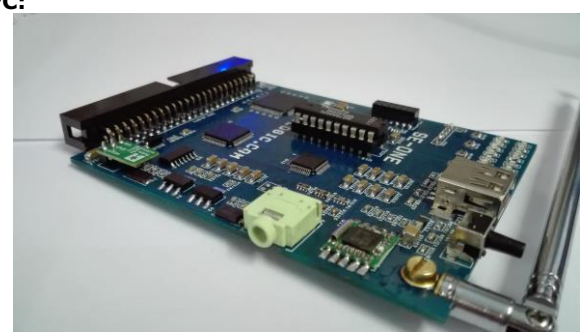
# SE-ONE

**MSX MP3 Player, FM receiver, USB host**

**MSX:**


**CPC:**



**Description:**

The SE-ONE is an extension cartridge for the MSX and CPC home computers. You can play mp3 music files on your home computer.
The SE-ONE has a lot of modes. This mode and another option can be set and read with a user-friendly AT command set.
Mode MP3A is the Sunrise compatible mode, there is IO compatibility with the Sunrise mp3 cartridge.
The FM mode turns the SE-ONE into a FM radio. European and a US Japan bands are available.

**Features:**

- Fully user-friendly AT command set
- MP3 chip VS1053 decodes multiple formats:
  - MP3 = MPEG 1 & 2 audio layer III (CBR+VBR+ABR)
  - Ogg Vorbis
- Mosty compatible with the Sunrise MP3 player
- Audio spectrum analyzer
- Player supporting:
  - Symbos Symamp
  - Sunrise software PLAYINFO
  - Sunrise software PLAYMP3
- FM stereo radio:
  - European, Japan and US FM bands
  - FM mixer for conversion to IF of the US/Europe (87.5 MHz to 108 MHz) and Japanese (76 MHz to 91 MHz) FM band. RF Automatic Gain Control (AGC) circuit
- Onboard DSP Hi-fi stereo audio processor
  - Source and mode selector for two stereo channels
  - Pseudo stereo, spatial stereo, linear stereo and forced mono switch
  - Volume and balance control
  - Bass, treble and mute control
- Stereo VU LED bar
  - Two bytes parallel
  - Software controlled
- MSX input audio switch
  - Software controlled
- System updatable with a DFU cable and pc software
- Used IO ports for MSX :0x20-0x27
- User IO ports for CPC: &FF20-&FF27
- Busdir, Wait, Int is connected.
- +5v + 12v
- The main controller is an ARM® 32-bit Cortex®-M4 CPU with FPU, Adaptive real-time accelerator (ART Accelerator™) allowing 0-wait state execution from Flash memory, frequency up to 168 MHz, memory protection unit, 210 DMIPS/ 1.25 DMIPS/MHz (Dhrystone 2.1), and DSP instructions
- Up to 1 Mbyte of Flash memory
- USB 2.0 full-speed device/host/OTG controller
- Up to 192+4 Kbytes of SRAM including 64- Kbyte of CCM (core coupled memory) data RAM

**Applications:**

- FM: SymbOs Radio (Edoz)
- FM: Basic Radio(Hans)
- AT CMD: SEone.com (Paul)
- AT CMD: AT.LDR (Hans T)
- MP3: Playinfo(Sunrise)
- MP3: Playmp3.com (Sunrise)

## Inhoud

# 1.0 Overview

## 1.1 About us

Two years ago we started again to develop products for the MSX C64 CPC and Enterprice home computers, using the name TMTLOGIC.
Special thanks go to the following hobbyists/programmers:

- EdoZ. (SymbOS) http://members.home.nl/evanzanten/appstore/
- Paul B.(MSX / C) https://github.com/PaulBoss/seone/releases
- Robbert-Jan B. (MSX / TEST),
- Rinus S (webshop), http://www.msx-shop.nl/winkel/
- Emil S (MSX hardware),
- Hans O (MSX hardware),
- MVM(MSX computerclub), http://www.m-v-m.nl/
- Frits H.(C controllers) http://nlmsx.generation-msx.nl/
- Frits K, (C64) Jan D.(C64),
- Erik L (cartridges)
- Leendert N.(C/C#),
- Marco vL.(C# Tmtnet server),
- Gideon Z(FPGA) http://www.1541ultimate.net
- Petra (manual),
- Dick T.(manual),
- Libbe R.(PHP/mySQL)
- Kebu synthesizer music http://kebu.fi
- Jorn P.(SymbOS /CPC) http://www.prodatron.net

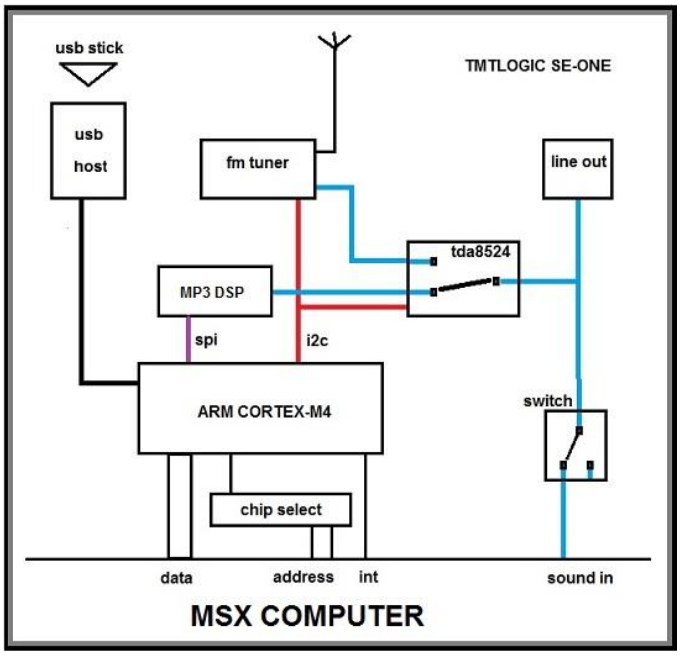
Also a word of thank to the Sunrise Foundation for the clear manual / documentation of the MP3 cartridge. This has contributed to the creation of the SE ONE, as it is now.

