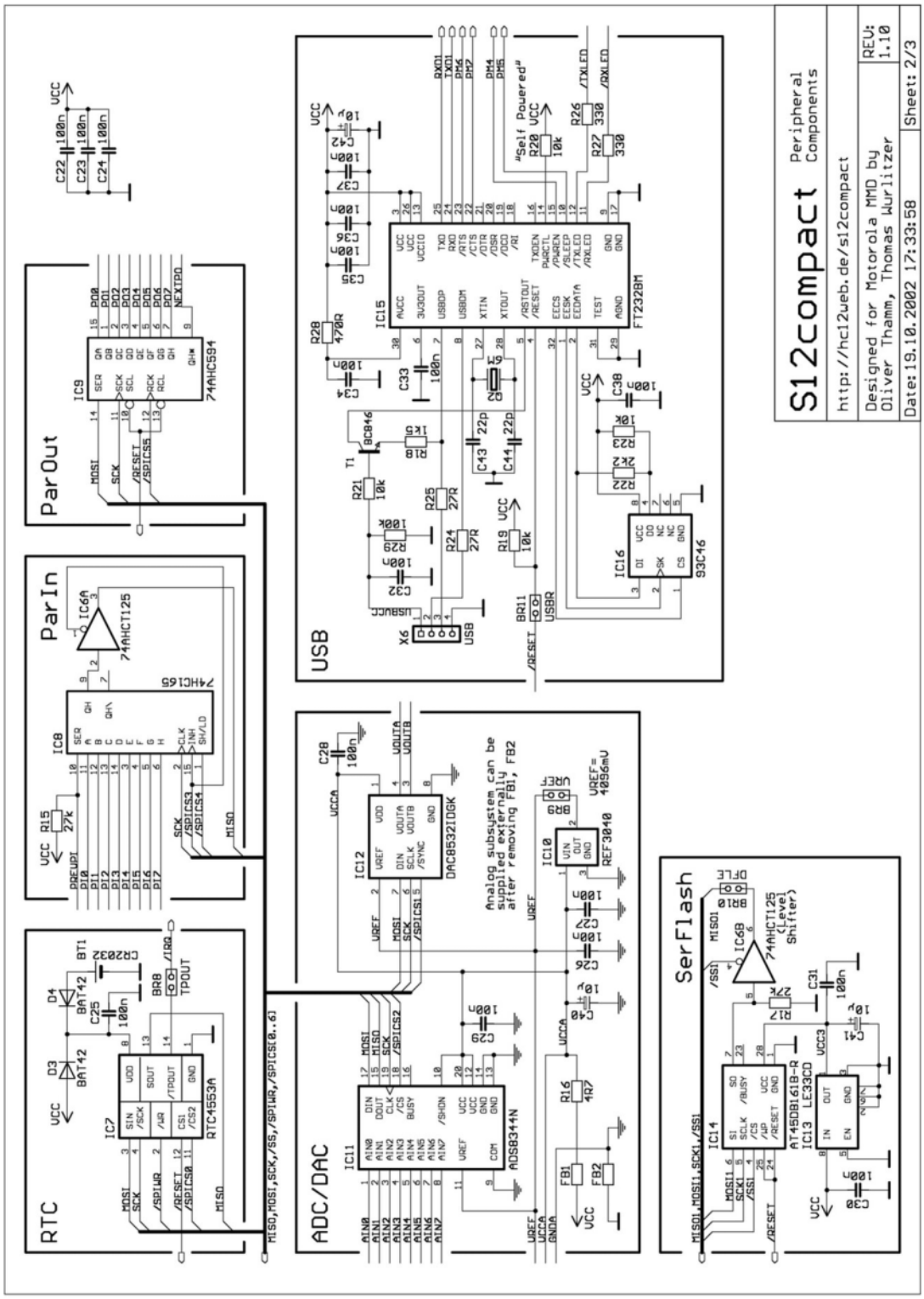


Figure 6-2. Schematics of the S12compact module (Part 2)

