

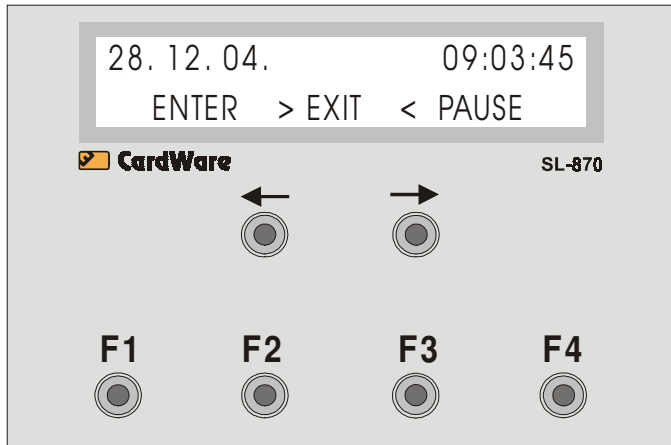
SL-870: AUX ID module with LCD and keypad

I

Technical description

0. General system overview

The SL-870 is an ID module with an LCD (Liquid Crystal Display) and keys for the selection of activity types. There are currently two versions, one for reading iButtons and one for RF cards. When an ID badge (iButton or RF card) is read, its code is stored in the SL-870's buffer, along with the code of the selected activity (Event Code or EC). This buffer is read out by the SL-8x terminal and a registration is made with a time stamp according to the SL-8x's clock. The EC values of the selectable activity types in the SL-870 are between 4 and 14. A user-configurable 6-character mnemonic can be assigned to each activity.



The mnemonics can be changed in-system, i.e. while the SL-870 is connected to an SL-8x terminal. Some possibilities include: LncBrk, BsnsTr, CofBrk etc.

Fig. 1

The activity mnemonics are displayed on the lower line of the LCD. The currently active one (the EC of which will be stored in case of a registration) is in the centre. The left and right arrow keys let the user move around the selected activities one by one. The shortcut keys (F1, F2, F3 i F4) can be assigned to any four activities. Pressing a shortcut key will put the associated activity in the centre, along with the preceding and succeeding activities, which are shown to the left and right.

The mnemonics and the four shortcut keys are configured with the AuxCnf program and its SL-870 plug-in.

The SL-870 has a two-row display. The top row shows the time and date according to the master SL-8x terminal's clock.

When a registration is made, the code of the read ID badge is shown in the upper row for a few seconds instead of the time and a short beep is heard. The second row of the display shows the previous, the current and the following activity's mnemonics.

There are six keys in total, two above, for moving around the activities (<- , ->), and four below, for the shortcuts (F1, F2, F3 i F4).

Fig. 1 shows the front panel of the SL-870. The case dimensions are: 200x152x62mm.

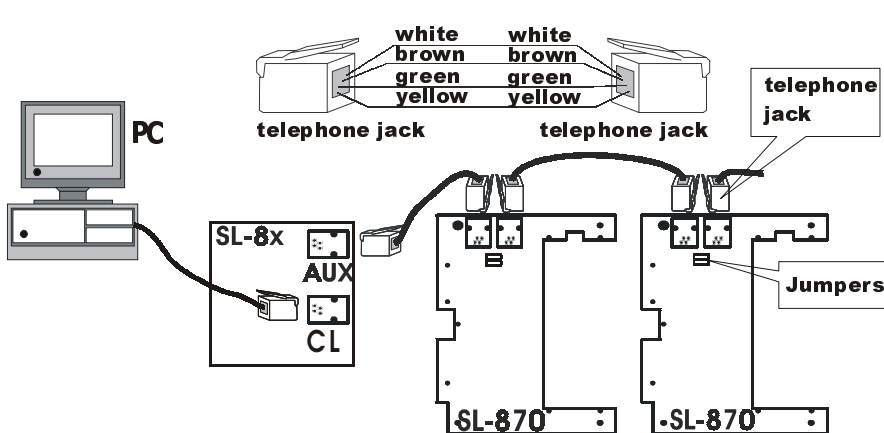


Fig. 2

Connection of the SL-870 to an SL-8x terminal is via the AUX connector, which is a 4 pin telephone jack (RJ-11). The 4-wire flat telephone cable carries both signals and power.