## 1.2 Discription of the SE-ONE

The SE-ONE hardware is simple in its setup. The cartridge responses to IO-addresses &h20-&h27
This is the same IO series as the SUNRISE MSXMP3 player.
The SE-ONE has multiple modes:

- MP3A mode: the IO is identical to the SUNRISE MP3 player, there are differences in the software because the SE-ONE has another chipset and all of the functions convert to this chipset.
- MP3B mode: this mode is an beta version , you can play music files from USB stick, and the MP3 chip is use with at commands.
- FM mode: The SE-ONE is an FM Radio

The heart of the SE-ONE is a powerful ARM microcontroller STM32F4. This controller arranges all data streams on the SE-ONE cartridge.
Besides that, it is possible to turn the sound going to the MSX computer on and off.




*Basic architecture SE-One*

The USB host will soon have more possibilities. A USB stick (Mass storage device) can be developed for playing MP3 files.

You can update the Firmware yourself with the included "DFU-cable" and software for the PC
You can find the software and DFU-files for the SE-ONE on the website of www.tmtlogic.com tab support

⚠ CAUTION !!
Only use the "DFU cable" for updating the SE-ONE. Not for anything else. This can cause serious damage.

## 1.3 AT commando set

### 1.3.1 What is an AT-command set?

The protocol of the AT-command set has been used for a long time. AT-commands exists of a short string of text with added settings or parameters. You can also request data from the hardware with the AT-commands. You need a small program to use AT-commands that sends the string of text to the SE-ONE and processes the response of the SE-ONE. They are working on a machine language program that will support the MSX CALL. Unfortunately there is no intern ROM present to save these Call commands.

Pros and cons of the AT-commands.

Normally we use registers. For example, if we want to put a character on the screen we express it like: OUT &h98,65. This works fast as it approaches the VDP chip directly. If we use a different VDP-chip of another brand, OUT &h98,65 doesn't work! New software has to be made for this new chip.

With the SE-ONE the hardware can be changed without the need of writing new software. For example, a sound processor (TDA8425) is included in the SE-ONE. This chip is quite old, in due course it will have to be replaced by a modern DSP chip. This new DSP chip contains other registers than the tda8425.

How can you fix that?.., with AT commands.

With sending AT+DSPVOLL=80 to IO port OUT &h20, the volume of the left channel is set.

The ARM microcontroller on the SE-ONE reads the AT string and knows how and which data needs to be send to the TDA8425. ARM also checks whether a different DSP exists.

AT command is clear, OUT &H99,65 cannot be understood well without using this manual.

A possible disadvantage can be that a program can be needed that handles the AT command. Another disadvantage can be that the speed is lower.

When using this device, the next values can be received: OK,ERR,ILL or BUSY

| | |
|---|---|
| OKe | means that everything is processed well |
| ERRor | means that there has been made a typo or that you typed a wrong value. |
| ILLeg*al error* | means that the SE-ONE does not recognize the AT instruction |
| BUSY | means that the response is taking longer than you are used to. The function needs more time |

This is the way the AT command procedure works:

<CR> = CHR$(13)

<LF> = CHR$(10)

<NULL>= CHR$(0)

1. the AT command will be send character by character to OUT &h20
2. send the <CR>)

Wait for the SE-ONE to perform the AT-command and responses.

3. read data of INP(&h20)
   when <NULL> there is no data, goto point 3
4. When this is not <NULL>, read the response in a buffer until you come across <LF>, this means you reached the end of the response
5. Analyze the buffer now

It is wise to take a timeout for the response string

This procedure works as follows:

1. The AT command will be send character by character to out &h20
2. send the <CR>

Wait for the SE-ONE to perform the AT-command and responses.

3. set the timeout counter on value
4. read the data of INP(&h20) and lower the timeout counter
   When the timeout counter is null goto the Error handler.
   read the response in a buffer until you come across <LF>, this means you reached the end of the response, goto point 5
   if not <LF> go back to point 4
5. Analyze the buffer now

Send (AT+SEMODE?)

| A | T | + | S | E | M | O | D | E | ? | <CR> |
|---|---|---|---|---|---|---|---|---|---|---|
| 65 | 84 | 43 | 83 | 69 | 77 | 79 | 68 | 69 | 63 | 13 |

Response:

When the processor is busy it displays many zero bytes chr$(0)

| 0 | 0 | 0 | 0 | 0 | .... |
|---|---|---|---|---|---|

| A | T | + | S | E | M | O | D | E | = | F | M | <CR> | O | K | <CR> | <LF> |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

No more information is available after response when &h20 data is CHR$(0) (zero)

Send (AT+FMSCANUP)

| A | T | + | F | M | S | C | A | N | U | P | <CR> |
|---|---|---|---|---|---|---|---|---|---|---|---|

Response:

If the processor is busy it displays many zero bytes <NULL>

0 0 0 0 0 ....

O K <CR> <LF>

### 1.3.2 AT quick reference help option:

For quick reference help with AT commands, you can also use the /H after a command.
For example: AT+SEMODE/H

## 1.4 Under construction

An expansion of the USB host port is under construction, including a possibility to connect a USB stick which can be used to play music files. Only use with mode MP3B

## 1.5 Safety

### 1.5.1 Safety points:

#### *1.5.1.1 Plugging in/out MSX computer:*

Only insert the SE-ONE in the MSX while this is turned off and is voltage-free! !
This also applies to when you unplug the SE-ONE from the MSX.

