

TimoTwo Development Documentation

Version 1.6.0

LumenRadio AB

Copyright © 2019-2025, LumenRadio AB

Table of contents

1. What is CRMX™	3
2. CRMX TimoTwo	3
3. Pin Assignments and Functions	4
4. Reference Design and PCB mounting	8
5. Reflow Soldering	11
6. LED Outputs	13
6.4 DMX LED	14
6.5 RDM LED	14
6.7 RGB LED	15
11. DMX Interface	45
11.1 SPI	45
12. RDM Interface	48
13.1 RDM RX	53
13.2 RDM TX	53
15.3 SPI	55
16. Flex mode selection	57
16.1 SPI	57
17. Specifications	59
20.3 CE	65

This documentation relates to firmware version 1.0.7.2 or later

1. What is CRMXTM

CRMXTM is an acronym for Cognitive Radio MultipleXer - it is the first smart wireless system to automatically and continuously adapt to its surroundings in real time. CRMXTM was specifically developed to meet the demand for reliable, easy to use, and cost effective wireless lighting controls.

2. CRMXTM TimoTwo

Building on the phenomenal success of CRMXTMchip and TiMo, LumenRadio offers TimoTwo – a next generation wireless DMX/RDM module designed for high volume manufacturing. TimoTwo keeps the small form factor as TiMo (33.5mm x 18.5mm), with pin- compatibility, offering even more features.

TimoTwo is fully compliant with the ETSI EN 300 328 (v2.2.2) revision as well as fully FCC certified with a modular approval to 100mW.

As with all LumenRadio receivers, TimoTwo is fully compatible with CRMXTM transmitters as well as transmitters from Wireless Solution running in G3, G4, G4S 2.4GHz or G5 2.4 GHz mode, enabling a single wireless component to maximise compatibility within fixtures. Manufacturers can add TimoTwo to their products easily with full OrCAD, Allegro and DXF symbol libraries available to simply drop onto existing layouts.

2.1 Features

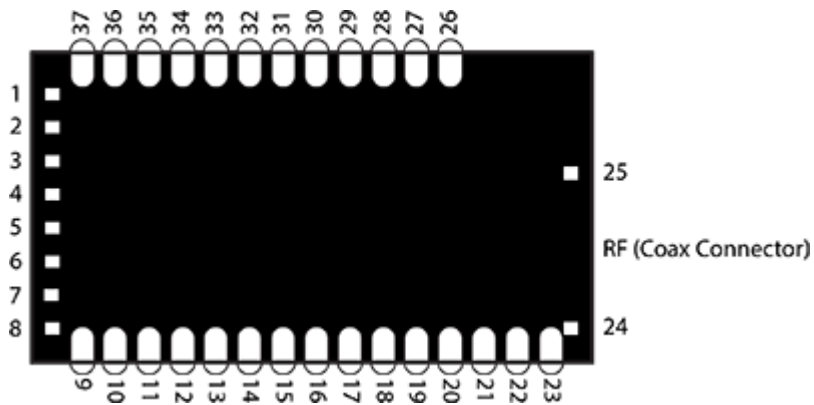
- Receives CRMXTM², CRMXTM Classic, W-DMX G3, G4, G4S and G5
- Transmit CRMXTM Classic, W-DMX G3 or G4S
- Supports ANSI E1.11 - DMX512-A and ANSI E1.20 - RDM
- Connectivity based on Bluetooth Smart, allowing for easy connection from any phone or tablet
- Cognitive coexistence - dynamically avoids occupied frequencies
- DMX fidelity and frame integrity
- DMX frame rate and frame size auto sensing
- Fixed 5 ms end-to-end latency in CRMXTM Classic or CRMXTM² modes
- Small footprint at 18.5 mm x 33.5 mm
- U.FL/IPEX external antenna connector
- All configuration data is stored in non-volatile memory, 20 years data retention
- TimoTwo contains upgradeable firmware for future proofing
- Over-the-air firmware upgrades

3. Pin Assignments and Functions

This section describes the pin assignments and pin functions.

3.1 Pin assignments

Module seen from the top



3.2 Pin functions

Pin	Name	Pin type	Description
1	VSS	Power	Ground (0V)
2	IRQ	Digital output	Interrupt signal, active low
3	CS	Digital input	Chip select, active low
4	SCK	Digital input	SPI clock
5	MOSI	Digital input	SPI Master Out, Slave In
6	MISO	Digital output	SPI Master In, Slave Out
7	VSS	Power	Ground (0V)
8	DMX_TXD	Digital output	DMX TXD
9	VSS	Power	Ground (0V)
10	RS485_DE	Digital output	RS485 driver control signal
11	RS485_RE	Digital output	RS485 driver control signal
12	DMX_RXD	Digital input	DMX RXD (3.3 V max)
13	RDI_LVL0	Digital output	Radio level LED
14	RDI_LVL1	Digital output	Radio level LED
15	RDI_LVL2	Digital output	Radio level LED
16	RDI_LVL3	Digital output	Radio level LED
17	RDI_LVL4	Digital output	Radio level LED
18	RDM	Digital output	RDM LED
19	DMX	Digital output	DMX LED

APin	Name	Pin type	Description
20	N.C.	No connection	Do not connect
21	N.C.	No connection	Do not connect
22	VSS	Power	Ground (0V)
23	VSS	Power	Ground (0V)d
24	VSS	Power	Ground (0V)
25	VSS	Power	Ground (0V)
26	VSS	Power	Ground (0V)
27	VSS	Power	Ground (0V)
28	SCL	Digital input/output	I2C clock to RGB LED driver ¹
29	SDA	Digital input/output	I2C data to RGB LED driver ¹
30	VSS	Power	Ground (0V)
31	N.C	No connection	Do not connect
32	LINK_SW	Digital input	Link control switch input
33	STATUS_LED	Digital output	Status LED
34	LINKED	Digital output	Linked to transmitter LED
35	RF_LINK	Digital output	RF link LED
36	VDD	Power	Power supply (3.3V)
37	VSS	Power	Ground (0V)
ANT	RF ANT	RF	Antenna connector

¹ Shall be pulled high (to Vcc) through a 4.7 kOhm resistor if RGB LED is used.

TimoTwo has some additional pads on the bottom layer not lined out in this table, these pads shall be left unconnected, and shall not be matched with any corresponding pads on the PCB. The PCB shall be left without any copper under the module.

4. Reference Design and PCB mounting

4.1 Reference design files

A complete reference design including PCB layout and PCB design guidelines is available from LumenRadio to make integration easier. Please send your inquiry to LumenRadio for details.

4.2 PCB mounting

4.2.1 Antenna

The antenna connector of the module is of u.FL type. A coax cable with a characteristic impedance of 50 ohm shall be used to connect an antenna. The antenna shall have a characteristic impedance of 50 ohm at 2.45GHz.

4.2.2 Layout considerations for the main board

TimoTwo has been specifically designed in order to achieve good RF performance. In order to maintain this, there are some guidelines that we would like to stress:

The use of ground planes also on the main board cannot be emphasized enough. Good decoupling of any high speed digital circuitry is a must. Many embedded type microprocessors today have clock frequencies with clocks or overtones that reach well into the GHz range. It is perfectly possible for an embedded design to pass any EMC certification and still cause disturbances that will block the RF reception of the TimoTwo module. The sensitivity of the TimoTwo receiver is -96dBm therefore it is recommended to keep disturbances below -100dBm in the frequency range of operation.

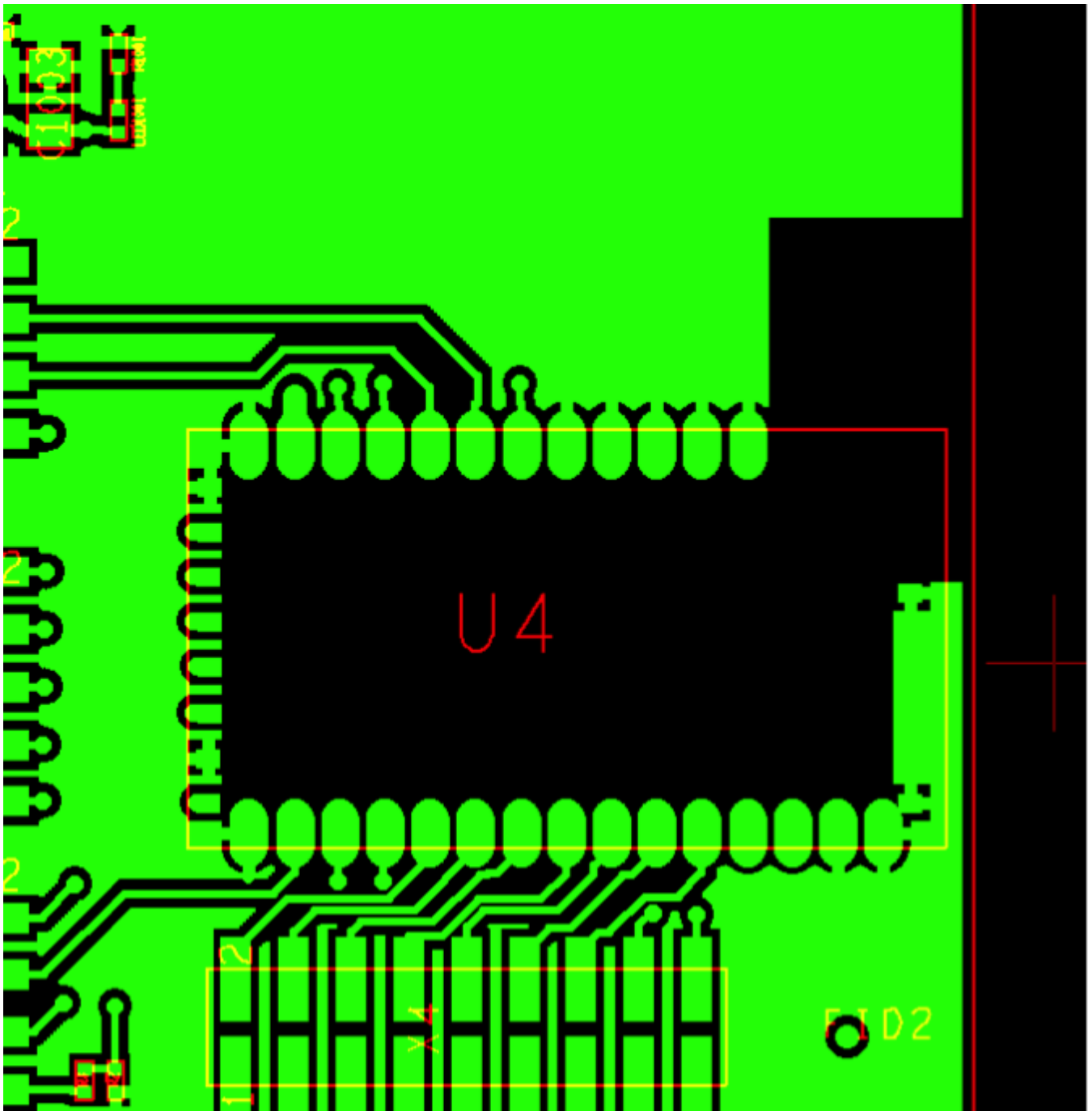
A near field probe connected to a spectrum analyzer will show if there are any disturbances present on the 2.45 GHz band generated by the microprocessor or any other device that is placed on the main board. Pay special attention to readymade LAN-products "Server in a RJ connector". They pass EMC certifications, but some of them radiate badly on 2.45 GHz. If disturbances can be seen on a spectrum analyzer

- then the TimoTwo module will have impaired reception.