Up to 4 SL-870 units can be connected in series onto an SL-8x terminal. The addresses are from 30h do 33h. The last AUX device in the chain must have the two pull-

up resistor jumpers set, as shown in Fig. 2.

The general overview of the system is shown in fig. 2.

The SL-870 units are configured with the AuxCnf.exe program on the PC, which is attached to the SL-8x terminal.

1. HARDWARE

Fig. 3 shows the SL-870 PCB.

Board dimensions are: 175x125mm.

Noteworthy are the following:

- **DIP SW**

Used for setting the address offset of the unit. This is set with SW3 and SW4. The base address for the SL-870 is 30h, so the range of addresses is 30h to 33h. (fig. 4).

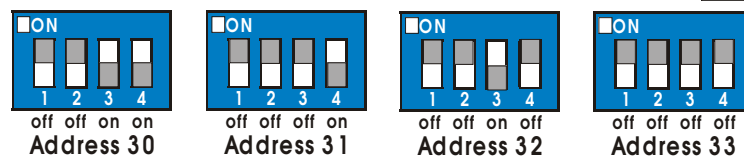
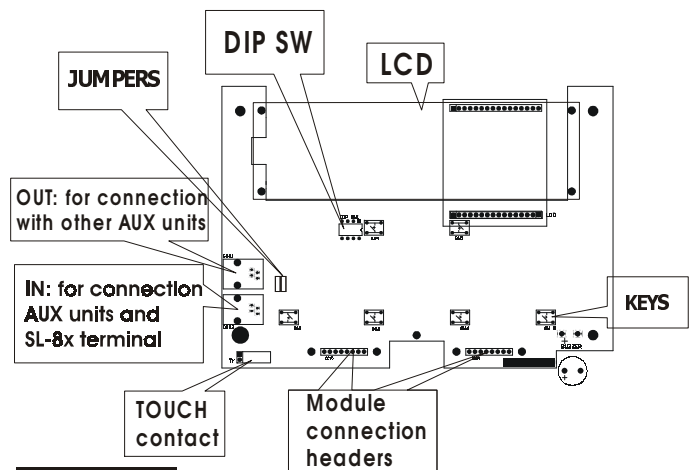


Fig. 4



Fifig. 3

- **LCD**

The SL-870 has a 20 character by two line LCD with LED backlight. The top row of the LCD shows the time and date according to the SL-8x and briefly, when a registration is made, the code of the ID badge is shown instead of the time and date. The lower row of the display shows the activity mnemonics: the previous, the currently active and the following.

After power is applied to the SL-870, the display shows the current firmware revision as well as the selected address of the unit. This is shown for a few seconds only.

If the date and time aren't sent by the SL-8x terminal (Function 00) for more than 5-6 seconds, the display starts blinking.

- **Keys**

The SL-870 has 6 keys, two in the upper row and four in the lower row.

The two keys in the upper row (with right and left arrow symbols) are used to move the currently selected (the one in the middle) activity mnemonic to left or to the right by one place. The four keys in the lower row are used to select four preconfigured activities by a single keystroke. These are the shortcut keys.

Pressing a key will cause a very short beep to be heard.

- **Two telephone 6/4 (RJ-11) jacks**

In: for connection to other AUX units towards the SL-8x terminal or to the SL-8x terminal itself.

Out: for connection to other AUX units away from the SL-8x. The last AUX unit in series must have its jumpers set.