#### *1.5.1.2 Antenna:*

Carefully move the antenna and keep a firm hold on the cartridge.

#### *1.5.1.3 ESD:*

ESD means:
electrostatic-sensitive device
An electrostatic-sensitive device (often abbreviated ESD) is any component (primarily electrical) which can be damaged by common static charges which build up on people, tools, and other non-conductors or semiconductors. ESD commonly also stands for electrostatic discharge.Be careful with the ESD. Don't touch the SE-ONE print or the contacts of the USB connection. This can cause damage. (Wikipedia)

Caution with ESD !! do not touch the SE-ONE circuit board contacts or the contacts of the USB connector. Without good earth. This can cause damage !!

#### *USB:*

Never use the USB portal to charge your phone or other power using devices !!

The maximum current is 300mA. The print entry points will burn if you do use too much current of the SE-ONE !!.

The power supply of your MSX computer will be heavily charged. The power supply of the MSX can possibly be damaged !!

The USB portal is also used to update Firmware.

This DFU update can only take place if the SE-ONE is unplugged from the MSX !!

The manual for these updates is on the website tmtlogic.com > support > SE-ONE

#### 1.5.1.4 USB Stick:

After plugging the USB stick in or out, the SE-ONE must be reset first. This is a familiar bug. We will solve this problem as soon as possible.

## 1.6 Warranty and Liability

### 1.6.1 Warranty

The warranty of the SE-ONE is 6 months, provided that it is used properly as written in this manual

Covered by the warranty is:

- Damage during transportation
- hardware faults that have appeared during production
- defective Dfu cable

Not covered by warranty:

- Breakdown of the antenna
- Defective aux out plug
- Defective USB connector
- Changes or attempted repairs to the device, or unauthorised modification of the circuitry
- Any other damage, including by improper use (e.g. placing or removing the SE-One while the MSX computer is turned on and power on the cartridge is locked)

### 1.6.2 Liability

Tmtlogic / Hans Tillema and all other people who have cooperated on this project do not accept any liability for the damage that may have appeared during the use of the SE-ONE.

# 2.0 AT command reference

## 2.1 CARTRIDGE

---

### 2.1.1 AT+CARTRIDGE

| | |
|---|---|
| SEONE/M | only MP3 |
| SEONE/R | only Radio |
| SEONE/MR | MP3 and Radio |

#### *Get cartridge type*

| | | |
|---|---|---|
| send: | AT+CARTRIDGE?<13> | |
| response: | AT+CARTRIDGE=SEONE/MR<13><10> | options: SEONE/M,SEONE/R,SEONE/MR |
| | OK<13><10> | |

### 2.1.2 AT+LIST

Get the SE-ONE AT command's list

| | | |
|---|---|---|
| send: | AT+LIST<13> | |
| response: | AT+LIST= | |
| | AT+CARTRIDGE<13> | |
| | AT+SEMODE<13> | |
| | AT+SELOGBOOK <13> | |
| | ….. | |
| | OK<13><10> | options: OK,ERR |

## 2.2 SE

---

### 2.2.1 AT+SEMODE

| | |
|---|---|
| FM | = FM tuner |

MP3A = Sunrise MSXMP3
MP3B = TMTLOGIC MP3

#### *Change mode from SE-ONE*

| | | |
|---|---|---|
| send: | AT+SEMODE=FM<13> | options: FM,MP3A,MP3B |
| response: | OK<13><10> | options: OK,ERR,ILL |

#### *Get SE-ONE mode*

| | | |
|---|---|---|
| send: | AT+SEMODE?<13> | |
| response: | AT+SEMODE=FM<13> | options: FM,MP3A,MP3B |
| | OK<13><10> | options: OK,ERR,ILL |

### 2.2.2 AT+SELOGBOOK

#### *Get SE-ONE system logbook*

| | | |
|---|---|---|
| send: | AT+SELOGBOOK?<13> | |
| response: | AT+SELOGBOOK=TEXT<13> | |
| | OK<13><10> | options: OK,ERR,ILL |

### 2.2.3 AT+SEVERSION

versions 1 = Radio
2 = MP3
3 = Radio and MP3

#### *Get SE-ONE system SE-ONE version number*

| | | |
|---|---|---|
| send: | AT+SEVERSION?<13> | |
| response: | AT+SEVERSION=1<13> | options: 1,2,3 |
| | OK<13><10> | options: OK,ERR,ILL |

### 2.2.4 AT+SEFIRMWARE

#### *Get firmware dfu file name*

| | | |
|---|---|---|
| send: | AT+SEFIRMWARE?<13> | |
| response: | AT+SEFIRMWARE= SEONEddmmyyyy.DFU<13> | options: SEONE[date].DFU |
| | OK<13><10> | options: OK,ERR,ILL |

### 2.2.5 AT+SETUNE

#### *Set startup tune*

Set or Reset (ON/OFF) factory start-tune, this will be store in the ARM flash memory
When set OFF, the SE-ONE play the factory start-tune not when system startup.

| | | |
|---|---|---|
| send: | AT+SETUNE=ON<13> | options: ON,OFF |
| response: | OK<13><10> | options: OK,ERR,ILL |

### 2.2.6 AT+SEPLAYTUNE

Plays then intro tune from TMTLOGIC

| | | |
|---|---|---|
| send: | AT+SEPLAYTUNE<13> | |
| response: | OK<13><10> | options: OK,ERR |

### 2.2.7 AT+SEFIRMWARECHK

***Check if the firmware must be updated.***

When firmware check is false, you must updating your SE-ONE !

send: AT+SEFIRMWARECHK=DDMMYYYY<13>
response: AT+SEFIRMWARECHK=TRUE <13> options: true, false
OK<13><10> options: OK,ERR,ILL

