EVALUATION KIT AVALIABLE

TIM-SC1 series

(TIM-SC1a, TIM-SC1b)

General Description

TIM-SC1 is a high quality and low price OEM ADS-B receiver series operating at 1090MHz. It is based on the proven FPGA-In-The-LoopTM technology, which is a unique combination of a single-core processor and FPGA. The patented solution allows high-speed RF data processing with significantly reduced number of electronic components. Simultaneous miniaturization of the module and its OEM nature open a wide range of possible applications. Fast UART interface and easy configuration with AT-commands allows for the simple integration of the module with the user's system. In addition, extra interfaces open the way to customize the firmware and extend the module with non-standard functions. There are several communication interfaces, protocols and special functionalities available on request. For more information please contact: support@aerobits.pl.

Applications

- SAA / DAA (Sense and Avoid / Detect and Avoid)
- UAS ground stations and high-density traffic surveillance
- UTM / U-Space construction (traffic surveillance network)
- Early warn devices for UAS operators
- Collision prediction devices for gliders / paragliders
- ADS-B/In/Out devices that meet the NextGen/SESAR philosophy
- Etc.

Features

- Smallest ADS-B implementation on a surface of <2cm^2
- Receiving of ADS-B with RF signal strength/quality analysis
- High-resolution ADC with real-time signal processing; best-in-class aircraft tracking
- High sensitive front-end, jamming and ESD protection (only TIM-SC1b) with ranges over 150 km (open space, 1dBi antenna)
- Simple module integration via UART interface and AT commands
- Up to 100 tracked aircrafts simultaneously
- Multiple supported protocols: AERO, MAVLink
- Scalable OEM solution with enormous customization potential (additional functions or interfaces on request)
- Firmware update capability (uC and FPGA)
- Low power consumption 3.3V/70mA
- Small outline: 16.0 x 13.0 x 2.65mm, ver. (a);
  20 x13.0 x 2.65mm, ver. (b), weight < 1.5g
- Designed to meet MOPS defined in TSO-C199 (DF17, DF18, DF19)

Ordering Information

Type	Symbol	Description
OEM Module	TIM-SC1a	Without antenna connector
OEM Module	TIM-SC1b	With antenna connector (U.FL)
Evaluation kit	EVAL-TIM-SC1	Based on TIM-SC1a

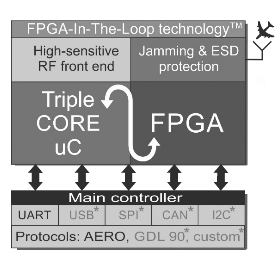

Figure 1: Structure, interfaces and protocols of TIM-SC1 series.
*on request only (custom solutions)

Contents

Electrical characteristics

Absolute Maximum Ratings			
Parameter	Min.	Max.	Unit
Storage temperature	-40	+85	°C
Supply voltage (VCC)	2.7	3.6	DCV
Other pin voltage	VSS-0.4	VCC + 0.4	DCV
RF input power	-	+15	dBm
Recommended Operating Conditions			
Parameter	Min.	Max.	Unit
Temperature	-40	+70[1]	°C
Supply voltage (VCC)	3.0	3.6	DCV

Table 1: Absolute maximum ratings and recommended operation conditions.

General Electrical Specification					
Parameter	Description	Min.	Typ.	Max.	Unit
Carrier frequency		-	1090	-	MHz
RX sensitivity	For operation at 50Ω U.fl connector (TIM-SC1b)	-	-80	-	dBm
Input Low Voltage	RESET, UARTs, CAN, USB, SPI, I2C	-0.3	-	0.8	DCV
Input High Voltage	RESET, UARTs, CAN, USB, SPI, I2C, GPIO	0.7 VCC	-	VCC + 0.3	DCV
Output Low Voltage	UARTs, CAN, USB, I2C, SPI, GPIO	-	-	0.4	DCV
Output High Voltage	UARTs, CAN, USB, I2C, SPI, GPIO	VCC - 0.4	-	-	DCV
Current consumption		-	70	-	mA

Table 2: General electrical specification.