- **TOUCH contact**

Used for connecting the iButton contact cup

- **Module connection headers**

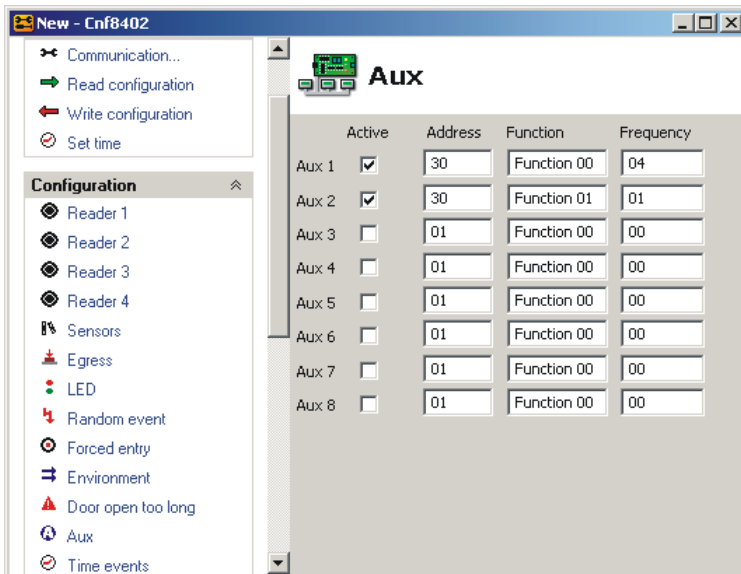
For attaching various daughter boards to the SL-870 (e.g. RF card reader module).

- **Jumpers**

Up to 8 AUX units can be chained together in series. The last (and only the last) one in the series must have these jumpers shorted, as shown in fig. 2.

2. Configuration

2.1. Configuration of the SL-8x terminal (Cnf 8302.exe or Cnf8402.exe)



The AUX tab of the Cnf8x02.exe program is shown in fig.5. The device address has to be set for each row that defines a cyclic function i.e. one that will be called at regular intervals. There is a checkbox for enabling or disabling a configured cyclic function. Function numbers 0 to 7 are possible and depend on the type of AUX unit, which is defined by the address..The frequency column is for setting the frequency with which the cyclic function will be called. The smaller the number, the higher the frequency.

In this example an SL-870 with address 30h has been configured for both available functions, 00h and 1h.

Fig. 5

- Function 00, with a frequency of 04h.

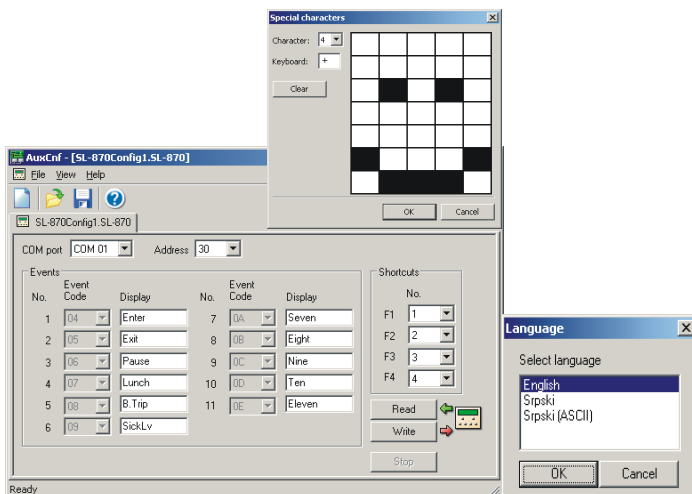
This function causes the SL-8x to send the current time & date to the SL-870, so that this may be displayed in the top row of the LCD. This data can be sent with a relatively low frequency, since the SL-870 can increment seconds, minutes and hours by itself.

- Function 01, with a frequency of 01h.

The SL-8x terminal reads the contents of the SL-870's buffer, which is filled with the code of the ID badge and the selected activity's event code (EC) when a registration occurs.

