CME8000-BUS

C-MAX®
RF Technology Specialist

Short Description

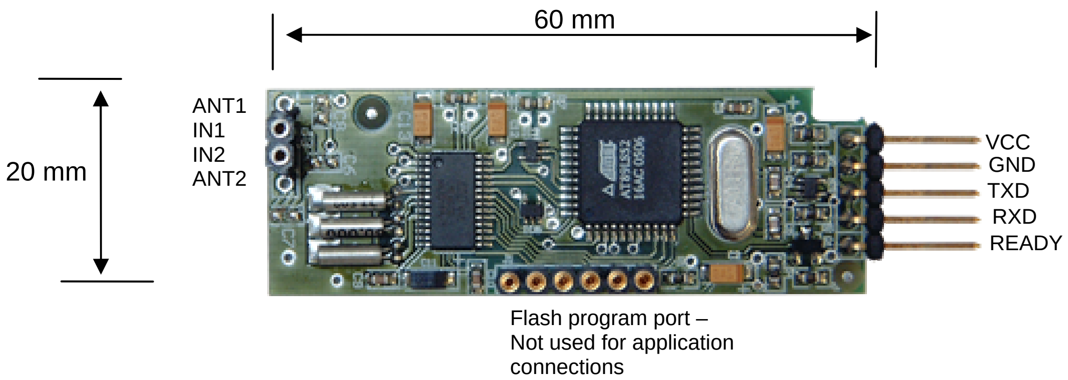
The CME8000 is a BiCMOS integrated straight through receiver with build in very high sensitivity and a pre-decoding of the time signal transmitted from WWVB, DCF77, JJY40, JJY60, MSF and HBG. The receiver is prepared for multi-frequency and country reception by using an integrated logic. The CME8000 is connected to an in-built micro-controller in the CME-BUS module which is programmed with a pre-defined serial interface protocol. The CME-BUS module can then act as a tap into a inter-processor serial communications bus, and is able to communicate between two or more boards, micro controllers or other devices distributed among one or more platforms. From this module, any host can always through its interface obtain absolute proofed time information.