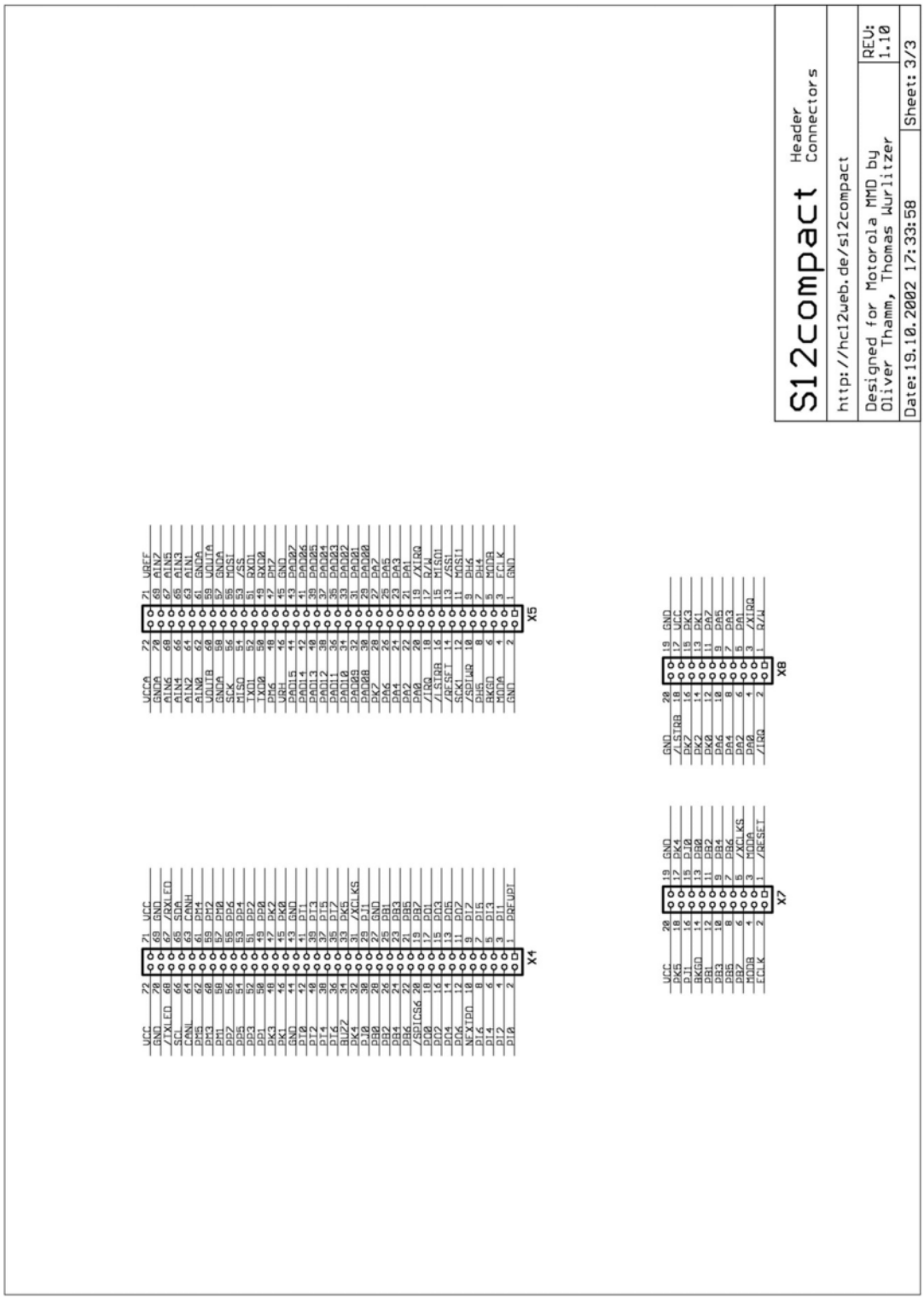


Figure 6-3. Schematics of the S12compact module (Part 3)