2.2. Configuration of the SL- 870 (AuxCnf.exe)

The AuxCnf program is used to configure all configurable AUX devices connected to SL-8x terminals in the system.



You have to set the COM port to which the SL-8x is attached and the address of the desired AUX device. New configurable AUX devices will have their own specific plugins. The SL-870's plugin is used to define the mnemonics, the 8 user-defined characters and the shortcut keys.

The desired mnemonics are entered into the *mnemonic* fields. Clicking the *Write* button will write the mnemonic fields into the EEPROM of the SL-870.

Fig. 6 shows the AuxCnf program's SL-870 plugin.

Fig. 6

Shortcuts are defined by assigning the index of the desired activity, the mnemonic of which will be in the middle, to each one of the four shortcut keys (F1, F2, F3 i F4).

The language of the AuxCnf program can be changed in View\ Language.

It is possible to create 8 user-defined characters in View\ Special characters. This is shown in fig. 7.

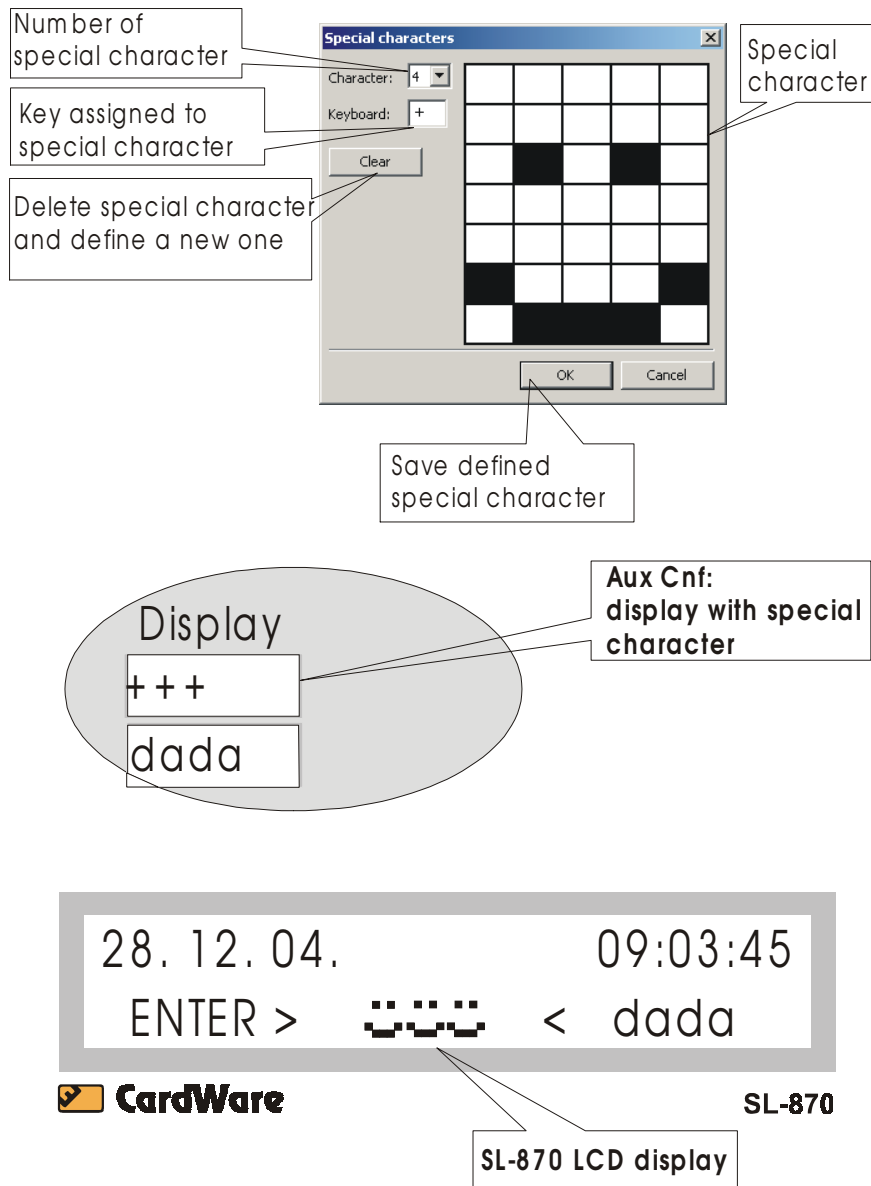


Fig. 7

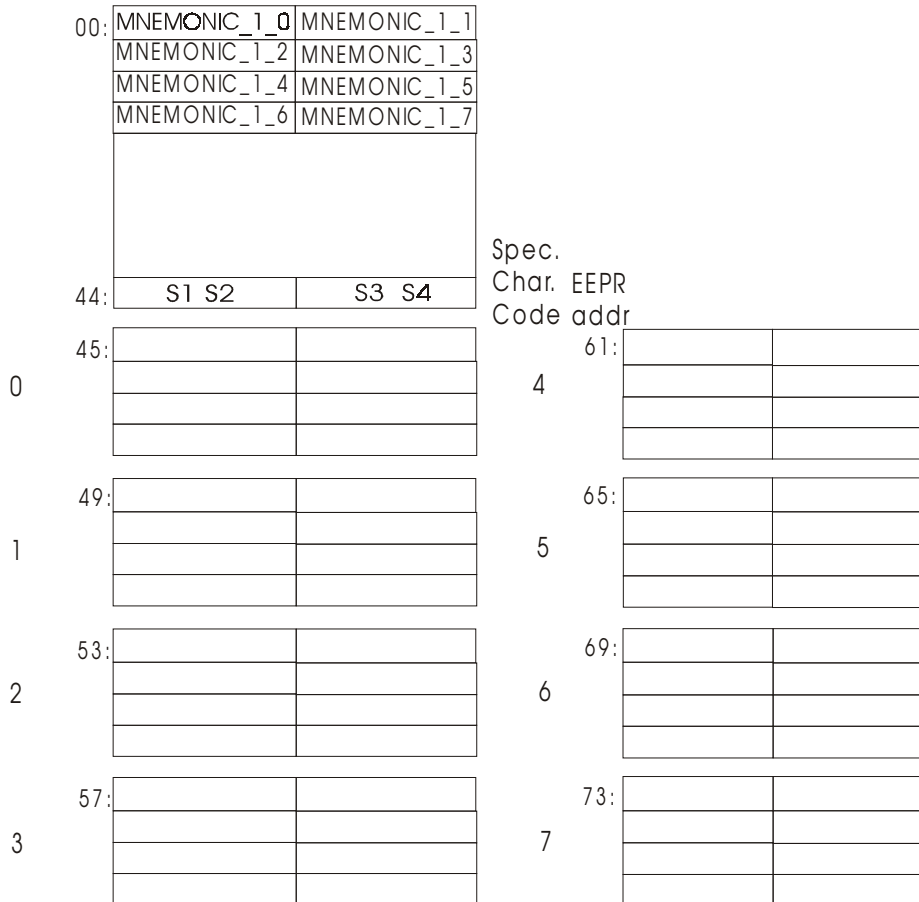
5.3.1.3 Memory map of the EEPROM

The SL-870 uses a 93c66 EEPROM, which has 128 16-bit words, or 256 bytes. Fig. 8 shows the memory map.

Spec.

Char. EEPR

Code addr



memory map

0-7 Special character codes

Fig. 8

EEPROM locations 00h to 43h contain the 11 mnemonics. Each mnemonic takes up 8 bytes or 4 16-bit words. The bytes are:

MNEMONIC_1_0 - MNEMONIC_1_5 ASCII codes of the 6 characters of the mnemonic,

MNEMONIC_1_6 EC (event code of the activity),

MNEMONIC_1_7 CRC (cyclic redundancy code) of the previous 7 bytes.