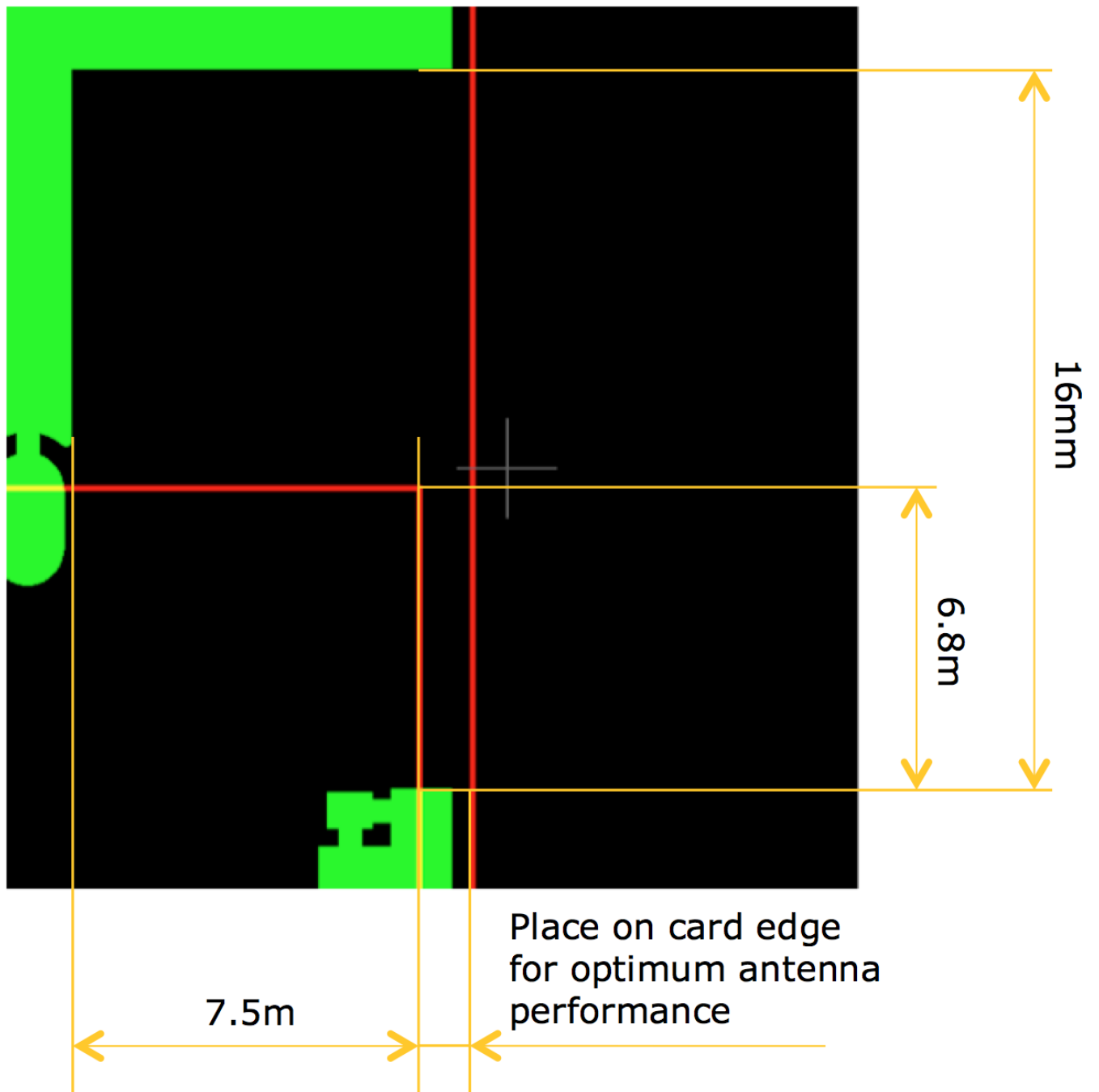
TimoTwo has a supply voltage decoupling on the circuit board. The supply voltage still needs to be adequately filtered. If any disturbance or intermittent communication failures occur, as one of the trouble shooting steps - check the supply voltage for drop-outs, switch supply ripple etc.

4.2.3 Layout Example

1. The TOP layer inside the footprint must be free from copper. There is a ground plane on TimoTwo, but there are also supply lines. It is an unnecessary risk to rely on solder mask lacquer for isolation.
2. An area around the board edge must be kept clear from copper on all layers as depicted in the picture below.



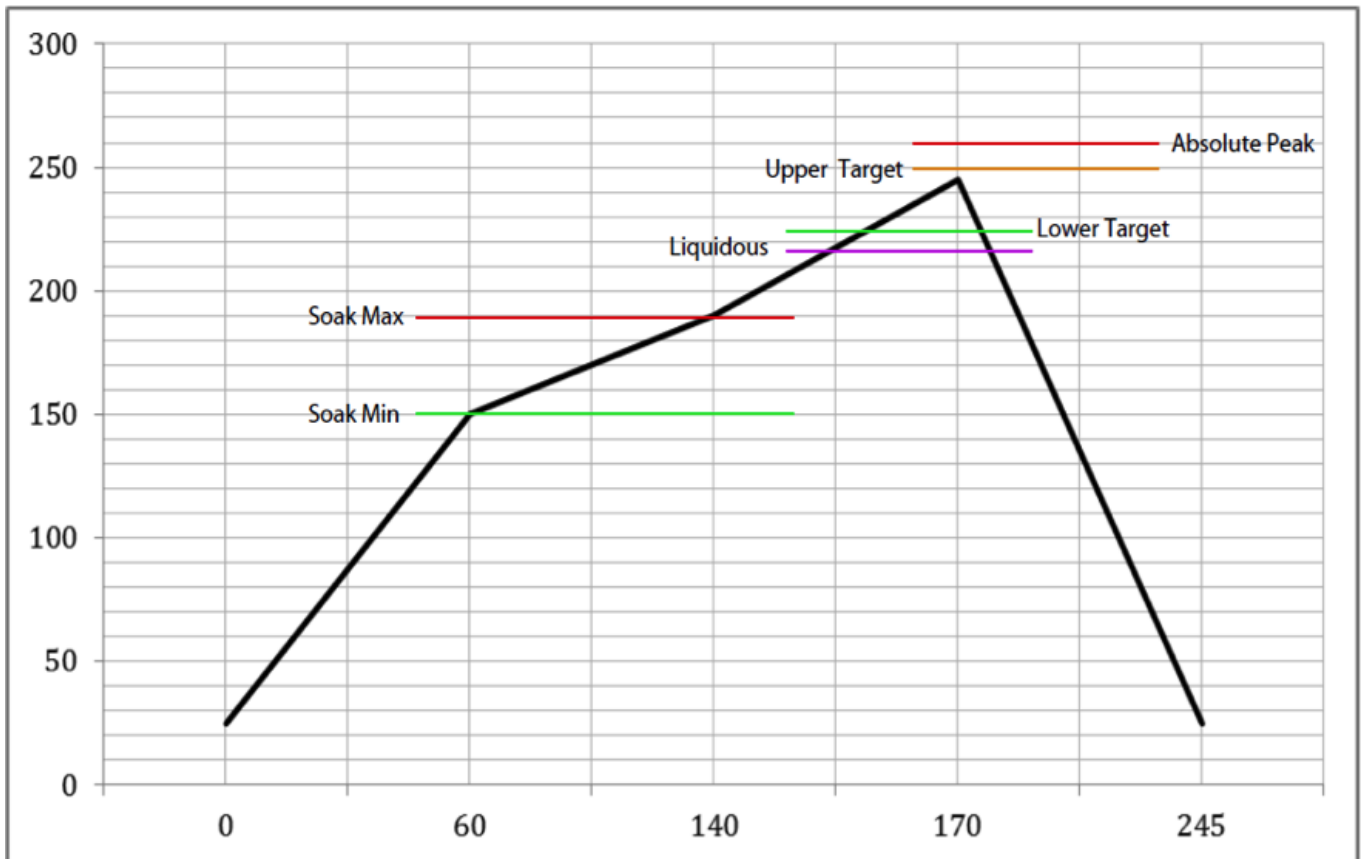
Minimum dimensions for ground plane clearance for optimum performance are shown below:



5. Reflow Soldering

5.1 TimoTwo reflow soldering specification

TimoTwo is a surface mounted device (SMD) designed to be easily integrated into high-volume production lines including reflow soldering to a PCB. It is ultimately the responsibility of the customer to choose the appropriate solder paste and to ensure oven temperatures during reflow meet the requirements of the solder paste. The TimoTwo module conforms to JSTD-020D1 standards for reflow temperatures.



Temperatures should not exceed the minimums or maximums presented in the table below

Specification	Value	Unit
Temperature Inc./Dec. Rate (max)	3	°C / sec
Temperature Decrease rate (goal)	2-3	°C / sec
Soak Temp Increase rate (goal)	0.5 – 1.0	°C / sec
Flux Soak Period (min)	70	sec
Flux Soak Period (max)	120	sec
Flux Soak Temp (min)	150	°C
Flux Soak Temp (max)	190	°C
Time Above Liquidous (max)	70	sec
Time Above Liquidous (min)	50	sec
Time In Target Reflow Range (goal)	30	sec
Time At Absolute Peak (max)	5	sec
Liquidous Temperature (SAC305)	218	°C
Lower Target Reflow Temperature	225	°C
Upper Target Reflow Temperature	250	°C
Absolute Peak Temperature	260	°C

6. LED Outputs

6.1 Status LED

The status LED (`STATUS_LED`) indicates the status of TimoTwo module. The LED indicator pin is an output pin capable of sourcing 5mA at the V_{DD} voltage. An appropriate current limiting resistor must be connected in series with the LED. This LED is also used when using the link switch based method of switching flex modes; please refer to [flex mode page](#) [flex-mode/] for details.

6.1.1 Receiver



Constant off (0V): Not linked to any transmitter



Flashing: off (0V) 100 ms / on (V_{DD}) 100 ms: linked to a transmitter, but no active radio link



Flashing: off (0V) 900 ms / on (V_{DD}) 100 ms: Active radio link, no DMX present



Constant on (V_{DD}): Active radio link, DMX data present

6.1.2 Transmitter



Flashing: off (0V) 900 ms / on (V_{DD}) 100 ms: Active radio link, no DMX present



Constant on (V_{DD}): Active radio link, DMX data present



Flashing: off (0V) 100 ms / on (V_{DD}) 100 ms: linking receivers



Flashing: off (0V) 200 ms / on (V_{DD}) 200 ms: unlinking receivers

6.2 Linked

6.2.1 Receiver

The Linked LED (`LINKED`) indicates whether the TimoTwo is linked to a transmitter or if it's available to be linked. High level (V_{DD}) on this pin indicates a linked state; low level (0V) indicates that the TimoTwo module is not linked.

6.2.2 Transmitter

High level (V_{DD}) on this pin indicates an ongoing linking activity; low level (0V) indicates that the TimoTwo module is currently not performing a linking activity.

6.3 RF Link

6.3.1 Receiver

A high level (V_{DD}) on the RF Link LED output (`RF_LINK`) indicates that the receiver is within range from the transmitter it is linked to and that an active radio link from the transmitter is present.

6.3.2 Transmitter

A high level (V_{DD}) on the RF Link LED output (`RF_LINK`) indicates that there is an active radio link.

6.4 DMX LED

The DMX LED (`DMX_LED`) indicates if a valid DMX stream is received. A high level (V_{DD}) indicates that DMX is present, a low level (0V) indicates that no valid DMX is present.

6.5 RDM LED

A high level (V_{DD}) on the RDM LED output (`RDM_LED`) indicates that the TimoTwo is performing RDM activity.

6.6 Radio level

TimoTwo has 5 output signals for controlling radio level LEDs in the form of a bar graph (`RDI_LVL0` - `RDI_LVL4`). Operation of these, and suggestion of LED colors, can be found in the table below. These are only used for receivers.