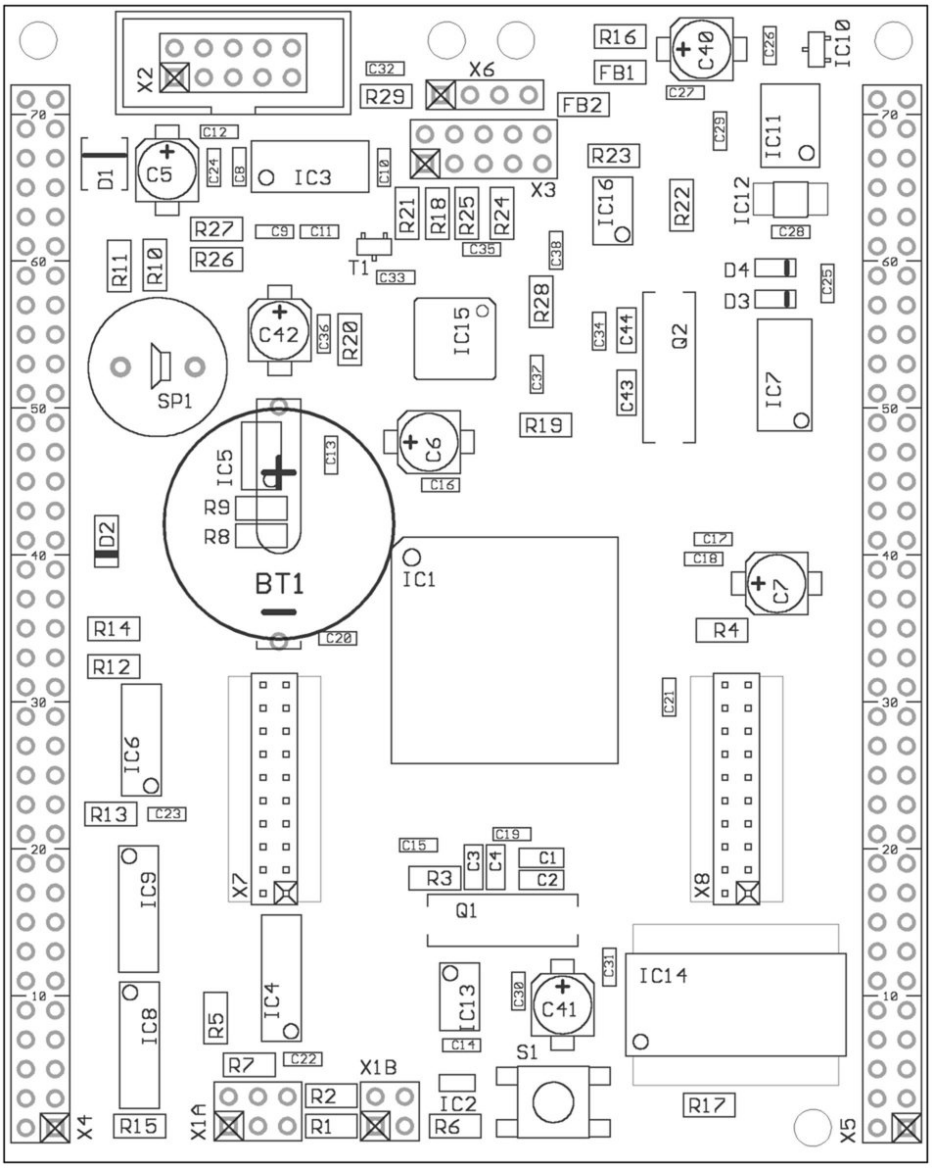


Figure 6-4. Parts Location Diagram of the S12compact module

Appendix

Table 6-1. S12compact Bill of Materials

Reference	Value	Package	Description	Supplier (Distr.)
IC1	MC9S12DP256BCPV	LQFP112	HCS12 MCU	Motorola
IC2	RN5VD47A	SOT23-5	Reset Controller	Ricoh
IC3	SP232ACN	SO16	RS232 Transceiver	Sipex
IC4	74AHC138	SO16	Decoder	ON Semi.
IC6	74AHCT125	SO14	Buffer	TI
IC8	74HC165	SO16	Shift Register	Fairchild
IC9	74AHC594	SO16	Shift Register	TI
IC5	PCA82C250T/N4	SO8	CAN Physical Interface	Philips
Q1	AT51CD2	SMD	Quartz Crystal	NDK (Frischer)
D2	LTST-C150UBKT	1206	LED blue	LiteOn
D1	P6SMB6.8A	SMB	Supressor Diode 6.8V	
C1 C2	3p9	0805	Ceramic Capacitor	
C43 C44	22p	0805	Ceramic Capacitor	
C4	2n2	0805	Ceramic Capacitor	
C3	22n	0805	Ceramic Capacitor	
C8 ... C38	100n	0603	Ceramic Capacitor	
C5 C6 C7 C40 C41 C42	10µ/35V	SMD 5mm x 5.5mm	Electrolytic Capacitor	
R4 R16	4R7	1206	Resistor	
R8 R12	120R	1206	Resistor	
R14	220R	1206	Resistor	
R10 R11	1k	1206	Resistor	
R3 R6 R7	4k7	1206	Resistor	
R9 R19 R20 R21 R23	10k	1206	Resistor	
R1 R2 R5 R13 R15 R17	27k	1206	Resistor	
R29	100k	1206	Resistor	
FB1 FB2	ACB3216M-600-T	0tan20ar20	Ferrite Bead	TDK
S1	DTSM61N	SMD	Push Button Switch	Diptronics
SP1	ALP-60P	TH	Buzzer	

Note: Optional components not listed

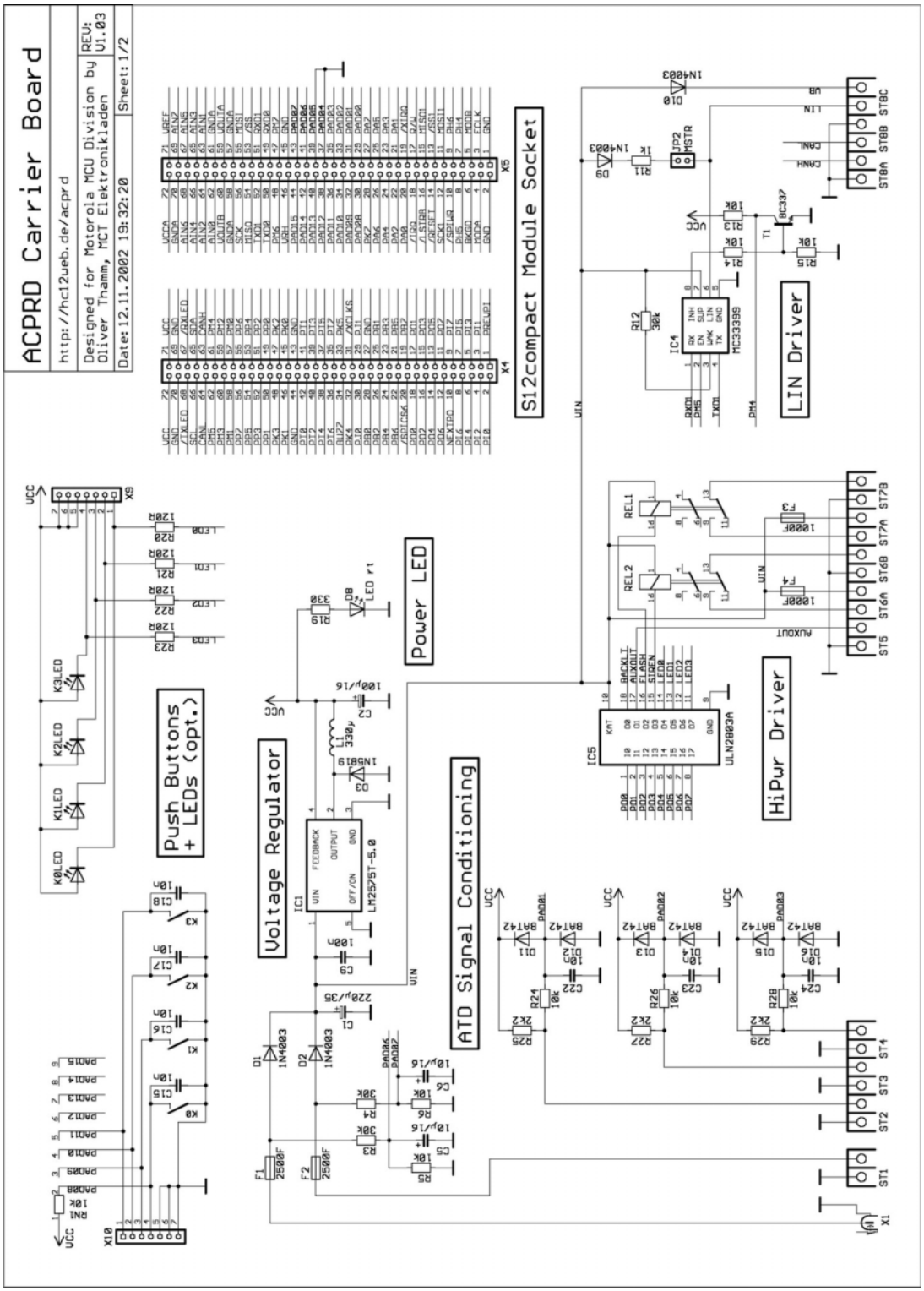


Figure 6-5. Schematics of the ACPRD Carrier Board (Part 1)