[1] Cooler required – please contact support@aerobits.pl

Pin definition

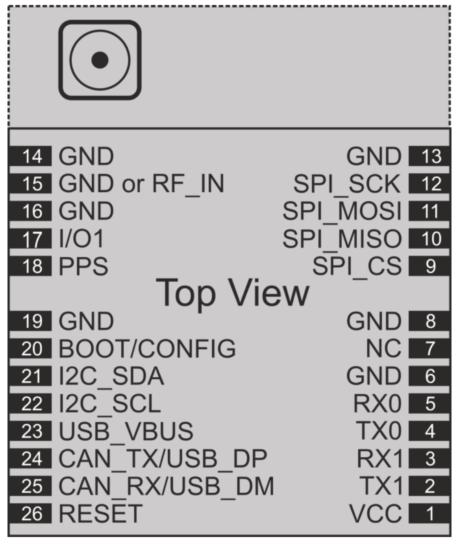

Figure 2: Pin assignment for TIM-SC1a and TIM-SC1b.

Pin No.	Pin Name	Pin Type	Description
1	VCC	Power	3.3V (digital supply)
2	TX1	CMOS Output	UART1 data output
3	RX1	CMOS Input	UART1 data input
4	TX0	CMOS Output	UART0 data output
5	TX1	CMOS Input	UART0 data input
6	GND	GND	Common ground
7	NC	Reserved	Keep floating
8	GND	GND	Common ground
9	SPI_CS	CMOS Output	Serial Peripheral Interface / chip select
10	SPI_MISO	CMOS Input	Serial Peripheral Interface / data input
11	SPI_MOSI	CMOS Output	Serial Peripheral Interface / data output
12	SPI_SCK	CMOS Output	Serial Peripheral Interface / clock
13	GND	GND	Common ground
14	GND	GND	Common ground
15 TIM-SC1a	RF_IN	RF Input	RF Input (antenna)
15 TIM-SC1b	GND	GND	Common ground
16	GND	GND	Common Ground
17	I/O1	CMOS Output	LED Output (digital)
18	PPS	CMOS Input	1PPS GNSS signal
19	GND	GND	Common ground
20	BOOT/CONFIG	CMOS Input	Bootloader / Configuration mode
21	I2C_SDA	Bi-directional	I2C Data line
22	I2C_SCL	Bi-directional	I2C Clock line
23	USB_VBUS	Power	USB Power line
24	USB_DP	Bi-directional	USB+
25	USB_DM	Bi-directional	USB-
26	RESET	CMOS Input	Reset input / active low
	Interfaces and I/Os not used in standard design (for custom implementation only)		

Table 3: Pinout.

General principle of operation

During work module goes through multiple states. In each state operation of the module is different. Each state and each transition is described in paragraphs below.

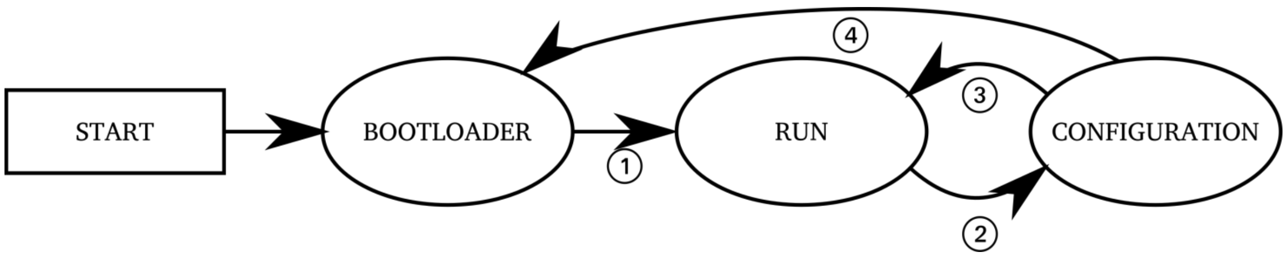

Figure 3: State machine of TIM-SC1module.