EEPROM address 44h contains the shortcut definitions as four 4-bit nibbles S1, S2, S3, S4.

S1: index of the activity mnemonic to jump to when F1 is pressed

S2: index of the activity mnemonic to jump to when F2 is pressed

S3: index of the activity mnemonic to jump to when F3 is pressed

S4: index of the activity mnemonic to jump to when F4 is pressed

The bitmaps of the user-defined characters are stored from address 45 to 76.

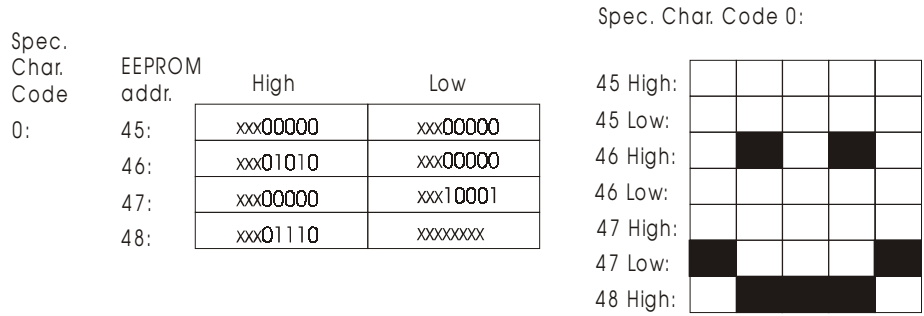


Fig. 9

The user defined characters have codes from 0 to 7. The bitmap of the character with code 0 will be stored in addresses 45 to 48, code 1 in EEPROM addresses 49 to 52 etc.

Fig. 9 shows how the bitmaps stored in the EEPROM are calculated for a smiley at code location 0. This character is stored in the EEPROM from word 45 to word 48. Each character is made up of 7 rows with 5 pixels each. Each row is represented by one byte (half a word) in the EEPROM. The first three bits are unused, the other 5 define the pixels.

5. PROTOCOL

The AUX interface is a four-wire bidirectional connection method for linking peripheral units with an SL-8x terminal. The protocol is very similar to I2C, but the voltage is 12V and there are two additional wires for power:

The SL-8x terminals' firmware has two types of support for AUX units: cyclic functions and commands.

5.1. Cyclic functions

Cyclic functions are executed periodically, after a preset number of cycles of the terminal's main loop. Some AUX units have only one executable function, while others have several.

The SL-870 can execute two functions, both of which have to be configured (with Cnf8x02.exe)

Function 00: sending the time & date to the SL-870 to be displayed

Function 01: reading the SL-870's buffer, i.e. reading registration data from the SL-870

5.2. Commands

Commands are sent from the PC, usually as a result of human intervention, and are executed only once.

The PC sends a command to the SL-8x, which puts it into a command buffer, and executes it (sends it to the addressed AUX unit) at the next available cycle, ahead of any cyclic functions that may be due.

If the command is of a type that expects a response from the AUX unit, the AUX unit's buffer will be read in the following cycle, delaying any cyclic functions for another cycle.

5.3. AUX protocol

The AUX bus has READ and WRITE cycles similar to the I2C bus.

When the R/W bit is 0, it's a Write cycle, when it's 1 it's a read cycle.

The Write cycle is shown in fig. 9. The master (SL-8x) is writing into the slave's (AUX unit) buffer.

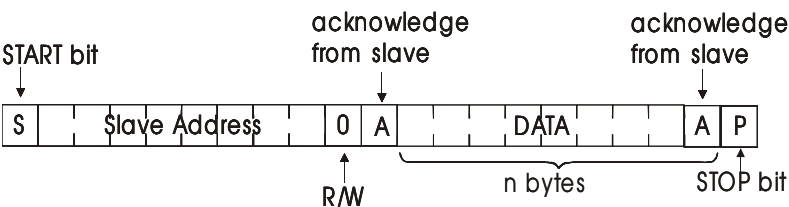


Fig. 9

Fig. 10 shows a Read cycle. The master is reading from the slave's buffer right after the first byte.