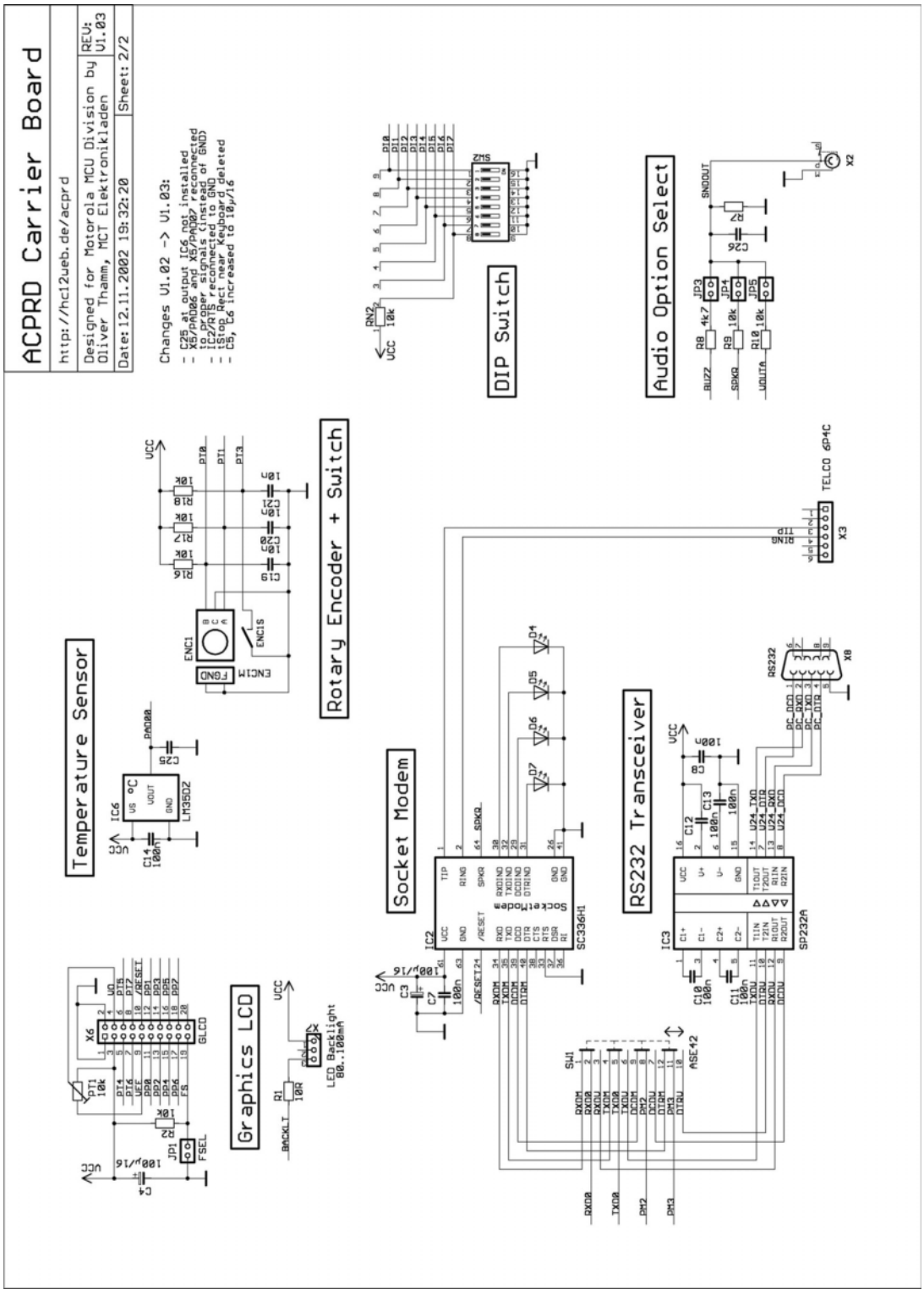


Figure 6-6. Schematics of the ACPRD Carrier Board (Part 2)