Signal name	Suggested LED color	On when signal quality
<code>RDI_LVL0</code>	Red	below 10%
<code>RDI_LVL1</code>	Amber / Yellow	above 20%
<code>RDI_LVL2</code>	Green	above 40%
<code>RDI_LVL3</code>	Green	above 60%
<code>RDI_LVL4</code>	Green	above 80%

6.7 RGB LED

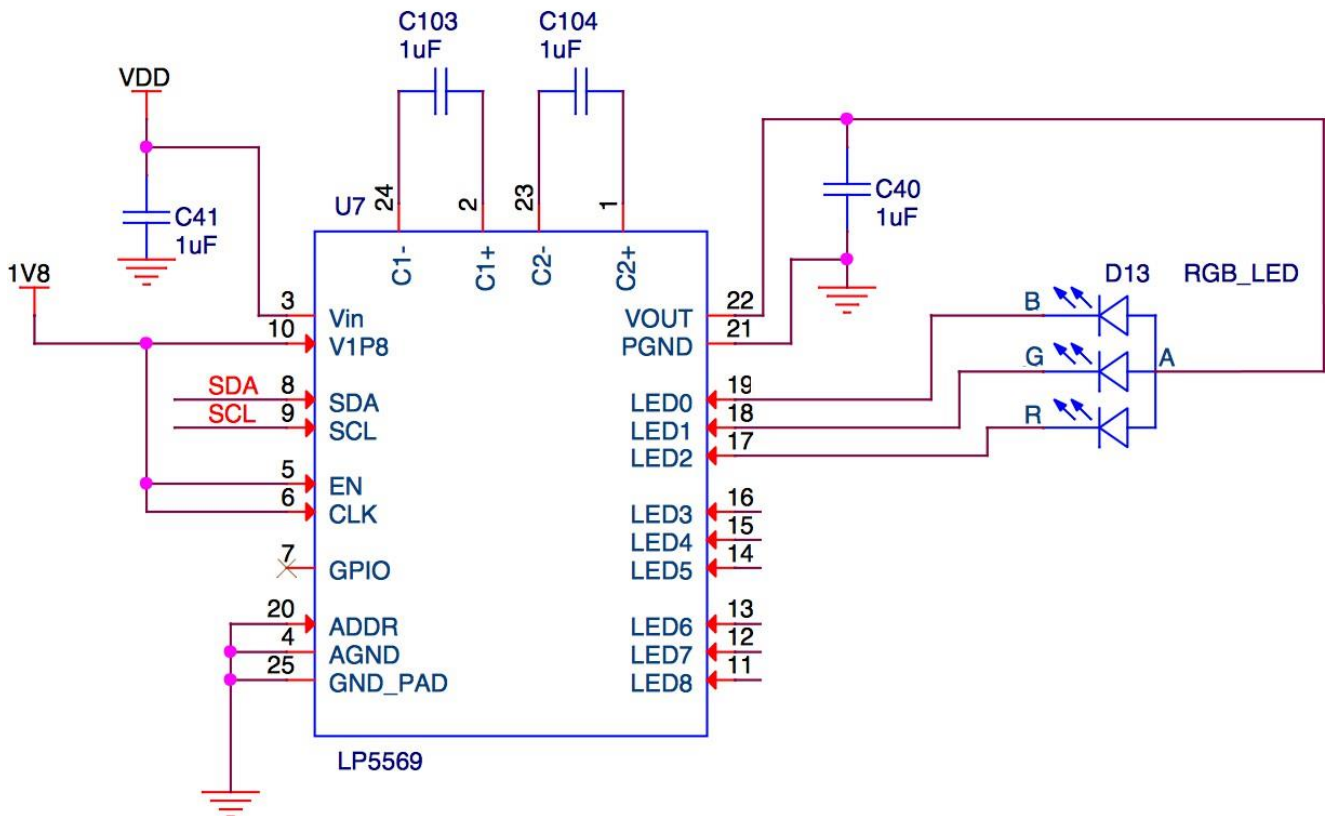
It is possible to connect an external RGB LED driver to the module's I²C interface. The RGB LED will indicate the universe color at the same time as it acts as the `STATUS_LED`, `RF_LINK` or the `LINKED` LEDs, depending on what LED output of the driver IC that is used.

- Output 1 (`LED0- LED2`): Mimics `STATUS_LED`
- Output 2 (`LED3- LED5`): Mimics `RF_LINK` LED
- Output 3 (`LED6- LED8`): Mimics `LINKED` LED

For more information about universe color, see the [universe metadata](#) [#universe-meta/:] section.

Example schematic

RGB LED Driver



7. Link switch

7.1 Link switch input

The link switch input (`LINK_SW`) can be used to interface with a momentary (monostable) closing push button to facilitate a simple user interface. This is an alternative to using the SPI interface to integrate into a host device's menu system.

This signal shall be pulled to V_{DD} using an external 4.7k Ω -10k Ω resistor to ensure proper function.

The switch input has several functions, among others to controlling the radio link or to force driver update mode. Please see the table below for details about the functions of the switch input:

The functions of the link switch button are available through SPI or via the CRMX Toolbox app, and can many times be recommended over using the link switch button for a more user friendly integration. But for minimal integrations, the functions can be used via the switch.

Function	Conditions
Link	Pull signal low (button pressed) for 0.1-1 second. (Only in TX mode.)
Unlink	Hold signal low (button pressed) for >3
seconds. Force firmware update mode	Hold signal low (button pressed)
during power on. Toggle Bluetooth on/off	Short push (0.1-1s) followed by
long push (>3s). Change TX protocol	3x short push (0.1-1s) followed
by long push (>3s). Change RX/TX mode	5x short push (0.1-1s) followed
by long push (>3s). Reset PIN code	7x short push (0.1-1s) followed by long push (>3s).
Enable/disable proxy	9x short push (0.1-1s) followed by long push (>3s). (Only in TX mode, with license.)

8. Linking Key

8.1 Linking receivers

Linking key can be used as a simple way to link receivers to a transmitter without the need to initiate the linking process on the transmitter. This allows the user to just enter the code into the receiver and it will be linked to the transmitter with the same code.

Note: The transmitter must have a linking key set to allow for this.

Note: Receivers can still be linked the traditional way when linking key is used by the transmitter.

8.2 Cloning transmitters

Linking key is also a handy way to clone transmitters for transmitters that support this. By entering an 8 digit linking key code in two, or more, transmitters they will act as if they were the same transmitter.

This allows to have transmitters in separate physical locations and move linked receivers between the locations without the need to re-link the receivers.

Note: Cloned transmitters must not be present in the same physical coverage area as the receivers will not know which one to connect to.

Note: Due to the nature of Linking Key that controls the identity of a radio link, the user should be advised to use "good" key. A "good" key is one that is not easily guessed, or one that is not likely to be used by someone else. Examples of key that are not considered to be good is for instance (but not limited to): 11111111, 22222222, 11223344, 12345678, 10000000, 11000000, etc.

9. Universe metadata

9.1 Universe metadata

CRMX transmitters may transmit some universe metadata information used to identify the received universe on the receiver side. These are;

- **Universe name:** A 16 character string with a human readable name identifying the universe.
- **Universe color:** RGB code for an LED that can easily be used to visually identify the universe by color.

9.1.1 Universe Name

The universe name is an up to 16 characters long textual name of the CRMX universe. When operating in RX mode, this information is available by reading the universe name register. In TX mode the first 16 characters of the `DEVICE_NAME` will be used. Device name can be set either via SPI or from the CRMX Toolbox app.

9.1.2 Universe color

The universe color is a help in identifying what receivers are linked to a particular one transmitter. By selecting a color at the transmitter, either by configuring it via SPI or through the CRMX Toolbox app, this color code is automatically distributed to all linked receivers.

To indicate this color code, an RGB LED can be connected through a RGB LED driver as shown in the [LED outputs](#) [led-outputs/:] section. It is also possible to read this data from the [SPI interface](#) [spi-interface/:] to show in a menu system or similar.

When operating with transmitters that does not support universe color the color will be set to green.

10. SPI Interface

The SPI interface gives access to all features of the TimoTwo module. The interface consists of five digital signals (V_{DD} max):

- **IRQ** – Interrupt signal. Active low, configurable through the interrupt mask register
- **CS** – SPI Chip select, active low
- **SCK** – SPI clock input
- **MOSI** – SPI data input
- **MISO** – SPI data output

10.1 Interface description

10.1.1 Bit and byte order

The data on the SPI bus is clocked with most significant bit first. All multi-byte register data are sent in big-endian byte order (aka. network byte order). In the tables below, bytes are numbered so that byte 0 is the first payload byte being sent on the SPI bus, byte 1 is the second payload byte, etc.

10.1.2 Clock polarity

Data is valid in the low-to-high transition of SCK. This is also known as the clock being active high with valid data on the leading clock edge.

10.1.3 Maximum clock speed

The maximum clock speed supported by TimoTwo is 8MHz. Clock speeds above this limit may result in unexpected behavior.

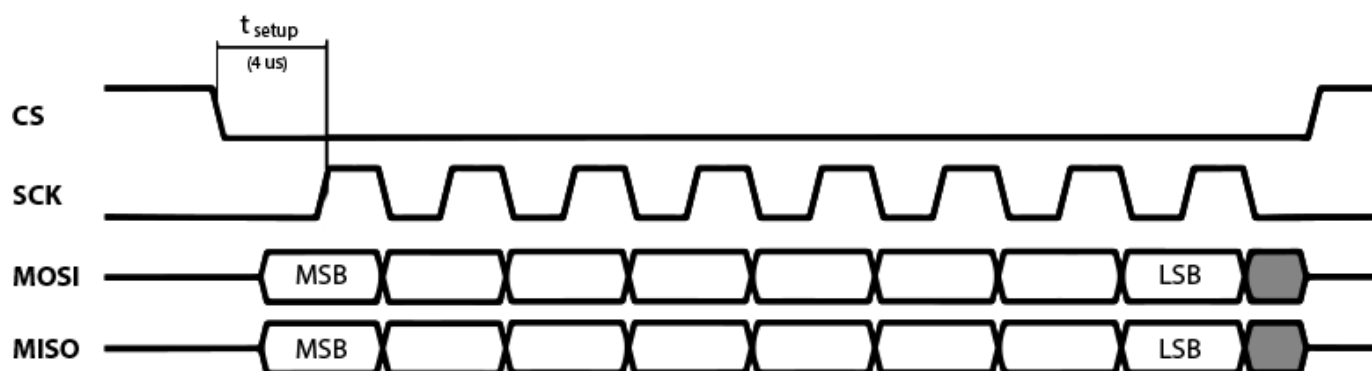
10.1.4 Setup time

The SPI slave unit has a setup time of 4 μ s after the high-to-low transition of the CS signal.

10.2 SPI operation

10.2.1 SPI transactions

All SPI transactions start with a high-to-low transition on the CS pin. The CS pin must be held low during the entire SPI transaction. The `IRQ_FLAGS` register is always shifted out as the first byte of each transaction.



Example SPI transaction

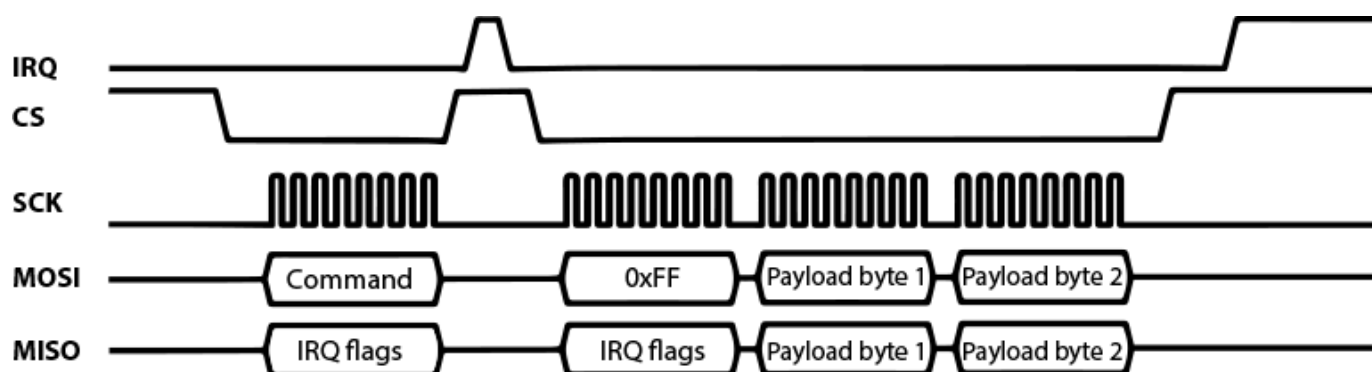
10.2.2 SPI commands

All SPI command sequences, except for the **NOP** command, consist of two SPI transactions. The first transaction shall be one byte long, this is the command byte. The second transaction is the payload. The second transaction must not be started until the TimoTwo module has confirmed the command by a high-to-low transition on the IRQ pin. The first byte being sent to TimoTwo in the second transaction will be ignored, however it is suggested this byte is being sent as 0xFF.

The IRQ signal will stay low during the processing of the command after the payload has been transferred. When the command has been processed, the IRQ signal goes high. It is mandatory to wait for the previous command to finish processing before any consecutive commands are being sent.

See below for an example full SPI command sequence.

NOTE: Bit 7 in the IRQ flags register **MUST** be observed. A 1 in this bit means that the SPI slave module is unable to process the current transaction, and the full command sequence **MUST** be restarted – this means sending the command transaction again.



Example SPI command sequence with a pending IRQ when sequence started

The available SPI commands are listed in the table below.