### 2.2.8 AT+SEFAULTNR

***Get the actual error number.***

send: AT+SEFAULTNR ?<13>
response: AT+ SEFAULTNR=12 <13><10> options: 0-29

NOTE: this is an beta version

| | |
|---|---|
| 0 | "Succeeded " |
| 1 | "Disk IO errror " |
| 2 | "Assertion failed " |
| 3 | "The physical drive cannt work " |
| 4 | "Could not find the file " |
| 5 | "Could not find the path " |
| 6 | "Invalid path name " |
| 7 | "Access denied directory full " |
| 8 | "Access denied " |
| 9 | "Invalid file/directory " |
| 10 | "Write protected " |
| 11 | "Invalid logical drive " |
| 12 | "The volume has no work area " |
| 13 | "No valid FAT volume " |
| 14 | "The f_mkfs() aborted " |
| 15 | "Could not to access volume " |
| 16 | "Rejected according sharing policy" |
| 17 | "LFN could not be allocated " |
| 18 | "Number of open files " |
| 19 | "Invalid parameter " |
| 20 | "Unknow at commando " |
| 21 | "Syntax error " |
| 22 | "Usb busy " |
| 23 | "Invalid at command " |
| 24 | "Usb already installed " |
| 25 | "AT parameter error " |
| 26 | "AT not in uppercase " |
| 27 | "ARM flash error " |
| 28 | "Invalid input " |
| 29 | "Wrong SE mode " |

### 2.2.9 AT+SEFAULTTXT

***Get the actual error text.***

send: AT+SEFAULTTXT? <13>
response: AT+ SEFAULTTXT= Invalid input <13><10>

## 2.3 FM Radio

---

### 2.3.1 AT+FMTYPE

#### *Get Radio chip type*

| | | |
|---|---|---|
| send: | AT+FMTYPE? | |
| response: | AT+FMTYPE=TEA5767<13> | options:TEA5767 |
| OK<13><10> | | options: OK,ERR,ILL |

### 2.3.2 AT+FMFREQ

#### *Set FM radio frequention*

| | | |
|---|---|---|
| send: | AT+FMFREQ=88.7<13> | options: (76 to 108)* |
| response: | OK<13><10><0> | options: OK,ERR,ILL |

#### *Get FM Radio frequention*

| | | |
|---|---|---|
| send: | AT+FMFREQ?<13> | options: (76 to 108)* |
| response: | AT+FMFREQ=88.3<13> | |
| | OK<13><10><0> | options: OK,ERR,ILL |

* one decimal

### 2.3.3 AT+FMSCANUP

#### *Scan up to next radio channel*

| | | |
|---|---|---|
| send: | AT+FMSCANUP<13> | |
| response: | OK<13><10> | options: OK,ERR,ILL |

### 2.3.4 AT+FMSCANDOWN

#### *Scan down to next radio channel*

| | | |
|---|---|---|
| send: | AT+FMSCANDOWN<13> | |
| response: | OK<13><10> | options: OK,ERR,ILL |

### 2.3.5 AT+FMSM

#### *Change search mode on or off , its look alike the AFC function*

| | | |
|---|---|---|
| send: | AT+FMSM=ON<13> | options: ON,OFF |
| response: | OK<13><10> | options: OK,ERR,ILL |

#### *Get the search mode*

| | | |
|---|---|---|
| send: | AT+FMSM><13> | |
| response: | AT+FMSM=ON<13> | options: ON,OFF |
| | OK<13><10> | options: OK,ERR,ILL |

bit 6 from byte 1 SM Search mode**,** if SM = 1= ON then in search mode; if SM = 0 = OFF then not in search mode

### 2.3.6 AT+FMMUTE

#### *Set FM radio mute*

| | | |
|---|---|---|
| send: | AT+FMMUTE=ON<13> | options: ON,OFF |
| response: | OK<13><10> | options: OK,ERR,ILL |

#### *Get FM radio mute state*

| | | |
|---|---|---|
| send: | AT+FMMUTE?<13> | |
| response: | AT+FMSM=ON<13> | options: ON,OFF |
| | OK<13><10> | options: OK,ERR,ILL |

bit 7 from byte 1 MUTE, if MUTE = 1 = ON then L and R audio are muted; if MUTE = 0 = OFF then L and R audio are not muted

### 2.3.7 AT+FMPLL

#### *Set FM PLL value*

| | | |
|---|---|---|
| send: | AT+FMPLL=1<13> | options: (0-8191) 13 bits |
| response: | OK<13><10> | options: OK,ERR,ILL |

#### *Get FM PLL value*

| | | |
|---|---|---|
| send: | AT+FMPLL?<13> | |
| response: | AT+FMPLL=1<13> | options: (0-8191) 13 bits |
| | OK<13><10> | options: OK,ERR,ILL |

5 to 0 PLL[13:8] + 7 to 0 PLL[7:0] from byte 1 and 2 setting of synthesizer programmable counter for search or preset

### 2.3.8 AT+FMSUD

#### *Set search up or down direction bit*

| | | |
|---|---|---|
| send: | AT+FMSUD=UP<13> | options: UP,DOWN |
| response: | OK<13><10> | options: OK,ERR,ILL |

#### *Get search direction*

| | | |
|---|---|---|
| send: | AT+FMSUD?<13> | |
| response: | AT+FMFMSUD=DOWN<13> | options: UP,DOWN |
| | OK<13><10> | options: OK,ERR,ILL |

bit 7 from byte 3 SUD **Search Up/Down,** if SUD = 1 = ON then search up; if SUD = 0 = OFF then search down