Features

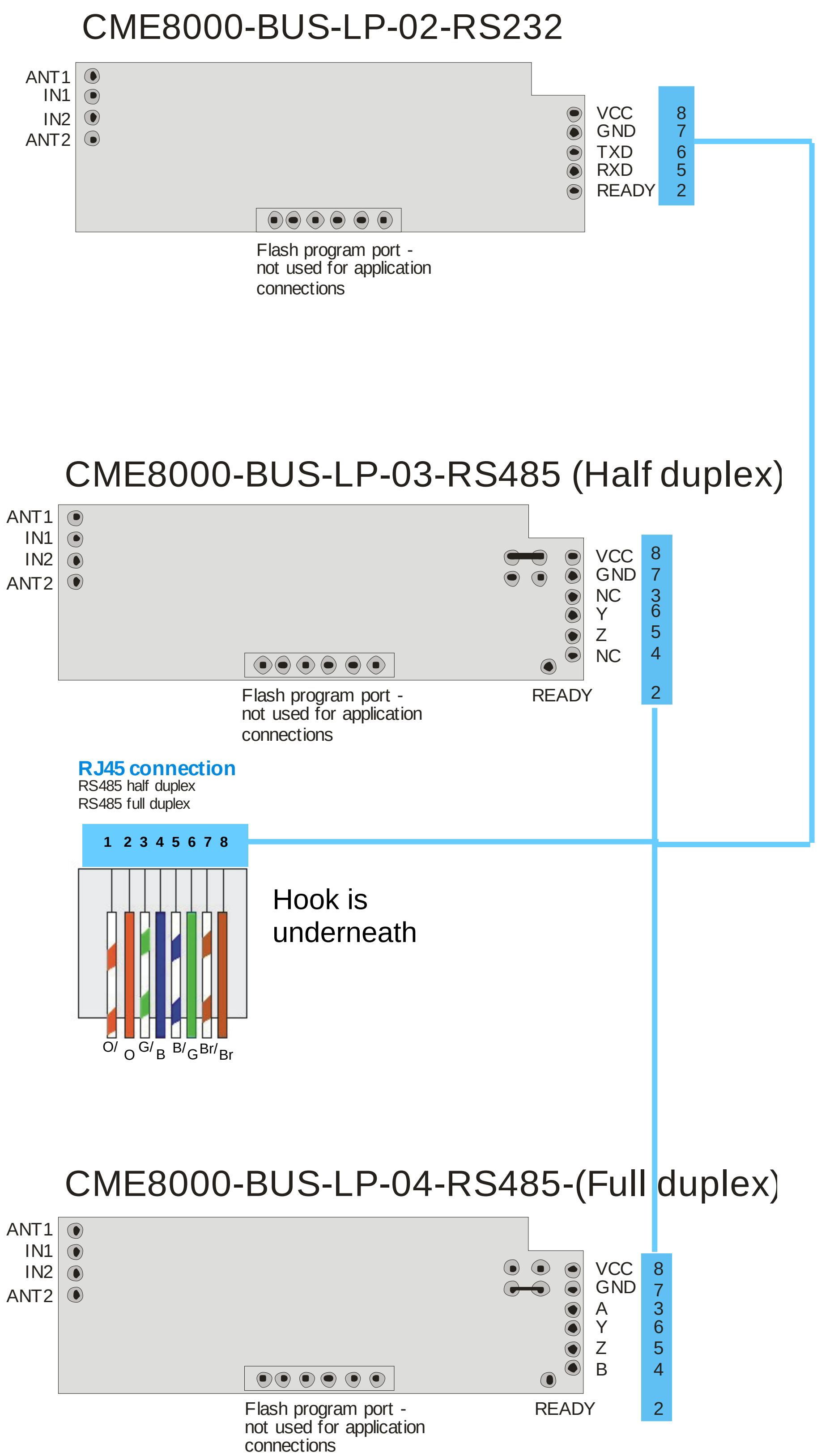
- Automatic reception of long wave time signals world wide
- Manual or automatic selection of radio control signal possible
- Forced reception mode
- Real time clock
- Real signal quality indicator during reception
- 24-hour system
- Host-controllable reception settings (including time and duration of reception)
- Low power consumption (< 2mA during reception active mode)
- Very high long wave reception sensitivity (0.4µV)
- Build in decoding for different signals
- Automatic switch between dual band signals.
- User settable reception mode by host software control command
- Wide operating range: 2.7 V to 5.0 V

Module Layout

CME8000-BUS-LP01 – serial BUS



Country settings are adjusted via software


CME8000-BUS-LP-02-RS232
ANT1
IN1
IN2
ANT2
VCC 8
GND 7
TXD 6
RXD 5
READY 2
Flash program port - not used for application connections
CME8000-BUS-LP-03-RS485 (Half duplex)
ANT1
IN1
IN2
ANT2
VCC 8
GND 7
NC 3
Y 6
Z 5
NC 4
READY 2
Flash program port - not used for application connections
RJ45 connection
RS485 half duplex
RS485 full duplex
1 2 3 4 5 6 7 8
O/ O G/ B B/ G Br/ Br
Hook is underneath
CME8000-BUS-LP-04-RS485-(Full duplex)
ANT1
IN1
IN2
ANT2
VCC 8
GND 7
A 3
Y 6
Z 5
B 4
READY 2
Flash program port - not used for application connections

System Block Diagram

A typical system block is shown in Figure 1.