Command	Binary value	Comment
WRITE_REG	01AA AAAA	Write to a register. AAAAAA = 6 bit register address.
READ_REG	00AA AAAA	Read from a register. AAAAAA = 6 bit register address.
READ_DMx	1000 0001	Read the latest received DMx values from the window set up by the DMx_WINDOW register.
READ_ASC	1000 0010	Read the latest received ASC frame.
READ_RDM	1000 0011	Read the received RDM request (RX mode) or response (TX mode).
READ_RXTX	1000 0100	Read available RXTX interface data.
WRITE_DMx	1001 0001	Write DMx to the internal DMx generation buffer.
WRITE_RDM	1001 0010	Write an RDM response (RX mode) or RDM request (TX mode).
WRITE_RXTX	1001 0011	Write RXTX interface data.
RADIO_DISCOVERY	1010 0000	Write Radio DUB command.
RADIO_DISCOVERY_RESULT	1010 0001	Read Radio DUB result.
RADIO_MUTE	1010 0010	Write Radio mute command.
RADIO_MUTE_RESULT	1010 0011	Read Radio mute response.
RDM_DISCOVERY	1010 0100	Write RDM DUB command.
RDM_DISCOVERY_RESULT	1010 0101	Read RDM DUB result.
NODE_QUERY	1010 0110	Write radio node query command.
NODE_QUERY_RESPONSE	1010 0111	Read radio node query response.
NOP	1111 1111	No operation. Can be used as a shortcut to read the IRQ_FLAGS register.

For more information on the RDM commands, see the [RDM](#) [#rdm/:] section.

10.3 Registers

All reserved bits in the registers shall be considered reserved for future use - don't care when read,

write as 0. Bits in registered is numbered so that bit 0 is least significant bit.

Do not read or write undefined registers – doing so could result in undefined behavior.

Address (hex)	Mnemonic	Description	Comment
0x00	CONFIG [#spi-interface/:config-register-0x00]	Configuration register	Write may cause reboot
0x01	STATUS [#spi-interface/:status-register-0x01]	Device status register	
0x02	IRQ_MASK [#spi-interface/:irq_mask-register-0x02]	IRQ mask register	
0x03	IRQ_FLAGS [#spi-interface/:irq_flags-register-0x03]	IRQ flags register	
0x04	DMX_WINDOW [#spi-interface/:dmx_window-register-0x04]	DMX windows configuration register	
0x05	ASC_FRAME [#spi-interface/:asc_frame-register-0x05]	Alternate start code frame information register	
0x06	LINK_QUALITY [#spi-interface/:link_quality-register-0x06]	Wireless link signal quality register	
0x08	DMX_SPEC [#spi-interface/:dmx_spec-register-0x08]	DMX parameter register for internal DMX generation	
0x09	DMX_CONTROL [#spi-interface/:dmx_control-register-0x09]	Internal DMX generation control register	
0x0A	EXTENDED_IRQ_MASK [#spi-interface/:extended_irq_mask-register-0x0a]	Extended IRQ mask register	
0x0B	EXTENDED_IRQ_FLAGS [#spi-interface/:extended_irq_flags-register-0x0b]	Extended IRQ flags register	
0x0C	RF_PROTOCOL [#spi-interface/:rf_protocol-register-0x0c]	Transmission protocol selection	Write may cause reboot
0x0D	DMX_SOURCE [#spi-interface/:dmx_source-register-0x0d]	DMX source information register	
0x0E	LOLLIPOP [#spi-interface/:lollipop-register-0x0e]	Lollipop counter register	

Address (hex)	Mnemonic	Description	Comment
0x10	VERSION [#spi-interface/:version-register-0x10]	Version register	
0x11	RF_POWER [#spi-interface/:rf_power-register-0x11]	RF power configuration register	
0x12	BLOCKED_CHANNELS [#spi-interface/:blocked_channels-register-0x12]	Channel blocking configuration register	
0x20	BINDING_UID [#spi-interface/:binding_uid-register-0x20]	RDM binding UID register	
0x21	LINKING_KEY [#spi-interface/:linking_key-register-0x21]	Linking key register	Write may cause reboot
0x30	STATUS [#spi-interface/:status-register-0x30]	Status register	Write may cause reboot
0x31	PIN [#spi-interface/:pin-register-0x31]	PIN code register	Write may cause reboot
0x32	BATTERY [#spi-interface/:battery-register-0x32]	Battery status register	
0x33	UNIVERSE_COLOR [#spi-interface/:universe_color-register-0x33]	Universe color register	
0x34	OEM_INFO [#spi-interface/:oem_info-register-0x34]	OEM device information register	
0x35	RXTX_STATUS [#spi-interface/:rxtx_status-register-0x35]	Status register for the RX/TX service	
0x36	DEVICE_NAME [#spi-interface/:device_name-register-0x36]	Device name register	
0x37	UNIVERSE_NAME [#spi-interface/:universe_name-register-0x37]	Universe name register	
0x3D	INSTALLED_OPTIONS [#spi-interface/:installed_options-register-0x3d]	Installed options register	
0x3F	PRODUCT_ID [#spi-interface/:product_id-register-0x3f]	Product ID register	

10.3.1 CONFIG register (0x00)

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7	RADIO_ENABLE	R/W	1	Enable wireless operation. 0 = Disabled 1 = Enabled
	6-4	Reserved	-	-	Reserved for future use
	3	SPI_RDM	R/W	-	0 = UART RDM is used 1 = SPI RDM is used
	2	Reserved	-	-	Reserved for future use
	1	RADIO_TX_RX_MODE	R/W	-	0 = Receiver 1 = Transmitter
	0	UART_EN	R/W	1	Enable UART output of DMX frames. 0 = Disabled 1 = Enabled

RADIO_ENABLE

When disabling wireless operation, this only affects the wireless DMX operation, Bluetooth is independently controlled via `STATUS` register.

SPI_RDM

In receiver mode only. Selects if SPI or UART shall be used for RDM commands.

RADIO_TX_RX_MODE

Writing this bit changes between RX and TX mode.

Note: When mode is changed from RX to TX or vice versa, the module will restart and all registers will revert to their respective reset values.

UART_EN

This bit enables or disables the UART input/output of DMX and RDM data.

10.3.2 STATUS register (0x01)

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7	UPDATE_MODE	R	0	0 = module operational 1 = In software update mode
	6-4	Reserved	-	-	Reserved for future use
	3	DMX	R	0	0 = No DMX available 1 = DMX available
	2	IDENTIFY	R/W	0	0 = No identify 1 = RDM identify active
	1	RF_LINK	R/W	0	0 = No radio link 1 = Active radio link In TX mode, write 1 to start linking
	0	LINKED	R/W	-	0 = Not linked 1 = Linked to TX (or pairing) Write 1 to unlink

UPDATE_MODE

This bit shows if the module is in update mode. When in update mode, some functions are limited.

DMX

In receiver mode this bit indicates that DMX data is being received either over the wireless DMX link.

In transmitter mode this bit indicates that DMX data is being received either via the UART interface, via if the internal DMX from SPI generation is active.

To get the current data source, read the `DMX_SOURCE` register.

IDENTIFY

In transmitter mode, writing 1 will start identifying all linked receivers, writing 0 will stop

identifying. In receiver mode this is a read-only bit. Reading 1 means this device shall identify.

RF_LINK

This bit shows the status of the wireless DMX link.

This bit is read-only in RX mode, writing 1 to this bit in TX mode starts a linking process.

LINKED

This bit shows if the module is linked to a TX or not. Writing 1 to this bit performs an unlink operation.

10.3.3 IRQ_MASK register (0x02)

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7	Reserved	-	-	Reserved for future use
	6	EXTENDED_IRQ_EN	RW	0	Enable extended interrupts
	5	IDENTIFY_IRQ_EN	RW	0	Enable identify device interrupt
	4	ASC_IRQ_EN	RW	0	Enable alternative start code interrupt
	3	RF_LINK_IRQ_EN	RW	0	Enable radio link status change interrupt
	2	DMX_CHANGED_IRQ_EN	RW	0	Enable DMX changed interrupt
	1	LOST_DMX_IRQ_EN	RW	0	Enable loss of DMX interrupt
	0	RX_DMX_IRQ_EN	RW	0	Enable DMX frame reception interrupt

10.3.4 IRQ_FLAGS register (0x03)

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7	SPI_DEVICE_BUSY	R	0	SPI slave device is busy and cannot comply with command. Command sequence MUST be restarted.
	6	EXTENDED_IRQ	R	0	Extended interrupt
	5	IDENTIFY_IRQ	R	0	Identify device state change interrupt
	4	ASC_IRQ	R	0	Alternative start code frame received interrupt
	3	RF_LINK_IRQ	R	0	Radio link status change interrupt
	2	DMX_CHANGED_IRQ	R	0	DMX changed in DMX window interrupt
	1	LOST_DMX_IRQ	R	0	Loss of DMX interrupt
	0	RX_DMX_IRQ	R	0	Complete DMX frame received interrupt

10.3.5 DMX_WINDOW register (0x04)

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7-0	WINDOW_SIZE_MSB	R/W	0x02	DMX windows size (MSB)
1	7-0	WINDOW_SIZE_LSB	R/W	0x00	DMX windows size (MSB)
2	7-0	START_ADDRESS_MSB	R/W	0x00	DMX start address (MSB)
3	7-0	START_ADDRESS_LSB	R/W	0x00	DMX start address (LSB)

WINDOW_SIZE

The size of the DMX window to be used in number of bytes. It shall not span beyond the 512th slot. This is a 16 bit value that is written with MSB in byte 2 of this register, LSB in byte 3.

The reset value is a length of 512 bytes (the full universe).

START_ADDRESS

The DMX start address to be used, zero-indexed. Shall be within the range 0-511. This is a 16 bit value that is written with MSB in byte 0 of this register, LSB in byte 1.

10.3.6 ASC_FRAME register (0x05)

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7-0	ASC_FRAME_LENGTH_MSB	R	-	Length of rec ASC frame (M
1	7-0	ASC_FRAME_LENGTH_LSB	R	-	Length of rec ASC frame (L
2	7-0	START_CODE	R	-	Start code of received ASC frame

Use `READ_ASC` command to read the ASC frame data after reading this register.

10.3.7 LINK_QUALITY register (0x06)

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7-0	PDR	R	-	Packet delivery rate

PDR

Packet delivery rate to be displayed as percent: 0 = 0%, 255 =

100%. This register is only available in RX mode.

10.3.8 DMX_SPEC register (0x08)

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7-0	N_CHANNELS_MSB	R/W	0x02	Number of slots/channels to generate (MSB)
1	7-0	N_CHANNELS_LSB	R/W	0x00	Number of slots/channels to generate (LSB)
2	7-0	INTERSLOT_TIME_MSB	R/W	0x00	Interslot spacing in μ s (MSB)
3	7-0	INTERSLOT_TIME_LSB	R/W	0x00	Interslot spacing in μ s (LSB)
4	7-0	REFRESH_PERIOD_MSB	R/W	0x00	DMX frame length in μ s (MSB)
5	7-0	REFRESH_PERIOD_B2	R/W	0x00	DMX frame length in μ s
6	7-0	REFRESH_PERIOD_B1	R/W	0x61	DMX frame length in μ s
7	7-0	REFRESH_PERIOD_LSB	R/W	0xA8	DMX frame length in μ s (LSB)

This register is only available in TX mode.

REFRESH_PERIOD

The refresh period is the time between two consecutive BREAK conditions. The frame rate of the generated DMX stream will be $1/\text{REFRESH_PERIOD}$. Defaults to 25000 μ s (refresh rate of 40 Hz). This value is a 32 bit value.

INTERSLOT_TIME

The interslot time is the time between one slot's last stop bit to the next slot's start bit. The interslot time is a 16 bit value, and defaults to 0 μ s.

N_CHANNELS

The number of channels/slots to generate. This is a 16 bit value from 1 to 512. This parameter defaults to 512.

10.3.9 DMX_CONTROL register (0x09)

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7-1	Reserved	-	-	Reserved for future use

0	ENABLE	R/W	0	0 = internal generation disabled 1 = internal generation enabled
---	--------	-----	---	---

This register is only available in TX mode.

ENABLE

Starts or stops the internal DMX generation for DMX data from SPI. The `DMX_SPEC` register should be set up before writing 1 to this bit.