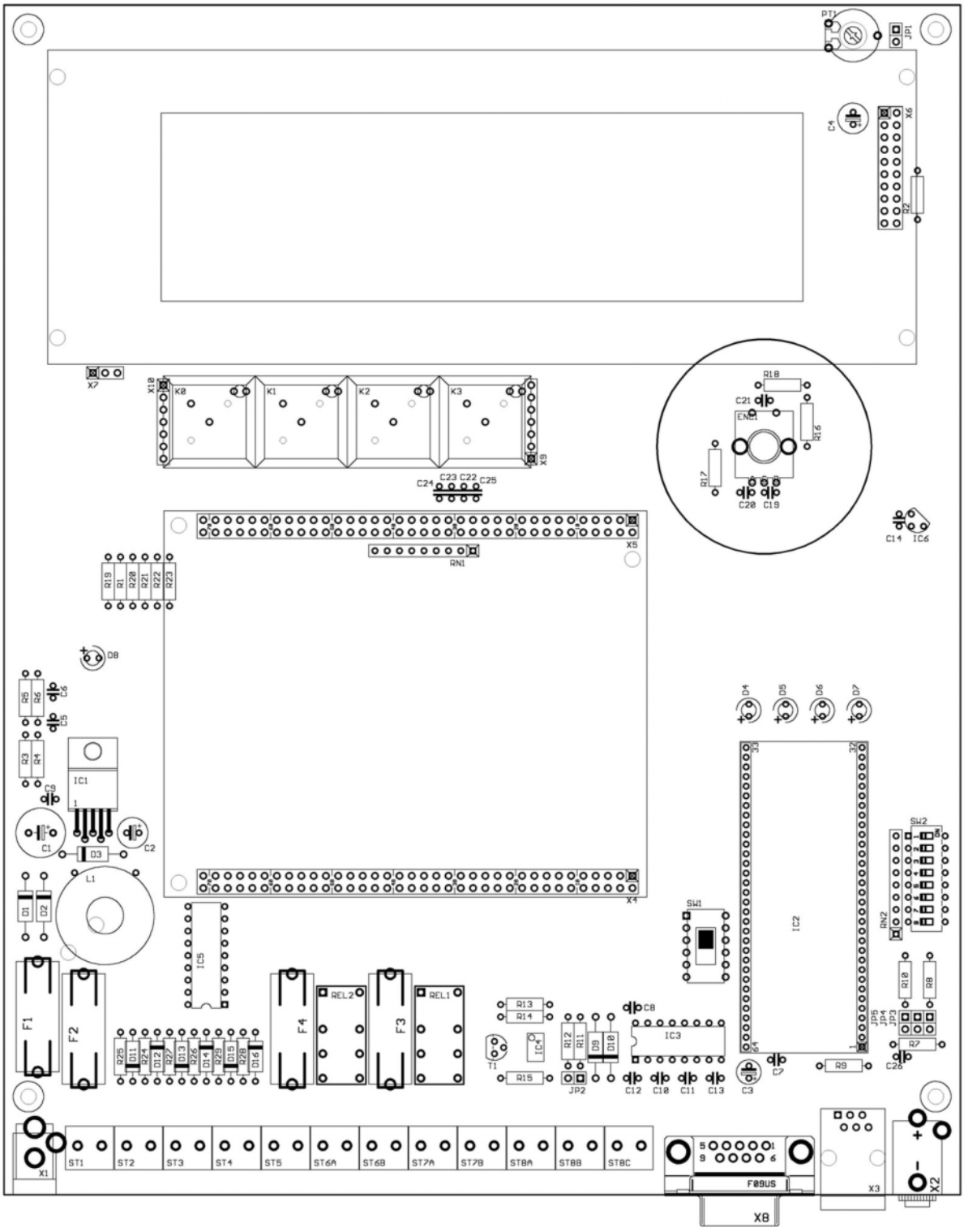


Figure 6-7. Parts Location Diagram of the ACPRD Carrier Board

Table 6-2. ACPRD Carrier Board Bill of Materials

Reference	Value	Package	Description	Supplier (Distr.)
LCD	PG24064		LC-Display with LED Backlight	Powertip (Actron)
ENC1	STEC11B03		Rotary Encoder, switched	ALPS (RS)
IC1	LM2575T-5	TO220-5	Switching Voltage Regulator	ON Semi.
IC2	SC336H-1		Socket Modem	Conexant (Unitronic)
IC3	SP232A	DIP16	RS232 Transceiver	Sipex
IC4	MC33399	SO8	LIN Physical Interface	Motorola
IC5	ULN2803A	DIP18	Power Driver	Allegro
IC6	LM35DZ	TO92	Celsius Sensor	National Semi.
L1	330µH/1A		Storage Choke	Talema
D1 D2 D9 D10	1N4007	DO41	Diode	
D3	1N5819	DO41	Schottky Diode	
D11 D12 D13 D14 D15 D16	BAT42	DO35	Schottky Diode	
D4 D5 D6 D7 D8	LED red		LED	
T1	BC337-25		Transisitor	
C15 C16 C17 C18	10n	1206	Ceramic Capacitor	
C19 C20 C21 C22 C23 C24	10n		Ceramic Capacitor	
C7 C8 C9 C10 C11 C12 C13 C14	100n		Ceramic Capacitor	
C5 C6	10µ/16		Electrolytic Capacitor	
C2 C3 C4	100µ/16		Electrolytic Capacitor	
C1	220µ/35		Electrolytic Capacitor	
R1	10R		Resistor	
R20 R21 R22 R23	120R		Resistor	
R19	330R		Resistor	
R11	1k		Resistor	
R25 R27 R29	2k2		Resistor	
R8	4k7		Resistor	
R2 R5 R6 R9 R10 R13 R14 R15 R16 R17 R18 R24 R26 R28	10k		Resistor	
R3 R4 R12	30k		Resistor	
RN1 RN2	10k x 8	SIL9	Resistor Network	
PT1	10k	5mm/10mm	Potentiometer	
K0 K1 K2 K3	MTP1241	0.75	"Keyboard Switch	Schurter
ST1..ST8	STLZ951	0.2	"Terminal Block	PTR
ST1..ST8	ARK950	0.2	"Terminal Block	PTR
REL1 REL2	D2N12		Print Relais 2	Tyco
SW1	ASE42FN		Slide Switch	Tyco
SW2	DIP-SW8	DIP16	DIP Switch	
X3	P131	6P4C	Modular Connector	Lumberg
X2	KLBR2		Audio Jack Connector	Lumberg
X1	NEB21R	2.1mm center pin	Power Connector	Lumberg
X8	Sub-D9 female	print, 90 deg.	RS232 Connector	
F1 F2	F 2,5A	5mm x 20mm	Fuse, socketed	
F3 F4	F 1A	5mm x 20mm	Fuse, socketed	