### 2.3.9 AT+FMSSL

#### *Set the search stop level*

| | | |
|---|---|---|
| send: | AT+FMSSL=1<13> | options: 1,2,3 |
| response: | OK<13><10> | options: OK,ERR,ILL |

#### *Get the search stop level value*

| | | |
|---|---|---|
| send: | AT+FMSSL?<13> | |
| response: | AT+FMSSL=1<13> | options:1,2,3 |
| | OK<13><10> | options: OK,ERR,ILL |

bit 6 and 5 from byte 3 SSL[1:0] **Search Stop Level:**
**SSL1 SSL0 Search stop level**

0) 0 0 not allowed in search mode
1) 0 1 low; level ADC output = 5
2) 1 0 mid; level ADC output = 7
3) 1 1 high; level ADC output = 10

### 2.3.10 AT+FMHLSI

#### *Set the High Low Signal Injection bit*

| | | |
|---|---|---|
| send: | AT+FMHISI=HLGH<13> | options: HIGH,LOW |
| response: | OK<13><10> | options: OK,ERR,ILL |

#### *Get the High Low Signal Injection bit*

| | | |
|---|---|---|
| send: | AT+FMHLSI?<13> | |
| response: | AT+FMFMHLSI=LOW<13> | options: HIGH.LOW |
| | OK<13><10> | options: OK,ERR,ILL |

bit 4 from byte 3 HLSI **High/Low Side Injection:** if HLSI = 1 = HIGH then high side LO injection; if HLSI = 0 = LOW then low side LO injection

### 2.3.11 AT+FMMS

#### *Change force stereo or mono*

| | | |
|---|---|---|
| send: | AT+FMMS=STEREO<13> | options: STEREO,MONO |
| response: | OK<13><10> | options: OK,ERR,ILL |

#### *Get the force value*

| | | |
|---|---|---|
| send: | AT+FMMS?<13> | |
| response: | AT+FMMS=MONO<13> | options: STEREO,MONO |
| | OK<13><10> | options: OK,ERR,ILL |

bit 3 from byte 3 MS **Mono to Stereo:** if MS = 1 = MONO then forced mono; if MS = 0 = STEREO then stereo ON

### 2.3.12 AT+FMBAND

#### *Set the FM band range Europa and Japan/VS*

| | | |
|---|---|---|
| send: | AT+FMBAND=EUR<13> | options: EUR,JAP |
| response: | OK<13><10> | options: OK,ERR,ILL |

#### *Get the FM band range Europa and Japan/VS*

| | | |
|---|---|---|
| send: | AT+FMBAND?<13> | |
| response: | AT+FMBAND=EUR<13> | options: EUR,JAP |
| | OK<13><10> | options: OK,ERR,ILL |

bit 5 from 4e byte BL **Band Limits:** if BL = 1 JAP then Japanese FM band; if BL = 0= EUR then US/Europe FM band

### 2.3.13 AT+FMHCC

#### *Set the High cut control bit*

| | | |
|---|---|---|
| send: | AT+FMHCC=ON<13> | options ON,OFF |
| response: | OK<13><10> | options: OK,ERR,ILL |

#### *Get the High cut control bit*

| | | |
|---|---|---|
| send: | AT+FMHCC? | |
| response: | AT+FMHCC=ON<13> | options: ON,OFF |
| | OK<13><10> | options: OK,ERR,ILL |

bit 2 from 4e byte HCC **High Cut Control:** if HCC = 1 = ON then high cut control is ON; if HCC = 0 = OFF then high cut control is OFF

### 2.3.14 AT+FMSNC

#### *Set the stereo noise canceling on/off*

| | | |
|---|---|---|
| send: | AT+FMSNC=ON<13> | options ON/OFF |
| response: | OK<13><10> | options: OK,ERR,ILL |

#### *Get the stereo noise cancellation setting*

| | | |
|---|---|---|
| send: | AT+FMSNC? | |
| response: | AT+FMSNC=ON<13> | options: ON.OFF |
| | OK<13><10> | options: OK,ERR,ILL |

bit 1 from byte 4 SNC **Stereo Noise Cancelling:** if SNC = 1 = ON then stereo noise cancellation is ON; if SNC = 0 = OFF then stereo noise cancellation is OFF

### 2.3.15 AT+FMREADY

#### *Get the ready flag*

| | | |
|---|---|---|
| send: | AT+FMREADY?<13> | |
| response: | AT+FMREADY=ON<13> | options: ON,OFF |
| | OK<13><10> | options: OK,ERR,ILL |

bit 7 from 1e byte RF **Ready Flag:** if RF = 1 =ON then a station has been found or the band limit has been reached; if RF = 0 = OFF then no station has been found

### 2.3.16 AT+FMBLF

#### *Get the band limit flag*

| | | |
|---|---|---|
| send: | AT+FMBLF?<13> | |
| response: | AT+FMBLR=ON<13> | options: ON,OFF |
| | OK<13><10> | options: OK,ERR,ILL |

bit 6 from 1e byte BLF **Band Limit Flag:** if BLF = 1 = ON then the band limit has been reached; if BLF = 0 =OFF then the band limit has not been reached

### 2.3.17 AT+FMSTEREO

#### *Get the stereo status from the FM channel*

| | | |
|---|---|---|
| send: | AT+STEREO?<13> | |
| response: | AT+FMSTEREO=ON<13> | options: ON,OFF |
| | OK<13><10> | options: OK,ERR,ILL |

bit 7 byte 2 STEREO **Stereo indication:** if STEREO = 1 = ON then stereo reception; if STEREO = 0 = OFF then mono reception

### 2.3.18 AT+FMIF

#### *Get the intermediate frequency*

| | | |
|---|---|---|
| send: | AT+FMIF?<13> | |
| response: | AT+FMIF=123<13> | options: 0-127 7 bits |
| | OK<13><10> | options: OK,ERR,ILL |