10.3.10 EXTENDED_IRQ_MASK register (0x0A)

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7-0	Reserved	R/W	-	Reserved for use
1	7-0	Reserved	R/W	-	Reserved for use
2	7-0	Reserved	R/W	-	Reserved for use
3	7	NODE_QUERY_EN	R/W	0	Enable node query interrupt (TX mode)
	6	UNIV_META_CHANGED_EN	R/W	0	Enable univer metadata chan interrupt (RX mode)
	5	RDM_DISCOVERY_EN	R/W	0	Enable RDM Discovery int (TX mode)
	4	RADIO_MUTE_EN	R/W	0	Enable Radio interrupt (TX mode)
	3	RADIO_DISCOVERY_EN	R/W	0	Enable Radio Discovery int (TX mode)
	2	RXTX_CTS_EN	R/W	0	Enable RXTX clear to send interrupt
	1	RXTX_DA_EN	R/W	0	Enable RXTX available inter
	0	RDM_IRQ_EN	R/W	0	Enable RDM request interru

10.3.11 EXTENDED_IRQ_FLAGS register (0x0B)

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7-0	Reserved	R/W	-	Reserved for future use
1	7-0	Reserved	R/W	-	Reserved for future use
2	7-0	Reserved	R/W	-	Reserved for future use
3	7	NODE_QUERY	R/W	0	Response from a node query command available (TX mode)
	6	UNIV_META_CHANGED	R/W	0	Universe metadata changed (RX mode)
	5	RDM_DISCOVERY	R/W	0	RDM Discovery command finished (TX mode)
	4	RADIO_MUTE	R/W	0	Radio Mute command finished (TX mode)
	3	RADIO_DISCOVERY	R/W	0	Radio Discovery command finished (TX mode)
	2	RXTX_CTS	R/W	0	RXTX clear to send
	1	RXTX_DA	R/W	0	RXTX data available
	0	RDM_IRQ	R/W	0	RDM request available (RX mode), RDM response available (TX mode).

10.3.12 RF_PROTOCOL register (0x0C)

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7-0	TX_PROTOCOL	R/W	-	0= CRMX 1= W-DMX G3 3= W-DMX G4S

TX_PROTOCOL

In TX mode it is possible to select what wireless protocol to transmit; CRMX, W-DMX G3 or W-DMX G4S.

10.3.13 DMX_SOURCE register (0x0D)

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7-0	DATA_SOURCE	R	-	0= No data 1= DMX port (UART) 2= Wireless DMX link 3 = SPI 4 = /

DATA_SOURCE

This value indicates what data source that is currently active. If there is data active on more than one source, this field returns what source the module has picked based on its priority order.

10.3.14 LOLLIPOP register (0x0E)

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7-0	COUNTER	R/W	0	Lollipop counter

COUNTER

This is a so called lollipop counter. It starts from 0 and is increased by 1 every time it is read. When it reaches 255 it will wrap around down to 128. Hence it will never return to a value below 128 unless the module has been rebooted.

This can be used for detecting module reboots by setting this counter to a value larger than 128 after everything has been initialized. By periodically reading this counter, as soon as the counter return a value less than 128 it is a sign that the module has rebooted.

After the module has been rebooted, non-volatile registers has returned to their specified reset values and should be re-initialized.

10.3.15 VERSION register (0x10)

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7-0	HW_VERSION_B4	R	-	Hardware version (B4)
1	7-0	HW_VERSION_B3	R	-	Hardware version (B3)
2	7-0	HW_VERSION_B2	R	-	Hardware version (B2)
3	7-0	HW_VERSION_B1	R	-	Hardware version (B1)
4	7-0	SW_VERSION_B4	R	-	Software version (B4)
5	7-0	SW_VERSION_B3	R	-	Software version (B3)
6	7-0	SW_VERSION_B2	R	-	Software version (B2)
7	7-0	SW_VERSION_B1	R	-	Software version (B1)

HW_VERSION

The hardware version is written as a 8 digit hexadecimal number according to B4B3B2B1, for example 0000A003.

SW_VERSION

The software version is written as decimal numbers with points in between, like B4.B3.B2.B1, for example 1.0.7.2 or 1.0.5.35.

10.3.16 RF_POWER register (0x11)

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7-0	OUTPUT_POWER	RW	2	RF Output power in transmitter mode

OUTPUT_POWER

When in transmitter mode, the output power can be controlled by writing this register. The reset value is 18dBm, which results in ETSI compliant 100mW when using a 2dBi antenna.

Value	Output power from 2dBi antenna
2	20 dBm / 100 mW
3	16 dBm / 40 mW
4	11 dBm / 13 mW
5	5 dBm / 3 mW

10.3.17 BLOCKED_CHANNELS register (0x12)

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7-0	FLAGS	R/W	0	Channel 7-0
1	7-0	FLAGS	R/W	0	Channel 15-8
2	7-0	FLAGS	R/W	0	Channel 23-16
3	7-0	FLAGS	R/W	0	Channel 31-24
4	7-0	FLAGS	R/W	0	Channel 39-32
5	7-0	FLAGS	R/W	0	Channel 47-40
6	7-0	FLAGS	R/W	0	Channel 55-48
7	7-0	FLAGS	R/W	0	Channel 63-56
8	7-0	FLAGS	R/W	0	Channel 71-64
9	7-0	FLAGS	R/W	0	Channel 79-72
10	7-0	FLAGS	R/W	0	Channel 87-80

FLAGS

When in transmitter mode, the transmission channels can be blocked. TimoTwo always employ LumenRadio's patented Cognitive Coexistence technology, but still allow channels to be blocked completely. One channel is 1MHz wide, and the transmission channels are 2402-2480MHz, represented by channel numbers 2-80.

To block a channel, write a **1** to the corresponding bit in the BLOCKED_CHANNELS register. Writing a **0** enables the channel to be used again.

Examples:

- Channel 2 is represented by bit 2 in byte 0
- Channel 14 is represented by bit 6 in byte 1
- Channel 80 is represented by bit 0 byte 10

Block flags written to bits corresponding to channels 0-1 and 81-87 will always be ignored.

At any given point at least 16 channels must be enabled. If less than 16 channels are used, TimoTwo will reset this register and allow transmission on all channels.

Note: For best performance, do not block more than half of the available channels. If blocking more channels than are allowed by certification requirements this register will be reset to all **0**.

10.3.18 BINDING_UID register (0x20)

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7-0	UID_MSB	R/W	0	RDM UID of the host device (MSB)
1	7-0	UID_B4	R/W	0	RDM UID of the host device
2	7-0	UID_B3	R/W	0	RDM UID of the host device
3	7-0	UID_B2	R/W	0	RDM UID of the host device
4	7-0	UID_B1	R/W	0	RDM UID of the host device
5	7-0	UID_LSB	R/W	0	RDM UID of the host device (LSB)

UID

The binding UID is the fixture's UID that contain the TimoTwo module. This must be used whe using RDM over SPI. It is also recommended to use this when using RDM over UART. This binding UID corresponds to the binding UID in the E1.20 (RDM) standard, and may be used by smart RDM controllers to group devices together through the binding UID when RDM devices are located in the same physical device.

10.3.19 LINKING_KEY register (0x21)

This register has two different layouts depending on if operating in TX or RX mode. For security reasons, this register can not be read. **Note:** Writing this register causes the module to restart and hence cause all non-volatile registers to revert to their respective reset values. **RX mode**

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7-0	KEY1	W	-	First digit of Linking key
1	7-0	KEY2	W	-	2nd digit of Linking key
2	7-0	KEY3	W	-	3rd digit of Linking key
3	7-0	KEY4	W	-	4th digit of Linking key
4	7-0	KEY5	W	-	5th digit of Linking key
5	7-0	KEY6	W	-	6th digit of Linking key
6	7-0	KEY7	W	-	7th digit of Linking key
7	7-0	KEY8	W	-	Last digit of Linking key
8	7-0	MODE	W	-	What mode to run in 0 = CRMX Classic 1= CRMX ²
9	7-0	OUTPUT	W	-	Output number to link to (0 - 7)

TX mode

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7-0	KEY1	W	-	First digit of Linking key
1	7-0	KEY2	W	-	2nd digit of Linking key
2	7-0	KEY3	W	-	3rd digit of Linking key
3	7-0	KEY4	W	-	4th digit of Linking key
4	7-0	KEY5	W	-	5th digit of Linking key
5	7-0	KEY6	W	-	6th digit of Linking key
6	7-0	KEY7	W	-	7th digit of Linking key
7	7-0	KEY8	W	-	Last digit of Linking key

KEY

The key is an 8 digit code encoded into 8 bytes. For example 12345678 is written as 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08.

To remove a custom linking key in TX mode, write 8 bytes of 255 (0xFF) as a key. This will revert back to the module's factory default key.

MODE

To link properly the mode of the transmitter needs to be written.

- 0x00: CRMX Classic
- 0x01: CRMX²

Only available in RX mode.

OUTPUT

This field corresponds to the outputs of Stardust or similar multi-universe transmitters. If linking to the first output of a Stardust (Output A), write 0x00 to this field, 0x01 for Output B, etc. *Note:* In CRMX Classic mode, only every second value (0x00, 0x02, 0x04, 0x06) is available.

If linking to a single universe transmitter, for instance Aurora, only universe 0x00 and CRMX Classic mode is available. Only available in RX mode.

Read more in the [Linking Key](#) [#linking-key/:] section.

10.3.20 BATTERY register (0x32)

The BATTERY register provides a way for a battery driven device to send the battery level . This will be shown in the CRMX Toolbox app, or other apps that supports it.

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7-0	BATTERY_LEVEL	W	255	Battery level in percent.

BATTERY_LEVEL

This value will be reported via the standard BLE Battery service.

- 0 = 0%, 100 = 100%
- 101 - 254 are reserved values
- 255 = no battery available

10.3.21 UNIVERSE_COLOR register (0x33)

This register is read-only in RX mode and R/W in TX mode.

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7-0	RED	R/W	-	Red level (0 - 255)
1	7-0	GREEN	R/W	-	Green level (0 - 255)
2	7-0	BLUE	R/W	-	Blue level (0 - 255)

RED/GREEN/BLUE

These bytes represent a 24 bit RGB value for the Universe color. In TX mode the written value will be propagated to all receivers that supports in. In RX mode this will show the color value that has been received from the transmitter. See the [universe metadata](#) [#universe- meta/:] section for more information.

10.3.22 OEM_INFO register (0x34)

The **OEM_INFO** register gives the manufacturer the possibility to let the CRMX Toolbox app, or other apps that supports it, show custom images etc in the connection list. This will make it easier for the end-user to distinguish different products from each other.

For more details about getting images added to the CRMX Toolbox app, please reach out to LumenRadio support.

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7-0	DEVICE_MODEL_ID_MSB	R/W	-	Device model I (MSB)
1	7-0	DEVICE_MODEL_ID_LSB	R/W	-	Device model I (LSB)
2	7-0	MANUFACTURER_ID_MSB	R/W	-	Manufacturer I (MSB)
3	7-0	MANUFACTURER_ID_LSB	R/W	-	Manufacturer I (LSB)

DEVICE_MODEL_ID

This is the manufacturer's 16 bit device model ID as per the E1.20 (RDM) standard. Together with the Manufacturer ID it uniquely identifies a certain model from a certain manufacturer.

MANUFACTURER_ID

This is the manufacturer's ESTA registered 16 bit manufacturer ID.

10.3.23 RXTX_STATUS register

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7-2	Reserved	-	-	Reserved for future use
	1	CLEAR_TO_SEND	R	1	Clear to write data to the RXTX interface
	0	DATA_AVAILABLE	R	0	There are data available to read in the RXTX interface

10.3.24 DEVICE_NAME register (0x36)

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7-0	DEVICE_NAME1	R/W	-	Device name 1st character
1	7-0	DEVICE_NAME2	R/W	-	Device name 2nd character
...	...	...	R/W	-	Device name character 3 - 30
30	7-0	DEVICE_NAME31	R/W	-	Device name 31st character
31	7-0	DEVICE_NAME32	R/W	-	Device name 32nd character

DEVICE_NAME

The device name is a 32 character ASCII string. It is also used as the device name in the Toolbox app, as well as

`DEVICE_LABEL` in RDM. For strings shorter than 32 characters the string shall be NULL terminated.

In TX mode the first 16 characters from this string will be transmitted as the universe name. For more information about universe name, see the [universe metadata](#) [#universe-meta/:] section.

10.3.25 UNIVERSE_NAME register (0x37)

This register is only available in RX mode.

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7-0	UNIVERSE_NAME1	R	-	Device name 1st character
1	7-0	UNIVERSE_NAME2	R	-	Device name 2nd character
...	...	...	R	-	Device name character 3 - 14
14	7-0	UNIVERSE_NAME15	R	-	Device name 15st character
15	7-0	UNIVERSE_NAME16	R	-	Device name 16st character

UNIVERSE_NAME

Universe name is a 16 character ASCII string sent from a transmitter to indentify a transmitted universe as a human readable name. For strings shorter than 16 characters the string will be NULL terminated.

For more information about universe name, see the [universe metadata](#) [#universe-meta/:] section.