States of operation

BOOTLOADER state

This is an initial state of TIM-SC1module after restart. Firmware update is possible here. Typically module transits automatically to RUN state. It is possible to lock module in this state (prevent transition to RUN state) using one of BOOTLOADER triggers. UART baudrate is constant and is set to 115200bps. After powering up module, it stays in this state for 1s. If any BOOTLOADER trigger is not present, module will transit to RUN state. Firmware upgrade is possible with use of AEROBITS_updater.exe program. To acquire this program please contact: support@aerobits.pl.

RUN state

In this state module is working and receiving the data from aircraft. It uses selected protocol to transmit received and decoded data to the host system. In this state of operation module settings are loaded from non-volatile internal memory, including baudrate.

CONFIGURATION state

In this mode change of stored settings is possible. Operation of the module is stopped and baudrate is set to fixed 115200bps. Change of settings is done by using AT-commands. Modifications are automatically stored in non-volatile memory on exiting state. Additional set of commands is also available in this state e.g. to reboot module into BOOTLOADER state, or to check serial number and firmware version. It is possible to lock module in this state (similarly to BOOTLOADER) using suitable command.

Transitions between states

For each of state transitions, different conditions must be met, which are described below. Generally, the onlystable state is RUN. Module always tends to transit into this state. Moving to other states requires host to take some action.

BOOTLOADER to RUN transition

BOOTLOADER state is semi-stable, that means the module requires additional action to stay in BOOTLOADER state. This transition will occur automatically if any of the required actions will not be taken. To stay in BOOTLOADER state (prevent this transition) one of following action must be processed:

- Pull BOOT/CONFIG pin low during start of module
- Send `AT+LOCK=1` command while device is in BOOTLOADER state (always after power on for approximately 1s)

- Send AT+REBOOT_BOOTLOADER command in CONFIGURATION state. This will move to BOOTLOADER state and will lock module in this state.

If none of preceding conditions are meet, the module will try to transit into RUN state. Firstly it will check firmware integrity. When firmware integrity is confirmed, module will transit into RUN state, if not, it will stay in BOOTLOADER state.

To transit into RUN state:

- Release or pull high BOOT/CONFIG pin
- If module is locked, send AT+LOCK=0 command

When module enters RUN mode it will send AT+RUN_START command.

RUN to CONFIGURATION transition

To transit from RUN into CONFIGURATION state, host should do one of the following:

- Pull BOOT/CONFIG pin low
- Send AT+CONFIG=1 (using current baudrate). This method is not recommended, because module will support multiple protocols in future and Aerobits Sp. z o.o. cannot ensure that this command will be present in all protocols.

When moduleleaves RUN state it will send AT+RUN_END command and then AT+CONFIG_START command, on entering CONFIGURATION state. The former will be send using baudrate from settings, the latter will be send using 115200bps baudrate.

CONFIGURATION to RUN transition

To transit from CONFIGURATION into RUN state, host should do one of the following:

- Release or pull high BOOT/CONFIG pin
- Send AT+CONFIG=0 command.

When module leaves CONFIGURATION state it will send AT+CONFIG_END command and then AT+RUN_START command, on entering RUN state. The former will be send using 115200bps baudrate, the latter will be send using baudrate from settings.

CONFIGURATION to BOOTLOADER transition

To transit from RUN into CONFIGURATION state, host should do one of the following:

- Send AT+REBOOT_BOOTLOADER command.
- Send AT+REBOOT and when module enters BOOTLOADER state, prevent transition to RUN state.

UART configuration

Communication between module and host device is done using UART interface. This interface has following settings.

UART Settings				
Parameter	Min.	Typ.	Max	Unit
Baudrate	-	115200	-	bps
Stop Bits Number	-	1	-	-
Flow Control	-	None	-	-
Parity Bit	-	None	-	-

Table 4: UART settings.

Settings

In RUN state, operation of the module is determined based on stored settings. Settings can be changed in CONFIGURATION state using AT-commands. Settings can be written and read. Values are stored in non-volatile memory when leaving CONFIGURATION state, so they will be preserved during restart.

Write settings

`AT+SETTING=VALUE`
For example `AT+RAW_S=1`
Response: `AT+OK`

Read settings

`AT+SETTING?`
For example: `AT+RAW_S?`
Response: `AT+RAW_S=1`

Errors

Errors are reported using following structure:
`AT+ERROR (DESCRIPTION)`
DESCRIPTION is optional and contains information about error.