IF[7:0] IF counter result

### 2.3.19 AT+FMADC

#### *Get the Analog signal level*

| | | |
|---|---|---|
| send: | AT+FMADC?<13> | |
| response: | AT+FMADC=5<13> | options: 0-15 4 bit |
| | OK<13><10> | options: OK,ERR,ILL |

bit 7 to 4 from 4e byte level ADC output

### 2.3.20 AT+FMSTARTFREQ

Flashed the FM radio frequency in the ARM memory.
When the radio turns on, the radio starts with this frequency

#### *the Set FM radio frequention*

| | | |
|---|---|---|
| send: | AT+FMSTARTFREQ=88.7<13> | options: (76 to 108)* |
| response: | OK<13><10> | options: OK,ERR,ILL |

#### *Get FM Radio frequention*

| | | |
|---|---|---|
| send: | AT+FMSTARTFREQ?<13> | options: (76 tot 108)* |
| response: | AT+FMFREQ=88.7<13> | |
| | OK<13><10> | options: OK,ERR,ILL |

* one decimal

## 2.4 DSP Digital Sound Processor / Audio Processor

---

### 2.4.1 AT+DSPTYPE

#### *Get the DSP chip type*

| | | |
|---|---|---|
| send: | AT+DSPTYPE?<13> | |
| response: | AT+DSPTYPE=TDA8425<13> | options: TDA8425 |
| | OK<13><10> | options: OK,ERR,ILL |

### 2.4.2 AT+DSPVOLL

#### *Set the left volume level*

| | | |
|---|---|---|
| send: | AT+DSPVOLL=80<13> | options: 0-100 |
| response: | OK<13><10> | options: OK,ERR,ILL |

#### *Get the left volume level*

| | | |
|---|---|---|
| send: | AT+DSPVOLL?<13> | |
| response: | AT+DSPVOLL=66<13> | options: 0-100 |
| | OK<13><10> | options: OK,ERR,ILL |

### 2.4.3 AT+DSPVOLR

#### *Set the right volume level*

| | | |
|---|---|---|
| send: | AT+DSPVOLR=80<13> | options: 0-100 |
| response: | OK<13><10> | options: OK,ERR,ILL |

#### *Get the left volume level*

| | | |
|---|---|---|
| send: | AT+DSPVOLR?<13> | |
| response: | AT+DSPVOLR=66<13> | options: 0-100 |
| | OK<13><10> | options: OK,ERR,ILL |

### 2.4.4 AT+DSPBASS

#### *Set the bass level*

| | | |
|---|---|---|
| send: | AT+DSPBASS=80<13> | options: 0-100 |
| response: | OK<13><10> | options: OK,ERR,ILL |

#### *Get the bass level*

| | | |
|---|---|---|
| send: | AT+DSPBASS?<13> | |
| response: | AT+DSPBASS=66<13> | options: 0-100 |
| | OK<13><10> | options: OK,ERR,ILL |

### 2.4.5 AT+DSPTREBLE

#### *Set the treble volume level*

| | | |
|---|---|---|
| send: | AT+DSPTREBLE=80<13> | options: 0-100 |
| response: | OK<13><10> | options: OK,ERR,ILL |

#### *Get the left volume level*

| | | |
|---|---|---|
| send: | AT+DSPTREBLE?<13> | |
| response: | AT+DSPTREBLE=66<13> | options: 0-100 |
| | OK<13><10> | options: OK,ERR,ILL |

### 2.4.6 AT+DSPVOLUP

#### *Increase the volume by 10 steps!*

| | | |
|---|---|---|
| send: | AT+DSPVOLUP<13> | |
| response: | OK<13><10> | options: OK,ERR |

### 2.4.7 AT+DSPVOLDOWN

#### *Decrease the volume by 10 steps*

send: AT+DSPVOLDOWN<13>
response: OK<13><10> options: OK,ERR

### 2.4.8 AT+DSPMUTE

#### *Set DSP mute*

send: AT+DSPMUTE=ON<13> options: ON,OFF
response: OK<13><10> options: OK,ERR,ILL

#### *Get DSP mute state*

send: AT+DSPMUTE?<13>
response: AT+DSPMUTE=ON<13> options: ON,OFF
OK<13><10> options: OK,ERR,ILL

## 2.5 MP3 DSP chip

---

### 2.5.1 AT+MP3TYPE

#### *Get the mp3 chip type*

send: AT+MP3TYPE?<13>
response: AT+ MP3TYPE=VS1053<13> options: VS1053
OK<13><10>

### 2.5.2 AT+MP3LOWLEVEL

#### *Set the low-level bit from status register inp(&h23) bit 7*

Note: this works only in MP3B mode !

When the content of the audiostream fifo lower is than this value, bit 7 will be set on

send: AT+MP3LOWLEVEL= 32<13> options: 0-65535
response: OK<13><10> options: OK,ERR,ILL

### 2.5.3 AT+MP3HIGHLEVEL

#### *Set the high-level bit from status register inp(&h23) bit 6*

Note: this works only in MP3B mode !