10.3.26 INSTALLED_OPTIONS register (0x3D)

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7-0	N_OPTIONS	R	-	Number of installed options (0 - 6)
1	7-0	OPTION1_MSB	R	-	Installed option #1 ID (MSB)
2	7-0	OPTION1_LSB	R	-	Installed option #1 ID (LSB)
...	...	...	R	-	Option 2-5...
11	7-0	OPTION6_MSB	R	-	Installed option #6 ID (MSB)
12	7-0	OPTION6_LSB	R	-	Installed option #6 ID (LSB)

N_OPTIONS

This byte indicate how many options that has been installed and will be available in the following list.

OPTION1 - OPTION6

This is a list of `N_OPTIONS` 16 bit elements showing what options that has been installed. The available options are:

Option ID	Description
0x2001	RDM TX via SPI
0x2002	RDM TX via Proxy

10.3.27 PRODUCT_ID register (0x3F)

Byte	Bit	Mnemonic	Type	Reset value	Description
0	7-0	ID0	R	-	Product ID byte 0
1	7-0	ID1	R	-	Product ID byte 1
2	7-0	ID2	R	-	Product ID byte 2
3	7-0	ID3	R	-	Product ID byte 4

ID

The product ID for TimoTwo is 0xF1, 0x40, 0x00, 0x00.

10.4 Interrupts

The IRQ pin is used to indicate that there is one (or more) pending interrupt that has been enabled through the IRQ_MASK register. The IRQ pin is also used to indicate that the SPI slave is ready to receive the second transaction of an ongoing SPI command sequence.

The IRQ pin will always go high (inactive) after a successful SPI transaction. If any interrupts are pending, or when the chip is ready for the second transaction in a SPI command sequence it will be indicated through a high-to-low transition on the IRQ pin.

10.4.1 RX_DMx_IRQ

Asserted when a complete DMX frame has been received. Cleared by issuing a READ_DMx command sequence. Only available for receiver modules and TimoTwo FX in receiver mode.

10.4.2 LOST_DMx_IRQ

Asserted when DMX stream is lost. This may be an effect of losing radio link, or if DMX stream in to the transmitter is terminated (for instance the DMX cable to the transmitter is unplugged). Cleared by reading the STATUS register.

Only available for receiver modules and TimoTwo FX in receiver mode.

10.4.3 DMX_CHANGED_IRQ

Asserted when a complete DMX frame has been received and any slot within the DMX window has changed value. Cleared by issuing a READ_DMx command sequence.

Only available for receiver modules and TimoTwo FX in receiver mode.

10.4.4 RF_LINK_IRQ

Asserted whenever the state of the radio link has changed. This may be:

- radio link is lost
- radio link is established
- receiver got paired to transmitter
- receiver got unpaired from

transmitter Cleared by reading the

STATUS register.

Only available for receiver modules and TimoTwo FX in receiver mode.

10.4.5 ASC_IRQ

Asserted when a complete ASC frame has been received. Cleared by reading the ASC_FRAME

register. Only available for receiver modules and TimoTwo FX in receiver mode.

10.4.6 IDENTIFY_IRQ

Asserted when TimoTwo is being told to start or stop it's identify device procedure. For more information about identify, please refer to "ANSI E1.20 - 2006 / Entertainment Technology-RDM-Remote Device Management over USITT DMX512 Networks" specification.

This bit is cleared by reading the STATUS register.

10.4.7 EXTENDED_IRQ

Asserted there is an IRQ in the extended IRQ range. Read the EXTENDED_IRQ_FLAGS register to obtain further details.

10.5 Extended IRQ_MASK

The EXTENDED_IRQ registers are used to control and read the extended interrupts as detailed below.

10.5.1 RDM

RX mode Asserted when there is an RDM request available to read. Cleared by issuing the READ_RDM command or if RDM request has timed out.

TX mode Asserted when there is an RDM response available, or when a request has timed out. Cleared by issuing the READ_RDM command.

10.5.2 RXTX_DA

Asserted when there is data available to read with the READ_RXTX command. Cleared by reading the data or by reading the RXTX_STATUS register.

10.5.3 RXTX_CTS

Asserted when it's safe to send more data to the RXTX interface. Cleared by writing data with the WRITE_RXTX command or by reading the RXTX_STATUS register.

10.5.4 UNIV_META_CHANGED

Asserted when universe color or universe name changes in RX mode. Cleared by reading any of those meta data registers.

10.5.5 RADIO_DISCOVERY

Asserted when a radio discover unique branch command has been completed. Cleared by reading the discovery result through the RADIO_DISCOVERY_RESULT command.

10.5.6 RADIO_MUTE

Asserted when a radio mute command has been completed. Cleared by reading the result through the RADIO_MUTE_RESPONSE command.

10.5.7 RDM_DISCOVER

Asserted when a RDM discover unique branch command has been completed. Cleared by reading the discovery result through the RDM_DISCOVERY_RESULT command.

10.5.8 NODE_QUERY

Asserted when a node query response is available or when the query has timed out. Cleared by reading the discovery result through the NODE_QUERY_RESPONSE command.

11. DMX Interface

This chapter explains the two separate DMX interfaces available on the TimoTwo module.

11.1 SPI

DMX data is available to be read over SPI when in receiver mode. This applies to both Null Start Code (NSC) data and Alternate Start Code (ASC) data.

NSC data can be written via SPI in transmitter mode.

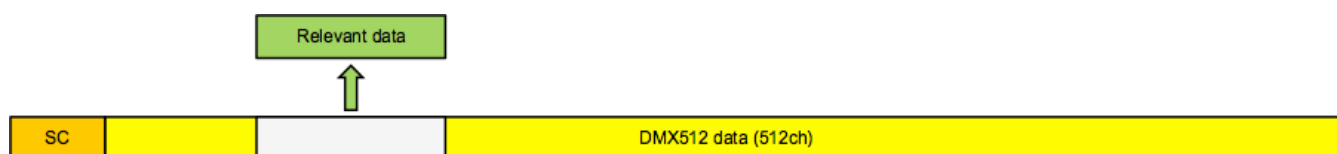
11.1.1 DMX window

The DMX window feature allows a host CPU to set up a span of DMX slots (aka. a DMX window) that the host is interested of. This will reduce the load of the host since it does not need to buffer and parse the entire DMX frame.

Instead the host can get an interrupt request (`DMX_CHANGED_IRQ`) from TimoTwo whenever data has changed inside the DMX window.

`RX_DMX_IRQ` is not affected by the settings of the DMX window.

DMX window functionality is only available in receiver modules and TimoTwo FX operating in receiver mode.



11.1.2 Reading DMX data over SPI (RX mode)

When reading DMX data over SPI, the longest block of data possible to read is 128 bytes. If it is required to read more than 128 bytes this must be done by performing multiple consecutive `READ_DMX` command sequences.

The internal data block counter is reset when the end of the DMX window is reached, or if any other command is being sent to the SPI slave.

11.1.3 Writing DMX over SPI (TX mode)

First the correct DMX specification needs to be configured in the `DMX_SPEC` register, data can then be written using the `WRITE_DMX` command. Data is written in blocks, no more than 128 slots at a time. To write a full universe of 512 slots four consecutive calls to `WRITE_DMX` with 128 slots each must be made.

It is possible to write smaller blocks, but not larger than 128 slots.

The internal data block counter is reset when any other command is being sent to the SPI slave.

11.2 UART DMX/RDM interface

The UART DMX/RDM interface of the TimoTwo module consist of 4 digital signals that can be used to interface an RS485 driver IC compliant with the ANSI E1.11 DMX512-A standard to facilitate a DMX512-A compatible interface. Please refer to the example schematic in the [pin assignment](#) [#module/:] section for details on how to connect an RS485 driver IC.

The DMX interface can also be used for CMOS/TTL level directly interfacing, for instance to a host CPU.

NOTE: Signal on RXD pin must **NOT** exceed V_{DD} ! If 5V signal is used, a level shifting circuit must be used. Please see example schematics in the [pin assignment](#) [#module/:] section for details on how to use a 5V IC.

11.3 DMX and RDM termination and line bias

DMX and RDM termination and line bias circuitry is not provided as part of TimoTwo (since the data is provided at TTL level). This circuit is left to the device manufacturer to provide as required for each particular application and device.

Termination and line bias circuitry requirements shall follow "ANSI E1.20 - 2006 / Entertainment Technology-RDM-Remote Device Management over USITT DMX512 Networks" or later revisions.

IMPORTANT: Biasing is **mandatory** for all RDM implementations.

11.4 DMX frame rate and size

TimoTwo will auto sense the DMX frame rate and frame size and accept all variations that are within the USITT DMX-512 (1986 & 1990) and DMX-512-A standards.

Minimum DMX frame size is 1 slot and maximum is 512 slots.

Minimum DMX frame rate for normal operation is 0.8 frames per second and maximum is 830 frames per second.

Input frame rates below 0.8 frames per second, i.e. more than 1.25s has elapsed since the start of the last frame, will be treated as a loss of DMX.

TimoTwo modules has a configurable loss of DMX behaviour, but the default behaviour is that it will set the RS485 driver IC to a high- impedance/tri-state mode until another DMX frame is detected.

CRMX will propagate DMX through the system maintaining the input frame rate and frame size with the exception of frame rates that exceed those allowed by the DMX 512-A standard.

Input DMX frame rates above 830 frames per second will propagate through the system at 830 frames per second to ensure that the DMX output is compliant with the DMX512-A standard.

11.5 DMX start code frames

DMX packets with start codes other than the DMX default 0x00 (also known as the Null Start Code, or NSC) and the RDM start code (0xCC) will be propagated through the system, and are subject to the same rules and limitations as the null start code packets. Such frames are called Alternate Start Code, or ASC, frames.

11.6 RDM start code frames

Frames with RDM start code (0xCC) are handled separately by transmitters in CRMX systems, as part of the proxy functionality. Transmitters manage the interleaving of RDM frames with null start code packets across the air, and may interleave other RDM frames that are needed to manage the proxy functionality. This may result in RDM frames can appear on the DMX/RDM interface in a different order than on the input of the transmitter.

All RDM frames are handling in compliance with the PLASA E1.20 standard.

TimoTwo FX in transmitter nodes discards all frames with RDM start code (0xCC) unless the proxy functionality is enabled (available through separate license). RDM draft start code (0xF0) is always discarded.

11.7 Alternate start code frames

ASC (Alternate Start Code) frames can be read separately from the SPI interface or the DMX/RDM interface. Over SPI, the `ASC_FRAME` register contains basic information about the last received ASC frame. The information available in this register is start code and length (number of slots).

11.7.1 Reading ASC data over SPI

When reading ASC data over SPI, the longest block of data possible to read is 128 bytes. If reading more than 128 bytes is required, this is done by performing multiple consecutive `READ_ASC` command sequences.

The internal data block counter is reset when the end of the ASC frame is reached, or if any other command is being sent to the SPI slave.

12. RDM Interface

12.1 RDM data service

Name	Description
UUID	bff00000-247d-6e1c-448c-223dfa0bd00c

The RDM service is always available on TimoTwo when in receiver mode. In transmitter mode, it requires a software license. For pricing and licensing details, please contact [LumenRadio sales](mailto:sales@lumenradio.com) [mailto:sales@lumenradio.com].

Note: When RDM activity is initiated via this service, it takes priority over any other RDM traffic.

12.1.1 Discovery Start characteristic

Name	Description
UUID	bff00001-247d-6e1c-448c-223dfa0bd00c
Read access	no
Write access	yes
Notifications	no
Max write size	1 byte

Write command to this characteristic to start or stop the discovery process. Once initiated, the discovery process will run until completed and self-terminate.

Available commands

Value	Note
0	Terminate pending discovery process
1	Trigger discovery process

12.1.2 Discovery Status characteristic

Name	Description
UUID	bff00002-247d-6e1c-448c-223dfa0bd00c
Read access	yes
Write access	no
Notifications	yes
Max write size	3 bytes

Read this characteristic to obtain the current status of the discovery process and the number of discovered devices. Notifications on changes is supported.

Response data

Byte	Note
------	------

1	Discovery in progress
2	Number of discovered devices (MSB)
3	Number of discovered devices (LSB)

Name	Description
------	-------------

12.1.3 List of Devices characteristic

UUID	bff00003-247d-6e1c-448c-223dfa0bd00c
Read access	yes
Write access	yes
Notifications	no
Max write size	variable, up to 247 bytes or negotiated MTU

Read this to obtain the list of discovered UUIDs.

Write this characteristic to `index` to offset the the starting index of the fragmented list.