Command endings

Every command must be ended with one of the following character sequences: “\n”, “\r” or “\r\n”. Commands without suitable ending will be ignored.

Uppercase and lowercase

All characters (except preceding `AT+`) used in command can be both uppercase and lowercase, so following commands are equal:

```
AT+RAW_S?
AT+raw_s?
```

NOTE: This statement is true in configuration state, not in bootloader state. in bootloader state all letters must be uppercase.

Available settings

Setting	Min value	Max value	Default value	Comment
AERO	0	1	1	Display aircraft messages in AERO protocol: 0 - disable 1 - enable
STATS	0	1	1	Showing statistics messages in AERO protocol: 0 - disable 1 - enable
PROTOCOL	0	0	0	Used protocol 0 - AERO protocol 1 - MAVLink

Table 5: Settings

Example

NOTE: TODO: Inny przykład bo nie ma tych komend w SC1!

Commands

Apart from settings, module supports set of additional commands. Format of this commands are similar to those used for settings, but they do not affect operation of module in RUN state.

Commands in BOOTLOADER and CONFIGURATION state

AT+LOCK

`AT+LOCK=1` - Set lock to enforce staying in BOOTLOADER or CONFIGURATION state
`AT+LOCK=0` - Remove lock
`AT+LOCK?` - Check if lock is set

AT+BOOT

`AT+BOOT?` - Check if module is in BOOTLOADER state

Response:

`AT+BOOT=0` - module in CONFIGURATION state
`AT+BOOT=1` - module in BOOTLOADER state

Commands in CONFIGURATION state

AT+CONFIG

`AT+CONFIG=0` - Transition to RUN state.
`AT+CONFIG?` - Check if module is in CONFIGURATION state.

Response:

`AT+CONFIG=0` - module in RUN state
`AT+CONFIG=1` - module in CONFIGURATION state

AT+SETTINGS?

`AT+SETTINGS?` - List all settings. Example output:

```
AT+AERO=1
AT+STATS=1
AT+PROTOCOL=0
```

AT+SETTINGS_DEFAULT

`AT+SETTINGS_DEFAULT` - Set all settings to their default value.

AT+SERIAL_NUMBER

`AT+SERIAL_NUMBER?` - Read serial number of module.

Response:

`AT+SERIAL_NUMBER=0202041E43`

AT+FIRMWARE_VERSION

`AT+FIRMWARE_VERSION?` - Read firmware version of module.

Response:

`AT+FIRMWARE_VERSION=10101017(May 11 2018)`

AT+REBOOT

`AT+REBOOT` - Restart module.

AT+REBOOT_BOOTLOADER

`AT+REBOOT_BOOTLOADER` - Restart module to BOOTLOADER state.

NOTE: This command also sets lock.

Commands in RUN state

AT+CONFIG=1 - transition to CONFIGURATION state.

NOTE: This command also sets lock.

NOTE: This command is depreciated and could be not supported in all future protocols. It is better to use BOOT/CONFIG pin to make module transit to CONFIGURATION state.

AERO protocol

AERO protocol is simple text protocol, that allows fast integration and analysis of tracked aircrafts. AERO messages starts with "#" character and ends with "\r\n" characters. There are 2 types of messages. AIRCRAFT message and STATISTICS message.

CRC

Frames include CRC value for consistency check. CRC value is calculated using standard CRC16 algorithm and is based on every character in frame starting from '#' to last comma ',' (excluding last comma). After calculation, value is appended to frame using hexadecimal coding. Function that calculates CRC is shown below.

```
uint16_t crc16(const uint8_t* data_p, uint32_t length){
    uint8_t x;
    uint16_t crc = 0xFFFF;
    while (length--){
        x = crc>>8 ^ *data_p++;
        x ^= x>>4;
         crc = (crc<<8) ^ ((uint16_t)(x <<12)) ^ ((uint16_t)(x <<5)) ^ ((uint16_t)x);
    }
    return swap16(crc);
}
```