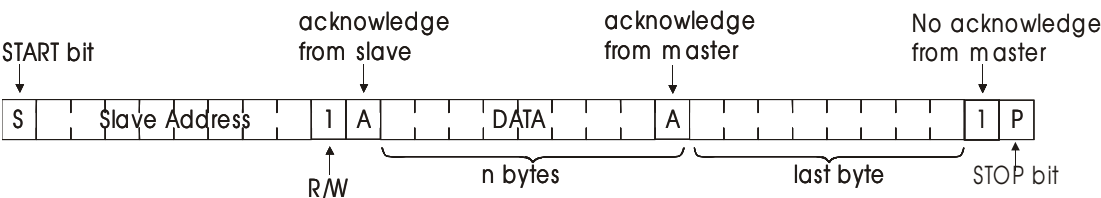


Fig.

5.3.1.3 AUX write commands

The AUX write cycle is used by the master (SL-8x) to send commands and data to the AUX unit. Each AUX unit recognises a few types of commands that precede the data it receives.

The SL-870 has the following command set:

5.3.1.3 'T' – update time & date buffer

The SL-8x sends the following data to the SL-870's receive buffer:

- 'T'
- 6 bytes, which contain the current time and date.

Fig.11 shows the command:

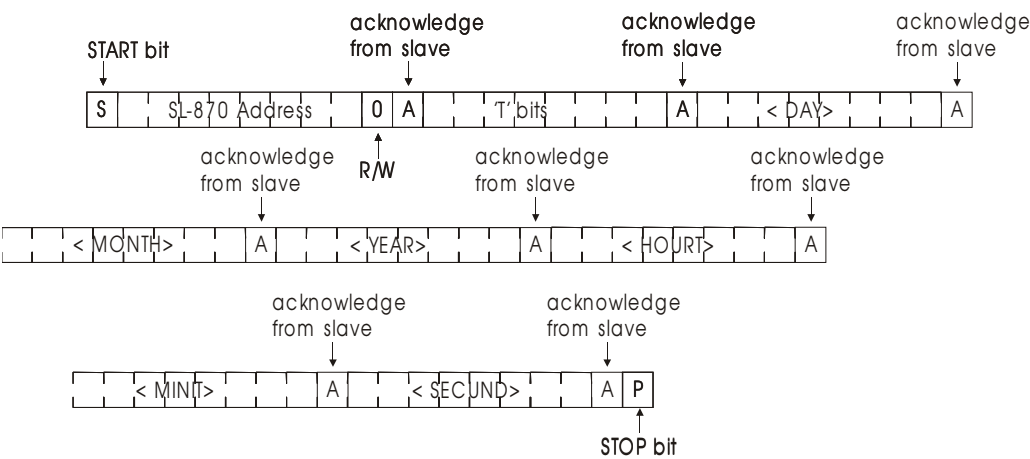


Fig. 11
5.3.1.3 'W' – write to

EEPROM

The SL-8x sends the following data to the SL-870's receive buffer:

- 'W'
- address of the 16-bit word to write to
- <DATAH><DATA L>: 2 bytes of data (high and low byte) to be written.

The command is shown in fig. 12.