Response data

Byte	Note
1	Start index for the list (MSB)
2	Start index for the list (LSB)
3	Number of discovered devices (MSB)
4	Number of discovered devices (LSB)
5 - X	List of discovered devices (Node type [#interface/:node-type])

NODE TYPE

Byte	Note
1	Index to parent node (MSB)
2	Index to parent node (LSB)
3 - 9	UUID of the device, see E1.20 - 5.2 UUID Text Representation

This type is used to describe basic hierarchy of the discovered devices. A device directly linked or connected to the master will default to

`0xFFFF`.

12.1.4 RDM TX characteristic

Name	Description
UUID	bff00004-247d-6e1c-448c-223dfa0bd00c
Read access	no
Write access	yes
Notifications	no
Max write size	variable, up to 247 bytes or negotiated MTU

Write RDM request packet to this characteristic to send it to the downstream device.

If the RDM request packet exceeds the MTU, it must be fragmented and written in sequential chunks as described in section [Packet fragmentation](#) [#interface/:packet-fragmentation].

Note: All writes to this characteristic while RDM status characteristic is reporting `RDM_IN_PROGRESS` will be ignored.

Packet fragmentation

For detailed packet structure description, see E1.20 - 6 Message Structure.

Byte	Name	Description
1	SC_RDM	START Code <code>0xCC</code> of the RDM request packet
2	SC_SUB_MESSAGE	Sub START Code <code>0x01</code> of the RDM request packet
3	Message Length	Length of the packet, including the SC_RDM and SC_SUB_MESSAGE, excluding checksum
4 - X	Payload	Content of the RDM request packet
X+1 - X+2	Checksum	Checksum of the packet, see E1.20 - 6.2.11 Checksum

The first fragment must contain at least the first 3 bytes of the RDM packet. Subsequent writes are treated as continuations until all expected bytes are received. Once the full packet is received, the packet integrity is verified using the checksum. Packets that fail validation are discarded.

Note: The client is responsible for dispatching the chunks in correct order.

12.1.5 RDM RX characteristic

Name	Description
UUID	bff00005-247d-6e1c-448c-223dfa0bd00c
Read access	yes
Write access	no
Notifications	no
Max write size	variable, up to 247 bytes or negotiated MTU

This characteristic is read to obtain available RDM response packet.

12.1.6 RDM status characteristic

Name	Description
UUID	bff00006-247d-6e1c-448c-223dfa0bd00c
Read access	yes
Write access	no
Notifications	yes
Max write size	1 byte

Read this characteristic to obtain the status of the pending RDM transaction. Notifications on changes is supported.

Response data

Value	Name	Note
0	RDM_NO_OPERATION	No operation
1	RDM_IN_PROGRESS	RDM transaction in progress
2	RDM_RESPONSE	Response to the RDM transaction is available
3	RDM_TIMEOUT	RDM transaction timed out

12.2 Generic RX/TX data service

The Generic RX/TX data service allows for apps to communicate with the host MCU via the SPI interfaces.

Before using this service it is recommended to use the OEM information codes register in the SPI interface. This allows any connecting apps to distinguish different applications from each other when using this service through identifying the correct manufacturer ID and product ID.

Name	Description
UUID	95221000-c733-6ac3-5256-467c1e68623f

12.2.1 RX characteristic

Name	Description
UUID	95221001-c733-6ac3-5256-467c1e68623f
Read access	yes
Write access	no
Data size	variable, up to 128 bytes or negotiated MTU

This characteristic is read to obtain available data.

12.2.2 TX characteristic

Name	Description
UUID	95221002-c733-6ac3-5256-467c1e68623f
Read access	no
Write access	yes
Data size	variable, up to 128 bytes or negotiated MTU

Write data to this characteristic to send it to the SPI interface.

12.2.3 Data available characteristic

Name	Description
UUID	95221003-c733-6ac3-5256-467c1e68623f
Read access	yes
Write access	yes
Notifications	yes
Data size	1 byte

Read this characteristic to obtain the number of bytes available in the RX characteristic. Notifications on changes is supported. Write this characteristic to `0` to confirm data has been read from the RX characteristic.

12.2.4 Clear-to-send characteristic

Name	Description
UUID	95221004-c733-6ac3-5256-467c1e68623f
Read access	yes
Write access	no
Notifications	yes
Data size	1 byte

Read this characteristic to see if data can safely be written to the TX characteristic. This characteristic has a value of `0` after data has been written, until the host MCU has read the data from the SPI interface. After host has read the data, this characteristic has a value of `1` and it's safe to write more data.

Notifications of changes are supported on this characteristic.

13. RDM

13.1 RDM RX

TimoTwo supports RDM in receiver mode in all cases. It can be used either via SPI or via the DMX interface. When used via the DMX interface a responder must implement the E1.20 RDM standard accordingly.

When used via SPI, LumenRadio has an example code for Arduino available on our [GitHub page](https://github.com/LumenRadio/crmx-timotwo-spi-rdm-responder) [https://github.com/LumenRadio/crmx-timotwo-spi-rdm-responder].

13.2 RDM TX

TimoTwo can support RDM transmissions to be used directly in controllers, or stand-alone as RDM transmitter unit. This can be supported in two ways:

- Via DMX interface - RDM proxy suitable for use in stand-alone products.
- Via SPI - suitable for integration in controllers.

When used via SPI, LumenRadio has an example code for Arduino available on our [GitHub page](https://github.com/LumenRadio/crmx-timotwo-spi-rdm-master) [https://github.com/LumenRadio/crmx-timotwo-spi-rdm-master].

Both these functions are optional addon functionality that is unlocked through a software license. Please contact [LumenRadio sales](mailto:sales@lumenradio.com) [mailto:sales@lumenradio.com] for further details on pricing.

14. Output Power

14.1 Receivers

The radio output power of the TimoTwo module in receiver mode cannot be directly controlled. Instead the radio output power is automatically adjusted to match the output power from the transmitter.

14.2 Transmitters

The radio output power of TimoTwo FX in transmitter mode can be controlled using SPI. Please see the corresponding section in the SPI chapter of this document. Output power can also be adjusted using the CRMX Toolbox app.

15. Firmware update

The radio firmware in TimoTwo can be updated. This can be performed via SPI from a host processor in a fixture, or over the air (via Bluetooth) or via the DMX/RDM interface.

For details about updates, please contact [support](mailto:help@lumenradio.com) [mailto:help@lumenradio.com].

15.1 DMX interface

It is also possible to update firmware in TimoTwo via the DMX interface and the [CRXM Upgrade cable](https://lumenradio.com/products/crmx-upgrade-cable/) [https://lumenradio.com/products/crmx-upgrade-cable/]. This requires the DMX interface to be accessible from outside the fixture.

See the [link switch section](#) [#link-switch/] for information on how to set TimoTwo into firmware update mode.

Use the [CRM Update utility](https://lumenradio.com/support/?post_id=1937&category=crmx-oem-modules&protocol=wireless-dmx) [https://lumenradio.com/support/?post_id=1937&category=crmx-oem-modules&protocol=wireless-dmx] to update the firmware.

15.2 SPI

15.2.1 Definitions

- FW_BLOCK_CMD_1 = 0x8E (size = 254 bytes)
- FW_BLOCK_CMD_2 = 0x8F (size = 18 bytes)
- CCI_CHUNK_SIZE = 272
- CCI_HEADER_SIZE = 4

The SPI driver can only handle 255 bytes, giving the CCI chunk has to be send as two parts, FW_BLOCK_CMD_1 and FW_BLOCK_CMD_2.

15.2.2 Procedure

1. Set the SPI bus speed to at most 2 MHz.
2. Wait for the module to reboot - approx 2 seconds.
3. Put the TimoTwo in UPDATE_MODE. Do this by writing 0x40 to the config register (0x00).
4. Wait at least 0.5 seconds for the module to prepare. The status LED will now be blinking slowly.
5. Read the status (0x01) register and make sure bit 7 is high.

6. Open the .cci firmware file and skip the first 4 bytes.
7. Read 254 bytes from the firmware file and write them to the module using the SPI command `FW_BLOCK_CMD_1`.
8. Read the next 18 bytes, write to the module using the SPI command `FW_BLOCK_CMD_2`.
9. The first block of data (254+18) is the header information of the firmware file. After each 16 blocks of data (after the header block), the module will write received firmware data to its memory, so you will have to wait until it's done before sending the next blocks (at least 100ms, recommended 500ms).
10. If not End-of-File: Go to step 6
11. If End-of-File: Wait for the chip to finish update, at least 1 second.
12. Read the `STATUS` register to make sure the module has exit the update mode. This will signify that the module has received all the firmware bytes and has verified it to be valid. The status LED should now be blinking quickly if the radio is on, or not blink at all.
13. Read the `VERSION` register to confirm you have flashed the selected firmware.

16. Flex mode selection

This chapter describes the different methods of selecting between the different flex modes – also known as modes of operation. Flex modes can be controlled using four different methods; SPI, link switch, digital pin input or via CRMX Toolbox app.

16.1 SPI

Bit #1 in the `CONFIG` register is a read/write bit for checking and controlling the flex mode.

Value	Mode
0	Receiver
1	Transmitter

Writing a value to this bit may change the flex mode, depending on the current flex mode. If switching from receiver to transmitter or from transmitter to receiver the TimoTwo module will re-initialize in the selected mode. The host then needs to reconfigure any settings that are desired.

If writing a value to this bit that is the same as the current value, no action is taken. This means that other bits in this register can be written without affecting the flex mode.

16.2 Link switch

In this section the terms "short push" and "long push" are used. Please refer to the table below for details.

Type	Switch closed time
Short push	10 ms – 500 ms
Long push	> 3 s

When the radio module is in either transmitter or receiver mode, follow this procedure to change the flex mode:

1. Five (5) short pushes followed by one (1) long push will enter flex mode selection.
2. Now the status LED will start flashing to indicate the currently selected flex mode (refer to diagrams below for details).
3. Each short push will toggle the currently selected flex mode.
4. To save the selection, perform a long push. This will save the selection and re-initialize the module.

If no selection is made within 15 seconds from the last push, mode selection will be cancelled and normal operation will resume in the previously selected flex mode.



Flashing: off (0V) 50 ms / on (V_{DD}) 50 ms: receiver mode selected

Flashing: off (0V) 500 ms / on (V_{DD}) 500 ms: transmitter mode selected

16.3 Digital input

The digital input selection method is designed to be used in very simple integrations where SPI is not used. This function is overloaded onto the SPI pins.

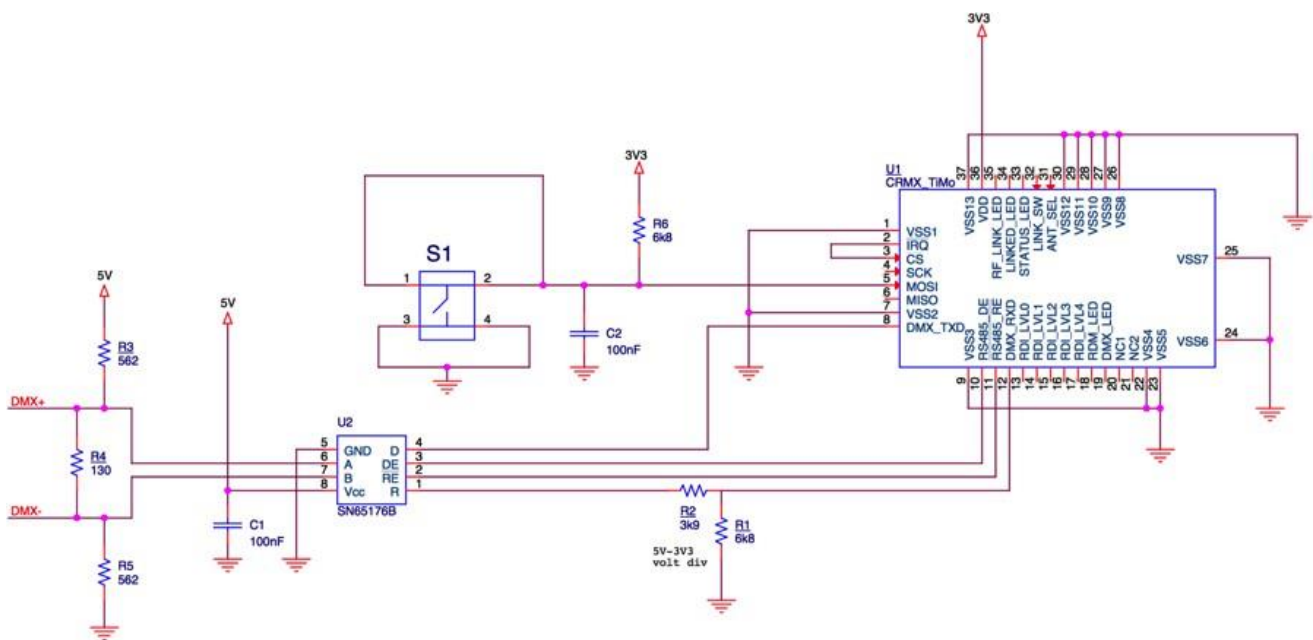
NOTE: SPI operation is **NOT** available when using this method.