Aircraft message

This message describes state vector of aircraft and is send once per second.

```
#A:ICAO,FLAGS,CALL,SQ,LAT,LON,TRACK,HEA,VELH,VELV,SIGS,SIGQ,FPS,RES,CRC\r\n
```

#A	Aircraft message start indicator	Example value
ICAO	ICAO number of aircraft (3 bytes)	3C65AC
FLAGS	Flags (see below)	1
CALL	Callsign of aircraft	N61ZP
SQ	SQUAWK of aircraft	7232
LAT	Latitude in degrees	57.57634
LON	Longitude in degrees	17.59554
ALT	Pressure altitude in feets	5000
TRACK	Track of aircraft in degrees [0,360)	35
VELH	Horizontal velocity of aircraft in knots	464
VELV	Vertical velocity of aircraft in ft/min	-1344
SIGS	Signal strength in mV	840
SIGQ	Signal quality in mV	72
FPS	Number of raw MODE-S frames received from aircraft during last second	5
RES	Reserved for future use	
CRC	CRC16 (described in CRC section)	2D3E

Table 6: Detailed message description.

Value	FLAG type	Description
0x0001	PLANE_ON_THE_GROUND	The aircraft is on the ground
0x0002	PLANE_IS_MILITARY	The aircraft is military object
0x0004	PLANE_ALTITUDE_BASED_ON_GNSS	The aircraft has no pressure altitude, but altitude based on GNSS
0x0100	PLANE_UPDATE_ALTITUDE	During last second altitude of this aircraft was updated
0x0200	PLANE_UPDATE_POSITION	During last second position (LAT & LON) of this aircraft was updated
0x0400	PLANE_UPDATE_TRACK	During last second track of this aircraft was updated
0x0800	PLANE_UPDATE_VELO_H	During last second horizontal velocity of this aircraft was updated
0x1000	PLANE_UPDATE_VELO_V	During last second vertical velocity of this aircraft was updated

Table 7: Flags description.

If data of any field of frame is not available, then this field is empty. For example:

```
#A:4CA948,300,,2122,52.99750,13.76526,37000,169,442,0,814,72,3,,6F1C\r\n
#A:424313,,,2362,52.43431,14.84535,37000,65,456,0,806,61,0,,6843\r\n
```

NOTE: SIGS and SIGQ fields are updated based on raw MODE-S frames. They are calculated from frames received in last second. If there were no receiver frames (FPS=0), those fields will not be updated.

NOTE: SIGS is measured based on analog RF signal. This signal has DC offset of about 700mV.

Statistics message

This message contains some useful statistics about operation of module. Format of that frame is shown below:

```
#S:CPU,RES,RES,FPSS,RES,RES,CRC
```

#S	Statistics message start indicator	Example
CPU	CPU load in %	12.1
RES	Reserved for future use	-
RES	Reserved for future use	-
FPSS	Number of MODE-S frames received in last second	3
RES	Reserved for future use	-
RES	Reserved for future use	-
CRC	CRC16 (described in CRC section)	2D3E

Table 8: Statistics message description.

MAVLink protocol

Module can be switched to use MAVLink protocol. This can be achieved by altering PROTOCOL setting. When MAVLink protocol is used, module is sending list of aircrafts every second. Aircrafts are encoded using ADSB_VEHICLE message. MAVLink messages has standard format, which is well described on this protocol webpage (ardupilot.org/dev/docs/mavlink-commands.html). MAVLink message contains several data fields which are described below.

Field Name	Type	Description
ICAO_address	uint32_t	ICAO address
lat	int32_t	Latitude, expressed as degrees * 1E7
lon	int32_t	Longitude, expressed as degrees * 1E7
altitude_type	uint8_t	Type from ADSB_ALTITUDE_TYPE enum
altitude	int32_t	Altitude(ASL) in millimeters
heading	uint16_t	Course over ground in centidegrees
hor_velocity	uint16_t	The horizontal velocity in centimeters/second
ver_velocity	uint16_t	The vertical velocity in centimeters/second, positive is up
callsign	char[9]	The callsign, 8 chars + NULL
emitter_type	uint8_t	Type from ADSB_EMITTER_TYPE enum
tslc	uint8_t	Time since last communication in seconds
flags	uint16_t	Flags to indicate various statuses including valid data fields
squawk	uint16_t	Squawk code