When the content of the audiostream fifo higher is than this value, bit 6 will be set on

send: AT+MP3HIGHLEVEL= 12232<13> options: 0-65535
response: OK<13><10> options: OK,ERR,ILL

## 2.6 VU meter

---

### 2.6.1 reserve at+vutype

### 2.6.2 AT+VULEFT

#### *Set the left VU byte*

| | | |
|---|---|---|
| send: | AT+VULEFT=80<13> | options: 0-255 |
| response: | OK<13><10> | options: OK,ERR,ILL |

#### *Get the left VU byte*

| | | |
|---|---|---|
| send: | AT+VULEFT?<13> | |
| response: | AT+VULEFT=66<13> | options: 0-255 |
| | OK<13><10> | options: OK,ERR,ILL |

### 2.6.3 AT+VURIGHT

#### *Get the left VU byte*

| | | |
|---|---|---|
| send: | AT+VURIGHT=80<13> | options: 0-255 |
| response: | OK<13><10> | options: OK,ERR,ILL |

#### *Get the left VU byte*

| | | |
|---|---|---|
| send: | AT+VURIGHT?<13> | |
| response: | AT+VURIGHT=66<13> | options: 0-255 |
| | OK<13><10> | options: OK,ERR,ILL |

### 2.6.4 AT+VULN

#### *Get the left VU number*

| | | |
|---|---|---|
| send: | AT+VULN?<13> | |
| response: | AT+VULN=7<13> | options: 0-255* |
| | OK<13><10> | options: OK,ERR,ILL |

Get IO base: PRINT INP(&H26)

*default 0-8

### 2.6.5 AT+VURN

#### *Get the right VU number*

| | | |
|---|---|---|
| send: | AT+VURN?<13> | |
| response: | AT+VURN=7<13> | options: 0-255* |
| | OK<13><10> | options: OK,ERR,ILL |

Get IO base: PRINT INP(&H27)

*default 0-8

### 2.6.6 AT+VULEDS

#### *Set the number of leds, default is 8*

| | | |
|---|---|---|
| send: | AT+VULEDS=8<13> | options: 1-255 |
| response: | OK<13><10> | |

#### *Get the number of leds*

send: AT+VULEDS?<13>
response: AT+VULEDS=8<13> options: 1-255
OK<13><10> options: OK,ERR,ILL

## 2.7 MSX

---

### 2.7.1 AT+MSXAUDIO

#### *Set MSX aux in switch*

Signal is the left output channel to pin 49 if the msx connector

send: AT+MSXAUDIO=ON<13> options: ON,OFF
response: OK<13><10> options: OK,ERR,ILL

#### *Get MSX aux line status*

send: AT+MSXAUDIO?<13>
response: AT+MSXAUDIO=OFF<13> options: ON,OFF
OK<13><10> options: OK,ERR,ILL

### 2.7.2 AT+MSXSTARTAUDIO

Flashed the this settings in the ARM memory. When the SE-ONE turns on, this setting will be used

#### *Set MSX aux in switch*

Signal is the left output channel

send: AT+MSXSTARTAUDIO=ON<13> options: ON,OFF
response: OK<13><10> options: OK,ERR,ILL

#### *Get MSX aux line status*

send: AT+MSXSTARTAUDIO?<13>
response: AT+MSXSTARTAUDIO=OFF<13> options: ON,OFF
OK<13><10> options: OK,ERR,ILL

## 2.8 USB universal serial bus

---

### 2.8.1 AT+USBINIT

Note: only in MP3B mode

#### *Install usb port*

When USB is good installed, the blue led is burn on the SE-ONE

send: AT+USBINIT<13>
response: OK<13><10> options: OK,ERR,ILL

### 2.8.2 AT+USBDEINIT

Note: only in MP3B mode

#### *De-Install the usb port*

send: AT+USBDEINIT<13>
response: OK<13><10> options: OK,ERR,ILL

### 2.8.3 AT+USBDEVICE

Note: only in MP3B mode

NONE: no device
MSD: Mass storge device (usb stick)

***Get which USB device is connected***

send: AT+USBDEVICE?<13>

response: AT+USBDEVICE =MSD<13> options: NONE,MSD
OK<13><10> options: OK,ERR,ILL

## 2.9 MSD mass storage device (only MP3B mode)

---

### 2.9.1 AT+MSDPATH

Note: only in MP3B mode

***Set the directory from the Mass Storage Device (usb stick)***

send: AT+MSDPATH=/MAP/MAP<13>
response: OK<13><10> options: OK,ERR,ILL

***Get the directory from the Mass Storage Device (usb stick)***

send: AT+MSDPATH?<13>
response: AT+MSDPATH=/MAP/MAP<13>
response: OK<13><10> options: OK,ERR,ILL

### 2.9.2 AT+MSDFILES?

Note: only in MP3B mode

***Get the available files and directories in the current directory.***

send: AT+MSDFILES?<13>
response: OK<13><10> options: OK,ERR,ILL

for read the files run this program:

```
1000 D = INP(&H25)
1010 PRINT CHR$(D);
1020 IF D = 10 THEN END
1030 GOTO 1000
```

### 2.9.3 AT+MSDPLAY

Note: only in MP3B mode

***Play an music file from Mass Storage Device (usb stick)***

Note: This function works stand-alone in the SE_ONE

send: AT+MSDPLAY=MUSIC.MP3<13>
response: OK<13><10> options: OK,ERR,ILL

### 2.9.4 AT+MSDPAUSE

Note: only in MP3B mode

***Pause playing***

send: AT+MSDPAUSE<13>
response: OK<13><10> options: OK,ERR,ILL

### 2.9.4 AT+MSDSTART

Note: only in MP3B mode

***Continue music playing***

send: AT+MSDSTART<13>
response: OK<13><10> options: OK,ERR,ILL

### 2.9.6 AT+MSDSTOP

Note: only in MP3B mode

***Stop playing***

send: AT+MSDSTOP<13>
response: OK<13><10> options: OK,ERR,ILL

## 2.10 SRAM serial Ram (works not in MP3A mode)

---

### 2.10.1 AT+SRAMRESET

Note: only in MP3B mode

***Reset the pointers of the SRAM FIFO***

send: AT+SRAMRESET<13>
response: OK<13><10> options: OK,ERR,ILL

### 2.10.2 AT+SRAMMAXADR?

Note: only in MP3B mode

***Get the content of the SRAM FIFO***

send: AT+SRAMMAXADR?<13>
response: AT+SRAMMAXADR=32000<13>
OK<13><10> options: OK,ERR,ILL