Glossary

A — See “accumulators (A and B or D).”

accumulators (A and B or D) — Two 8-bit (A and B) or one 16-bit (D) general-purpose registers in the CPU. The CPU uses the accumulators to hold operands and results of arithmetic and logic operations.

acquisition mode — A mode of PLL operation with large loop bandwidth. Also see ’tracking mode’.

address bus — The set of wires that the CPU or DMA uses to read and write memory locations.

addressing mode — The way that the CPU determines the operand address for an instruction. The M68HC12 CPU has 15 addressing modes.

ALU — See “arithmetic logic unit (ALU).”

analogue-to-digital converter (ATD) — The ATD module is an 8-channel, multiplexed-input successive-approximation analog-to-digital converter.

arithmetic logic unit (ALU) — The portion of the CPU that contains the logic circuitry to perform arithmetic, logic, and manipulation operations on operands.

asynchronous — Refers to logic circuits and operations that are not synchronized by a common reference signal.

ATD — See “analogue-to-digital converter”.

B — See “accumulators (A and B or D).”

baud rate — The total number of bits transmitted per unit of time.

BCD — See “binary-coded decimal (BCD).”

binary — Relating to the base 2 number system.

Glossary

binary number system — The base 2 number system, having two digits, 0 and 1. Binary arithmetic is convenient in digital circuit design because digital circuits have two permissible voltage levels, low and high. The binary digits 0 and 1 can be interpreted to correspond to the two digital voltage levels.

binary-coded decimal (BCD) — A notation that uses 4-bit binary numbers to represent the 10 decimal digits and that retains the same positional structure of a decimal number. For example,

234 (decimal) = 0010 0011 0100 (BCD)

bit — A binary digit. A bit has a value of either logic 0 or logic 1.

branch instruction — An instruction that causes the CPU to continue processing at a memory location other than the next sequential address.

break module — The break module allows software to halt program execution at a programmable point in order to enter a background routine.

breakpoint — A number written into the break address registers of the break module. When a number appears on the internal address bus that is the same as the number in the break address registers, the CPU executes the software interrupt instruction (SWI).

break interrupt — A software interrupt caused by the appearance on the internal address bus of the same value that is written in the break address registers.

bus — A set of wires that transfers logic signals.

bus clock — See "CPU clock".

byte — A set of eight bits.

CAN — See "Motorola scalable CAN."

CCR — See “condition code register.”

central processor unit (CPU) — The primary functioning unit of any computer system. The CPU controls the execution of instructions.

CGM — See “clock generator module (CGM).”

clear — To change a bit from logic 1 to logic 0; the opposite of set.

clock — A square wave signal used to synchronize events in a computer.

clock generator module (CGM) — The CGM module generates a base clock signal from which the system clocks are derived. The CGM may include a crystal oscillator circuit and/or phase-locked loop (PLL) circuit.

comparator — A device that compares the magnitude of two inputs. A digital comparator defines the equality or relative differences between two binary numbers.

computer operating properly module (COP) — A counter module that resets the MCU if allowed to overflow.

condition code register (CCR) — An 8-bit register in the CPU that contains the interrupt mask bit and five bits that indicate the results of the instruction just executed.

control bit — One bit of a register manipulated by software to control the operation of the module.

control unit — One of two major units of the CPU. The control unit contains logic functions that synchronize the machine and direct various operations. The control unit decodes instructions and generates the internal control signals that perform the requested operations. The outputs of the control unit drive the execution unit, which contains the arithmetic logic unit (ALU), CPU registers, and bus interface.

COP — See "computer operating properly module (COP)."

CPU — See “central processor unit (CPU).”

CPU12 — The CPU of the MC68HC12 Family.

CPU clock — Bus clock select bits BCSP and BCSS in the clock select register (CLKSEL) determine which clock drives SYSCLK for the main system, including the CPU and buses. When EXTALi drives the SYSCLK, the CPU or bus clock frequency (f_o) is equal to the EXTALi frequency divided by 2.

CPU cycles — A CPU cycle is one period of the internal bus clock, normally derived by dividing a crystal oscillator source by two or more so the high and low times will be equal. The length of time required to execute an instruction is measured in CPU clock cycles.

CPU registers — Memory locations that are wired directly into the CPU logic instead of being part of the addressable memory map. The CPU always has direct access to the information in these registers. The CPU registers in an M68HC12 are:

- A (8-bit accumulator)
- B (8-bit accumulator)
  - D (16-bit accumulator formed by concatenation of accumulators A and B)
- IX (16-bit index register)
- IY (16-bit index register)
- SP (16-bit stack pointer)

- PC (16-bit program counter)
- CCR (8-bit condition code register)

cycle time — The period of the operating frequency: $t_{CYC} = 1/f_{OP}$.

D — See “accumulators (A and B or D).”

decimal number system — Base 10 numbering system that uses the digits zero through nine.

duty cycle — A ratio of the amount of time the signal is on versus the time it is off. Duty cycle is usually represented by a percentage.

ECT — See “enhanced capture timer.”

EEPROM — Electrically erasable, programmable, read-only memory. A nonvolatile type of memory that can be electrically erased and reprogrammed.