Table 9: MAVLink ADSB_VEHICLE message description

Minimum integration recommendation

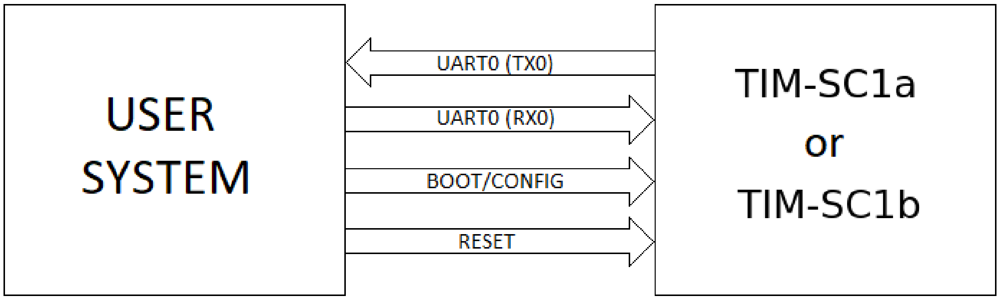

Figure 4: Simplest module integration with user MCU.

Mechanical drawing

All dimensions are in mm.

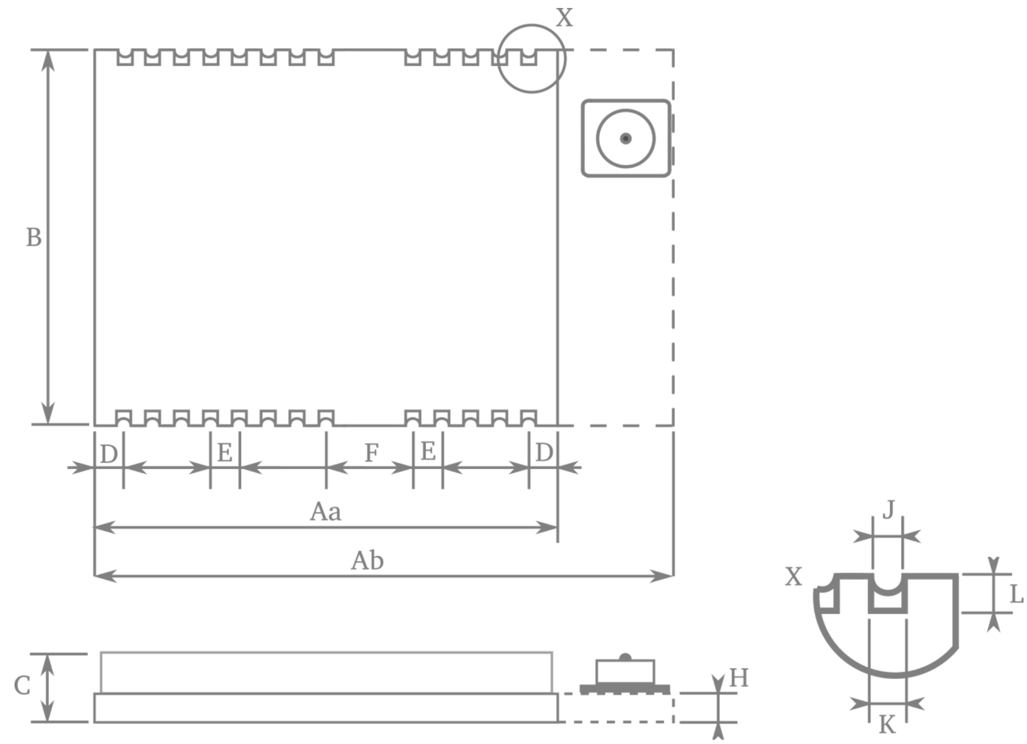

Figure 5: Dimensions.