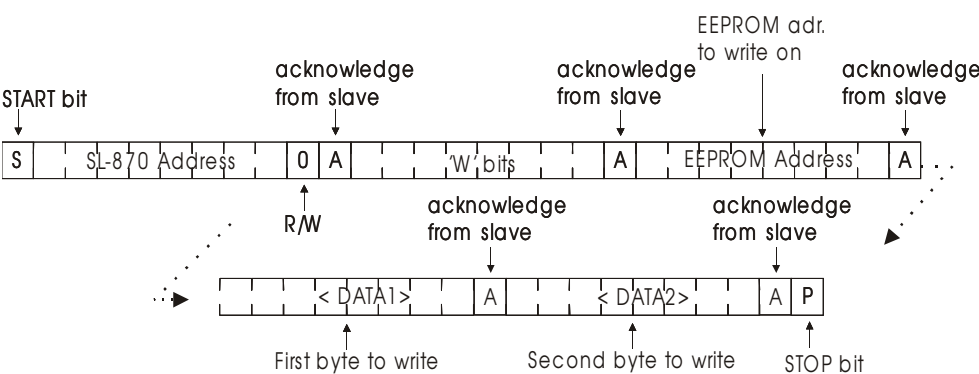


Fig. 12

5.3.1.3 'R' – read from EEPROM

The SL-8x sends the following data to the SL-870's receive buffer:

- 'R'
- address of the EEPROM from which to read.

The cycle is shown in fig. 13.

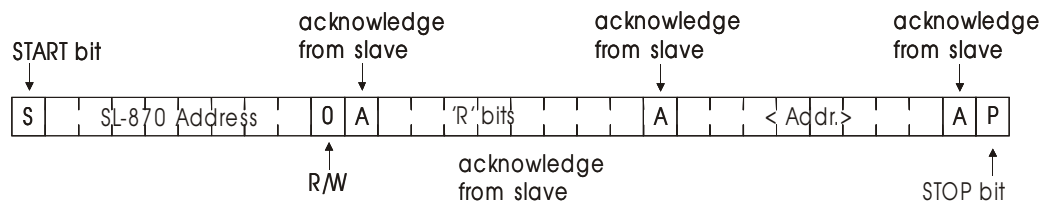


Fig. 13

After receiving this command, the SL-870 reads the given EEPROM address and stores the result in the transmit buffer, ready for a read cycle from the master (SL-8x) which will enable it to send the requested data.

In the next cycle of the SL-8x main firmware loop, the SL-8x reads the SL-870 buffer with an AUX read cycle. This is shown in fig. 14.

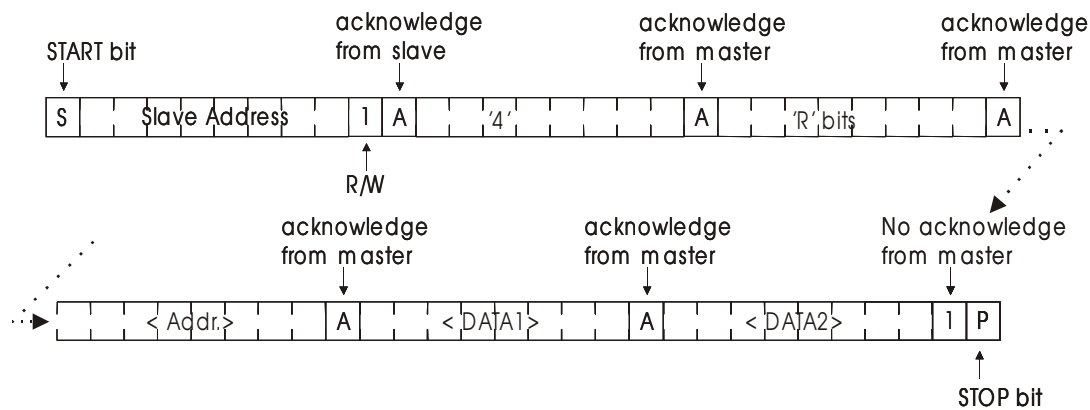


Fig. 14

Five bytes are sent:

- 4 the total number of bytes being sent,
- 'R' the original command
- <Addr> EEPROM address which has been read
- <DATA1><DATA2> 2 bytes of data read from the EEPROM