EPROM — Erasable, programmable, read-only memory. A nonvolatile type of memory that can be erased by exposure to an ultraviolet light source and then reprogrammed.

enhanced capture timer (ECT) — The HC12 Enhanced Capture Timer module has the features of the HC12 Standard Timer module enhanced by additional features in order to enlarge the field of applications.

exception — An event such as an interrupt or a reset that stops the sequential execution of the instructions in the main program.

fetch — To copy data from a memory location into the accumulator.

firmware — Instructions and data programmed into nonvolatile memory.

free-running counter — A device that counts from zero to a predetermined number, then rolls over to zero and begins counting again.

full-duplex transmission — Communication on a channel in which data can be sent and received simultaneously.

hexadecimal — Base 16 numbering system that uses the digits 0 through 9 and the letters A through F.

high byte — The most significant eight bits of a word.

illegal address — An address not within the memory map

illegal opcode — A nonexistent opcode.

index registers (IX and IY) — Two 16-bit registers in the CPU. In the indexed addressing modes, the CPU uses the contents of IX or IY to determine the effective address of the operand. IX and IY can also serve as a temporary data storage locations.

input/output (I/O) — Input/output interfaces between a computer system and the external world. A CPU reads an input to sense the level of an external signal and writes to an output to change the level on an external signal.

instructions — Operations that a CPU can perform. Instructions are expressed by programmers as assembly language mnemonics. A CPU interprets an opcode and its associated operand(s) and instruction.

inter-IC bus (I^2C) — A two-wire, bidirectional serial bus that provides a simple, efficient method of data exchange between devices.

interrupt — A temporary break in the sequential execution of a program to respond to signals from peripheral devices by executing a subroutine.

interrupt request — A signal from a peripheral to the CPU intended to cause the CPU to execute a subroutine.

I/O — See “input/output (I/0).”

jitter — Short-term signal instability.

latch — A circuit that retains the voltage level (logic 1 or logic 0) written to it for as long as power is applied to the circuit.

latency — The time lag between instruction completion and data movement.

least significant bit (LSB) — The rightmost digit of a binary number.

logic 1 — A voltage level approximately equal to the input power voltage (V_{DD}).

logic 0 — A voltage level approximately equal to the ground voltage (V_{SS}).

low byte — The least significant eight bits of a word.

M68HC12 — A Motorola family of 16-bit MCUs.

mark/space — The logic 1/logic 0 convention used in formatting data in serial communication.

mask — 1. A logic circuit that forces a bit or group of bits to a desired state. 2. A photomask used in integrated circuit fabrication to transfer an image onto silicon.

MCU — Microcontroller unit. See “microcontroller.”

memory location — Each M68HC12 memory location holds one byte of data and has a unique address. To store information in a memory location, the CPU places the address of the location on the address bus, the data information on the data bus, and asserts the write signal. To read information from a memory location, the CPU places the address of the location on the address bus and asserts the read signal. In response to the read signal, the selected memory location places its data onto the data bus.

memory map — A pictorial representation of all memory locations in a computer system.

MI-Bus — See "Motorola interconnect bus".

microcontroller — Microcontroller unit (MCU). A complete computer system, including a CPU, memory, a clock oscillator, and input/output (I/O) on a single integrated circuit.

modulo counter — A counter that can be programmed to count to any number from zero to its maximum possible modulus.

most significant bit (MSB) — The leftmost digit of a binary number.

Motorola interconnect bus (MI-Bus) — The Motorola Interconnect Bus (MI Bus) is a serial communications protocol which supports distributed real-time control efficiently and with a high degree of noise immunity.

Motorola scalable CAN (msCAN) — The Motorola scalable controller area network is a serial communications protocol that efficiently supports distributed real-time control with a very high level of data integrity.

msCAN — See "Motorola scalable CAN".

MSI — See "multiple serial interface".

multiple serial interface — A module consisting of multiple independent serial I/O sub-systems, e.g. two SCI and one SPI.

multiplexer — A device that can select one of a number of inputs and pass the logic level of that input on to the output.

nibble — A set of four bits (half of a byte).

object code — The output from an assembler or compiler that is itself executable machine code, or is suitable for processing to produce executable machine code.

opcode — A binary code that instructs the CPU to perform an operation.

open-drain — An output that has no pullup transistor. An external pullup device can be connected to the power supply to provide the logic 1 output voltage.

operand — Data on which an operation is performed. Usually a statement consists of an operator and an operand. For example, the operator may be an add instruction, and the operand may be the quantity to be added.

oscillator — A circuit that produces a constant frequency square wave that is used by the computer as a timing and sequencing reference.

OTPROM — One-time programmable read-only memory. A nonvolatile type of memory that cannot be reprogrammed.

overflow — A quantity that is too large to be contained in one byte or one word.

page zero — The first 256 bytes of memory (addresses $0000–$00FF).

parity — An error-checking scheme that counts the number of logic 1s in each byte transmitted. In a system that uses odd parity, every byte is expected to have an odd number of logic 1s. In an even parity system, every byte should have an even number of logic 1s. In the transmitter, a parity generator appends an extra bit to each byte to make the number of logic 1s odd for odd parity or even for even parity. A parity checker in the receiver counts the number of logic 1s in each byte. The parity checker generates an error signal if it finds a byte with an incorrect number of logic 1s.

PC — See “program counter (PC).”

peripheral — A circuit not under direct CPU control.

phase-locked loop (PLL) — A clock generator circuit in which a voltage controlled oscillator produces an oscillation which is synchronized to a reference signal.