Symbol	Min. (mm)	Typ. (mm)	Max. (mm)
Aa (TIM-SC1a)	15.9	16.0	16.1
Ab (TIM-SC1b)	19.9	20.0	20.1
B	12.9	13.0	13.1
C	2.3	2.40	2.5
D	0.9	1.0	1.1
E	0.9	1.0	1.1
F	2.9	3.0	3.1
H	0.6	0.7	0.8
J	0.4	0.5	0.6
K	0.6	0.7	0.8
L	0.7	0.8	0.9

Table 10: Dimensions and tolerances.

Recommended layout

All dimensions are in mm.

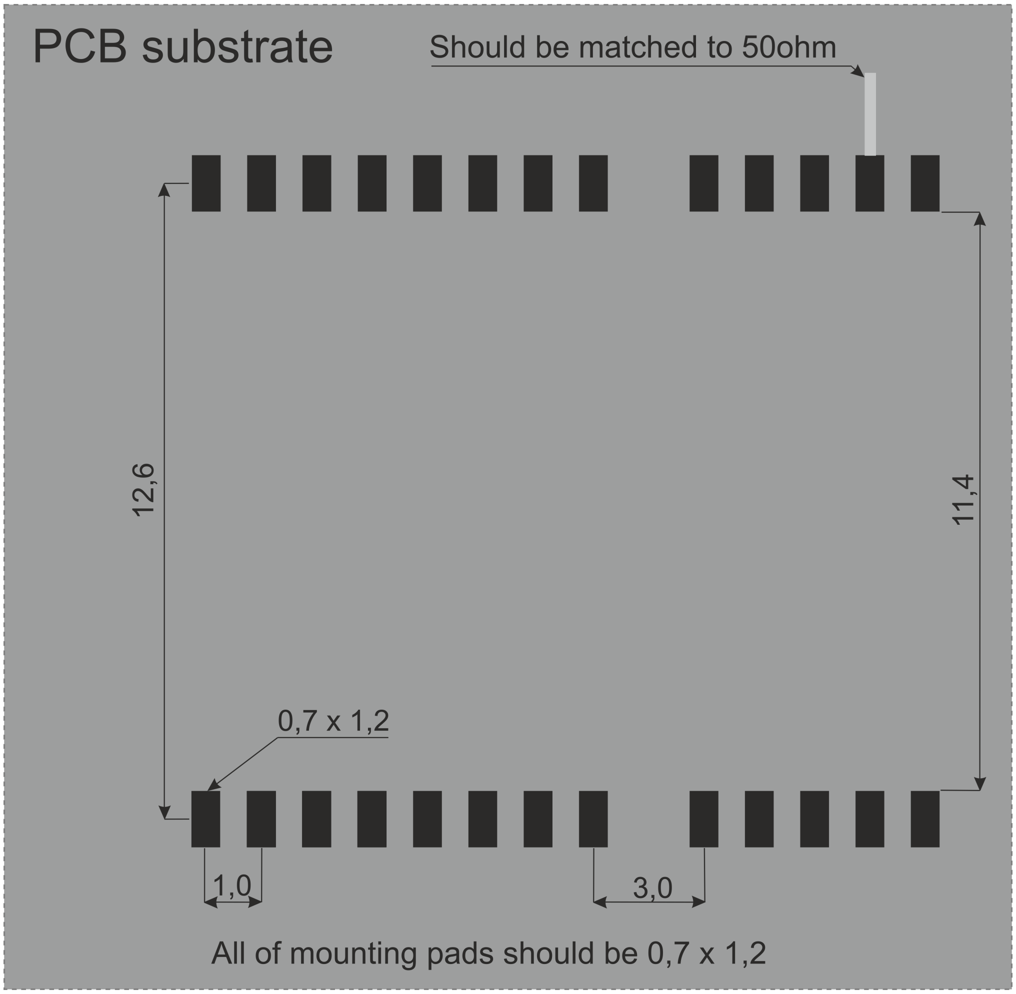

Figure 6: Recommended footprint.

NOTE: In case of TIM-SC1a the RF input (pad 18) should be matched to 50ohm.

Revision history

Date	Revision	Changes
24-May-2018	1	Initial release.
20-Sep-2018	2	Corrected inconsistency in mechanical drawing.

Table 11: Document revision history.

Please read carefully