### 2.10.3 AT+SRAMREADADR

Note: only in MP3B mode

***Set the serial read address pointer***

send: AT+SRAMREADADR=10<13> options: 0-SRAM MAX
response: OK<13><10> options: OK,ERR,ILL

***Get the serial read address pointer***

send: AT+SRAMREADADR?<13> options: 0-SRAM MAX
response: AT+SRAMREADADR=10<13>
OK<13><10> options: OK,ERR,ILL

### 2.10.4 AT+SRAMWRITEADR

Note: only in MP3B mode

***Set the serial write address pointer***

send: AT+SRAMWRITEADR=10<13> options: 0-SRAM MAX
response: OK<13><10> options: OK,ERR,ILL

### Get the serial write address pointer

send: AT+SRAMWRITEADR?<13> options: 0-SRAM MAX
response: AT+SRAMWRITEADR=10<13>
OK<13><10> options: OK,ERR,ILL

# 3.0 How to... examples

## 3.1 How to set mode to MP3 ?

send:
AT+SEMODE=MP3A

Run a MP3 player program.

Note:
After reset, the SE-ONE startup default in MP3A mode

## 3.2 How to use the radio receiver?

send: AT+SEMODE=FM
send: AT+FMFREQ=88.0

***To search UP***
Send: AT+FMSCANUP

***To search Down***
Send: AT+FMSCANDOWN

## 3.3 How to play form USB stick?

(use this only when USB is not installed):
{
send: AT+SEMODE=MP3B
send: AT+USBINIT
note: The blue led must burn, than the USB port is active
}

Send: AT+MSDPATH=MAP (when not in root of the usb-stick)
send: AT+MSDPLAY=YOURFILE.MP3

***To pause playing:***
send: AT+MSDPAUSE

***To continu playing after pause:***
send: AT+MSDSTART

***To stop playing:***
send: AT+MSDSTOP

## 3.4 How to get the files list form USB stick?

(use this only when USB is not installed):
{
send: AT+SEMODE=MP3B
send: AT+USBINIT
note: The blue led must burn, than the USB port is active
}

send: AT+MSDFILES?

Note: go to BASIC

for read the files run this program:

```
1000 D = INP(&H25)
```

```
1010 PRINT CHR$(D);
1020 IF D = 10 THEN END
1030 GOTO 1000
```

### 3.5 How to change the current directory form USB stick?

(use this only when USB is not installed):

{
send: AT+SEMODE=MP3B
send: AT+USBINIT
note: The blue led must burn, than the USB port is active
}

send: AT+PATH=/MAP/MAP

***To read the current path:***
send: AT+PATH?

### 3.6 How to use the Serial Ram?

works only in MP3B and FM mode !

for the first time:
{
send: AT+SEMODE=MP3B
send: AT+SRAMRESET
}

***How to write data into the fifo:***
send: AT+SRAMWRITEADR=10

Note: go to BASIC

```
Out &h21,65     :rem ('A')
Out &h21,66     :rem ('B')
Out &h21,67     :rem ('C')
Out &h21,68     :rem ('D')
```

Note: go to AT.BAS
send: AT+SRAMWRITEADR?
Response: AT+SRAMWRITEADR=14

***How to read data from the fifo:***
send: AT+SRAMREADADR=10

Note: goto BASIC

```
PRINT (INP(&h21));
PRINT (INP(&h21));
PRINT (INP(&h21));
PRINT (INP(&h21));
```

Screen shows: ABCD

Note: goto AT.BAS
send: AT+SRAMREADADR?
Response: AT+SRAMREADADR=14

# 4.0 Applications

This applications can been found at tmtlogic.com tap support.

### 4.1 Radio apps:

Radio app (SymbOS) written by Edoz
Basic Radio Radio.asc saved in ASCII format

### 4.2 MP3 apps:

Playinfo.bas (sunrise)
Playmp3.com (sunrise)
Symamp (symbOS)

### 4.3 UTILITY apps:

SEONE.COM a MSXDOS AT handler (Paul B)
AT.LDR a basic AT handler (Hans t)
AT_RESP.BAS a basic AT handler with response (Hans t)

# 5.0 Basic examples

## *5.1 AT.BAS*

```
100 WIDTH 80
110 '
120 'Example:  AT COMMAND:? AT+SEMODE?
130 '
140 'set caps lock
150 DEFUSR1=&HF36:A=USR1(0)
160 '
170 '
180 AT$=""
190 INPUT"AT COMMAND:";AT$
200 L = LEN(AT$)
210 IF L = 0 THEN GOTO 180
220 FOR T = 1 TO L
230 S = ASC(RIGHT$(LEFT$(AT$,T),1))
240 OUT &H20,S
250 NEXT T
260 OUT &H20,13
270 '
280 'read response
290 IF INKEY$<>"" THEN GOTO 180
300 '
310 I = INP(&H20)
320 IF I = 0 THEN GOTO 280
330 IF I > 31 THEN PRINT CHR$(I);
340 IF I = 13 THEN PRINT
350 IF I = 10 THEN GOTO 180
360 GOTO 280
```

(Hans t)

# 6.0 Z80 examples

# 7.0 Electrical descriptions

# 8.0 Troubleshooting

## *8.1 No Sound*

- send:  AT+MSXAUDIO=ON
- send:  AT+DSPVOLL=100
- send:  AT+DSPVOLR=100

## *8.2 USB don't work*

- After mode MP3b and USBINIT the blue led must be burn on, otherwise reset the SE-ONE and try again. This is an bug in the mp3b software.