PLL — See "phase-locked loop (PLL)."

pointer — Pointer register. An index register is sometimes called a pointer register because its contents are used in the calculation of the address of an operand, and therefore points to the operand.

polarity — The two opposite logic levels, logic 1 and logic 0, which correspond to two different voltage levels, V_{DD} and V_{SS}.

polling — Periodically reading a status bit to monitor the condition of a peripheral device.

port — A set of wires for communicating with off-chip devices.

prescaler — A circuit that generates an output signal related to the input signal by a fractional scale factor such as 1/2, 1/8, 1/10 etc.

program — A set of computer instructions that cause a computer to perform a desired operation or operations.

program counter (PC) — A 16-bit register in the CPU. The PC register holds the address of the next instruction or operand that the CPU will use.

pull — An instruction that copies into the accumulator the contents of a stack RAM location. The stack RAM address is in the stack pointer.

pullup — A transistor in the output of a logic gate that connects the output to the logic 1 voltage of the power supply.

pulse-width — The amount of time a signal is on as opposed to being in its off state.

pulse-width modulation (PWM) — Controlled variation (modulation) of the pulse width of a signal with a constant frequency.

push — An instruction that copies the contents of the accumulator to the stack RAM. The stack RAM address is in the stack pointer.

PWM period — The time required for one complete cycle of a PWM waveform.

RAM — Random access memory. All RAM locations can be read or written by the CPU. The contents of a RAM memory location remain valid until the CPU writes a different value or until power is turned off.

RC circuit — A circuit consisting of capacitors and resistors having a defined time constant.

read — To copy the contents of a memory location to the accumulator.

register — A circuit that stores a group of bits.

reserved memory location — A memory location that is used only in special factory test modes. Writing to a reserved location has no effect. Reading a reserved location returns an unpredictable value.

reset — To force a device to a known condition.

SCI — See "serial communication interface module (SCI)."

serial — Pertaining to sequential transmission over a single line.

serial communications interface module (SCI) — A module that supports asynchronous communication.

serial peripheral interface module (SPI) — A module that supports synchronous communication.

set — To change a bit from logic 0 to logic 1; opposite of clear.

shift register — A chain of circuits that can retain the logic levels (logic 1 or logic 0) written to them and that can shift the logic levels to the right or left through adjacent circuits in the chain.

signed — A binary number notation that accommodates both positive and negative numbers. The most significant bit is used to indicate whether the number is positive or negative, normally logic 0 for positive and logic 1 for negative. The other seven bits indicate the magnitude of the number.

software — Instructions and data that control the operation of a microcontroller.

software interrupt (SWI) — An instruction that causes an interrupt and its associated vector fetch.

SPI — See "serial peripheral interface module (SPI)."

stack — A portion of RAM reserved for storage of CPU register contents and subroutine return addresses.

stack pointer (SP) — A 16-bit register in the CPU containing the address of the next available storage location on the stack.

start bit — A bit that signals the beginning of an asynchronous serial transmission.

status bit — A register bit that indicates the condition of a device.

stop bit — A bit that signals the end of an asynchronous serial transmission.

subroutine — A sequence of instructions to be used more than once in the course of a program. The last instruction in a subroutine is a return from subroutine (RTS) instruction. At each place in the main program where the subroutine instructions are needed, a jump or branch to subroutine (JSR or BSR) instruction is used to call the subroutine. The CPU leaves the flow of the main program to execute the instructions in the subroutine. When the RTS instruction is executed, the CPU returns to the main program where it left off.

synchronous — Refers to logic circuits and operations that are synchronized by a common reference signal.

timer — A module used to relate events in a system to a point in time.

toggle — To change the state of an output from a logic 0 to a logic 1 or from a logic 1 to a logic 0.

tracking mode — A mode of PLL operation with narrow loop bandwidth. Also see ‘acquisition mode.’

two's complement — A means of performing binary subtraction using addition techniques. The most significant bit of a two's complement number indicates the sign of the number (1 indicates negative). The two's complement negative of a number is obtained by inverting each bit in the number and then adding 1 to the result.

unbuffered — Utilizes only one register for data; new data overwrites current data.

unimplemented memory location — A memory location that is not used. Writing to an unimplemented location has no effect. Reading an unimplemented location returns an unpredictable value.

variable — A value that changes during the course of program execution.

VCO — See "voltage-controlled oscillator."

vector — A memory location that contains the address of the beginning of a subroutine written to service an interrupt or reset.

voltage-controlled oscillator (VCO) — A circuit that produces an oscillating output signal of a frequency that is controlled by a dc voltage applied to a control input.

waveform — A graphical representation in which the amplitude of a wave is plotted against time.

wired-OR — Connection of circuit outputs so that if any output is high, the connection point is high.

word — A set of two bytes (16 bits).

write — The transfer of a byte of data from the CPU to a memory location.

HOW TO REACH US:

USA/EUROPE/LOCATIONS NOT LISTED:

Motorola Literature Distribution;
P.O. Box 5405, Denver, Colorado 80217
1-303-675-2140 or 1-800-441-2447

JAPAN:

Motorola Japan Ltd.; SPS, Technical Information Center,
3-20-1, Minami-Azabu Minato-ku, Tokyo 106-8573 Japan
81-3-3440-3569

ASIA/PACIFIC:

Motorola Semiconductors H.K. Ltd.;
Silicon Harbour Centre, 2 Dai King Street,
Tai Po Industrial Estate, Tai Po, N.T., Hong Kong
852-26668334

TECHNICAL INFORMATION CENTER:

1-800-521-6274

HOME PAGE:

http://www.motorola.com/semiconductors

MC68HC912D60A/D