To enable this mode, SPI pins IRQ and CS shall be connected together. In this mode, MOSI is used to control the flex mode. This pin is internally pulled to 3.3V, so if MOSI is not connected, the module defaults to receiver mode. To enable transmitter mode, pull MOSI signal to 0V.

MISO and SCK shall not be connected in this configuration.

Please refer to example schematics below for details on how to use digital pin selection method.

MOSI pin input signal	Mode
0 V	Transmitter
3.3 V	Receiver



Example using flex mode selection switch

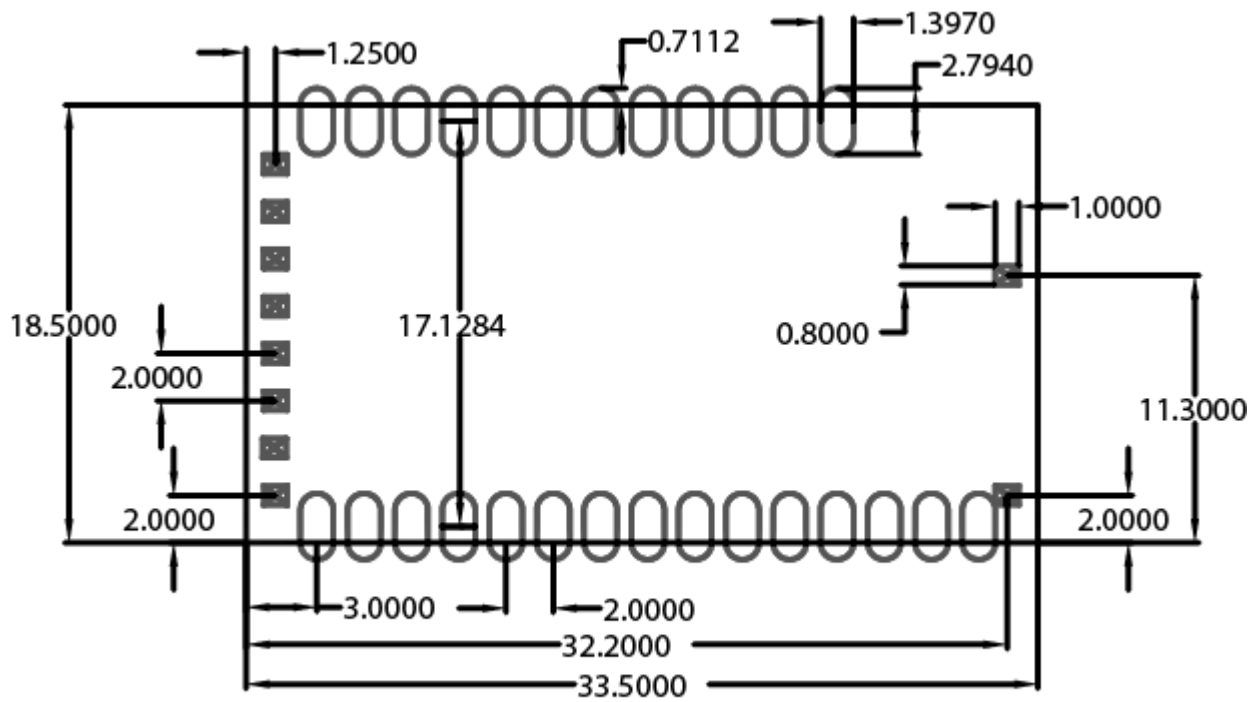
17. Specifications

17.1 Electrical

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{DD}	Supply voltage	3.0	3.3	3.6	V
I_{DD}	Supply current		150	250	mA
T_A	Operating temperature	-20		75	°C
V_{IL}	Input voltage logic low	0		0.9	V
V_{IH}	Input voltage logic high	2.5		3.3	V
I_{LED}	Max current drive on LED pins			5	mA
f_{range}	Operating frequency range	2402		2480	MHz
RX_{sens}	Receiver sensitivity (0.1% BER)		-93		dBm
TX_{pout}	TX output power ¹	5		20	dBm
DMX_{size}	DMX frame size (excluding start code)	0		512	
DMX_{rate}	DMX frame rate	0.8		830	fps

¹From 2dBi antenna

17.2 Mechanical



All Dimensions in mm
"x" in pads are for dimensional purposes only

18. Product marking

Products containing a TimoTwo module shall be marked such that it is easy to identify the presence of LumenRadio's CRMX technology within the product. LumenRadio artwork is available by contacting us at sales@lumenradio.com [mailto:sales@lumenradio.com]

In marketing materials the radio link shall be referred to as either a "wireless link", "LumenRadio wireless DMX", "CRMX wireless DMX", "LumenRadio inside", "Powered by LumenRadio" or similar. Additional body text is acceptable to explain this is a DMX receiver. The wireless link **must not** be referred to as "W-DMX", "WDMX" nor only "Wireless DMX" in any literature related to products using a TimoTwo wireless module or other CRMX technology.

18.1 Product documentation and menu systems

When referring to the TimoTwo module and related behaviors within documentation and menu systems, the system should be referred to as a "wireless link" and/or "CRMX" (or derivations thereof), and must not be referred to as "W-DMX", "WDMX" or similar, nor only "Wireless DMX". "CRMX wireless DMX" is accepted.

Suggested terms and definitions are contained in the table below:

Term	Definition
CRMX Wireless link	The top level term used to describe the CRMX radio system
Linked	The CRMX radio system has been linked with a compatible transmitter.
Unlinked	The CRMX radio system is awaiting linking from a compatible transmitter.

18.2 Logo Syndication

By using CRMX modules in your product, you become one of LumenRadio's valued partners. Our website and catalog carry an array of partner logos and it is expected that your logo will be included alongside these. Marketing information, logos and case studies can be sent to the marketing contacts for inclusion in future marketing efforts.

19. Verification and Testing

19.1 Design Verification

LumenRadio operates a full RF laboratory in Sweden and can offer design verification and testing services – please contact LumenRadio for advice.

19.2 Production Testing

All CRMX modules are factory tested before being shipped. However, it is advised to perform some level of testing as part of your products overall test process. LumenRadio would be happy to advise on production testing – please contact LumenRadio for advice.

20.1 FCC information

20.1.1 FCC Information to User

This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with minimum distance 20cm between the radiator & your body.

20.1.2 FCC Guidelines for Human Exposure

This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with minimum distance of 20 cm between the radiator and your body. This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.

20.1.3 FCC Declaration of Conformity

We Aputure Imaging Industries Co., Ltd. declare TimoTwo FX, comply with Part 15 of FCC Rules. Operation is subject to the following two conditions:

- This device may not cause harmful interference, and
- This device must accept any interference received, including interference that may cause undesired operation.

20.1.4 FCC Radio Frequency Interference Warnings & Instructions

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following methods:

- Reorient or relocate the receiving antenna
- Increase the separation between the equipment and the receiver
- Connect the equipment into an electrical outlet on a circuit different from that which the radio receiver is connected
- Consult the dealer or an experienced radio/TV technician for help.
-

FCC Caution: Any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate this equipment. This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.

Antenna Placement Within the Host Platform

The module is tested for standalone mobile RF exposure use condition.

- (1) The antenna must be installed such that 20 cm is maintained between the antenna and users,
- (2) The transmitter module may not be co-located with any other transmitter or antenna.

In the event that these conditions cannot be met (for example certain laptop configurations or co-location with another transmitter), then the FCC authorization is no longer considered valid and the FCC ID cannot be used on the final product. In these circumstances, the OEM integrator will be responsible for re-evaluating the end product (including the transmitter) and obtaining a separate FCC authorization.

Antenna Type and Gain

The following antennas have been certified for use with this module.

Only antennas of the same type with equal or lower gain may also be used with this module.

Other types of antennas and/or higher gain antennas may require the additional authorization for operation.

Antenna Specification list below:

There are two antennas as below:

Model(P/N)	Type	Max Gain
ANT1: KPS-PR-307-L3-2.4Ghz	PIFA Antenna	3.32 dBi
ANT2: YG-PR-339-L2-2.4Ghz	PIFA Antenna	1.32 dBi

End Product Labeling

When the module is installed in the host device, the FCC ID label must be visible through a window on the final device or it must be visible when an access panel, door or cover is easily re-moved. If not, a second label must be placed on the outside of the final device that contains the following text: “Contains FCC ID: 2AABZ-TIMOTWOFX”. The FCC ID can be used only when all FCC compliance requirements are met.

The end product shall bear the following 15.19 statement: This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions:

- (1) This device may not cause harmful interference, and
- (2) This device must accept any interference received, including interference that may cause undesired operation.

Information on test modes and additional testing requirements

This transmitter is tested in a standalone mobile RF exposure condition and any co-located or simultaneous transmission with other transmitter(s) class II permissive change re-evaluation

20.2 Industry Canada statement

IC Regulatory Compliance

This device contains licence-exempt transmitter(s)/receiver(s) that comply with Innovation, Science and Economic Development Canada's licence-exempt RSS(s). Operation is subject to the following two conditions:

(1) This device may not cause interference; and (2) This device must accept any interference, including interference that may cause undesired operation of the device.

Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes :

(1) l'appareil ne doit pas produire de brouillage;

(2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

This equipment complies with IC radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with minimum distance 20cm between the radiator and your body.

Cet équipement est conforme aux limites d'exposition au rayonnement du CI établies pour un environnement non contrôlé. Cet équipement doit être installé et utilisé à une distance minimale de 20 cm entre le radiateur et votre corps.

Please notice that if the ISED certification number is not visible when the module is installed inside another device, then the outside of the device into which the module is installed must also display a label referring to the enclosed module. This exterior label

can use wording such as the following: “Contains IC: 28850-TIMOTWOFX” any similar wording that expresses the same meaning may be used.

l'appareil hôte doit porter une étiquette donnant le numéro de certification du module d'Industrie Canada, précédé des mots « Contient un module d'émission », du mot 28850-TIMOTWOFX ou d'une formulation similaire exprimant le même sens, comme suit

This equipment complies with ISED radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with a minimum distance of 20cm between the radiator & your body. This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.

Cet équipement est conforme aux limites d'exposition aux rayonnements établies par l'ISED pour un environnement non contrôlé. Ce équipement doit être installé et fonctionné avec une distance minimale de 20 cm entre le radiateur et votre corps. Ce transmetteur ne doit pas être placé côte à côte ou fonctionner en combinaison avec toute autre antenne ou émetteur.

Le resant appareil numerique német pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de la classe B prescrites dans le Règlement sur le brouillage radioélectrique édicté par le ministère des Communications du Canada.

20.3 CE

800-8107 TimoTwo FX comply with the Essential Requirements of RED (Radio Equipment Directive) of the European Union (2014/53/ EU). TimoTwo FX meet the ETSI EN 300 328 V2.2.2 conformance standards for radio performance.

20.4 Other modular approvals

TimoTwo also carries other modular approvals, for instance MIC for Japan and KC for South Korea. Contact LumenRadio for specific questions about certain territories.

20.5 Compliance Marking

20.5.1 FCC & Industry Canada

The CRMX modules are certified for FCC as a single-modular transmitter.

CRMX modules are FCC certified radio module that carries a "Modular" grant CRMX radio modules complies to the "Intentional Radiator" portion (Part 15c) for FCC certification: Part 15.247 Transmitter tests.

An end product, incorporating a CRMX module, does not require additional testing or authorization for the CRMX transmitter (or transceiver, in the case of RDM or Flex products). Host end products can use the FCC ID of the certified module as the FCC ID of the host end product. A label displaying the CRMX module's FCC ID must be affixed and visible on the host end product for approval

FCC IDs are required for host end products with radio transmitters.

20.5.2 Other Compliances

For other local compliance regulations (CE, UL, CSA, SRRC, C-Tick, etc.) you are responsible as the product manufacturer to ensure all required compliance testing is completed. LumenRadio are happy to advise on compliance testing – please contact LumenRadio for details.