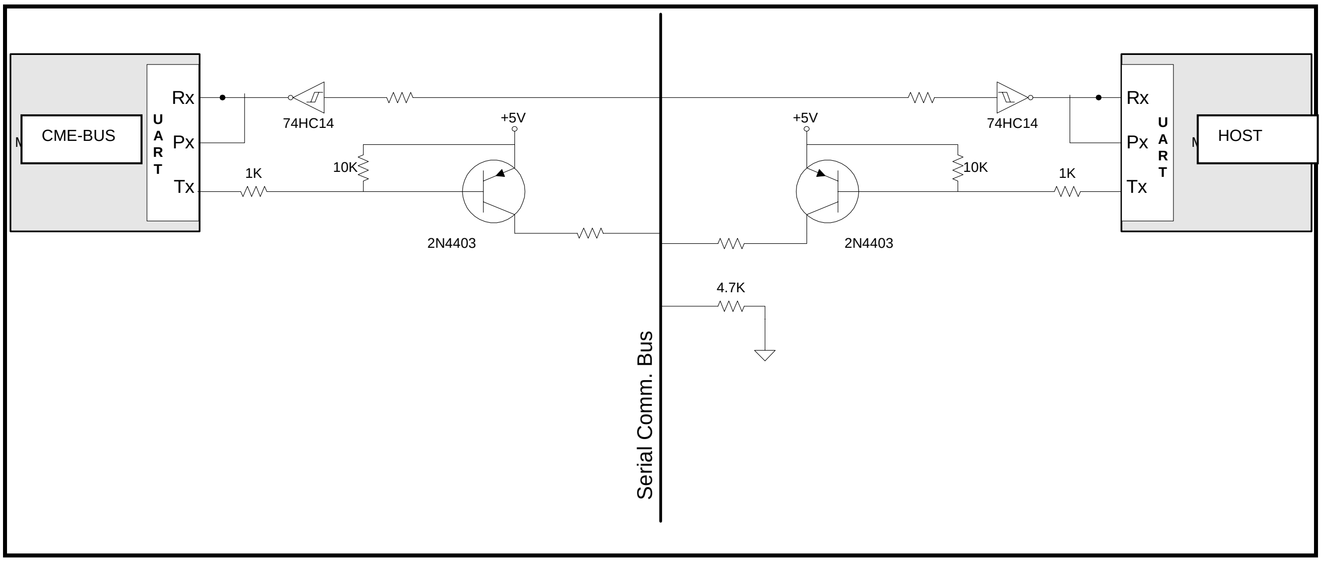


Figure 1. Typical Circuit for Single-line Serial Communication Bus

I/O Specifications

- TTL interface
- Half-duplex
- Asynchronous
- 9600 baud, no parity, 1 start bit, 8 data bits, 1 stop bit

1. OVERVIEW

1.1 Interface with CPU

The CME-BUS uses three wires (READY , RXD, TXD) to communicate with other devices on the serial communication bus. A minimum of two wires can be used if the host is an asynchronous UART and does not make use of the READY interrupt available.

The READY pin is controlled by the bit status of the bits from the command RD_RTC_STATE. Once there is a successful synchronization within the past 27 hours, this pin will be high, or else this pin will be low.

1.2 Time piece functions

The module will respond to commands from the host and return required time information. Time information available includes hours, minutes, second, day of week, month, year(in last two digits), special information (e.g. DST status)

1.3 Time signal reception functions

When receiving time signals, it is possible to designate the module to try search presence of either only one, two, or all signals and synchronise to the signal with better reception. There are two reception activation possibilities: automatic reception activates radio control reception at set regular intervals and times of the day. This can be set in one hour units. The other mode : forced reception mode allows the host to start reception of radio control signal whenever activated.

1.4 The duration of reception, the number of auto receptions, interval between auto reception trials (if more than one is defined), start time of 1st auto reception of the day can all be defined by the host.

1.5 Radio Control Reception Mode

Radio control reception can be set by software command from an external host. Various reception modes can be defined :

Single band :	Dual band:
DCF	JJY40 / JJY60
JJY40	MSF / DCF
JJY60	
MSF	
WWVB	

1.6 Default settings

There will be also a set of default values for all above upon power on:

Radio control reception mode	WWVB
Time	00:00:00 (UTC time)
Date	1st January 2005
DST	OFF
Maximum duration of reception	15 minutes
Number of receptions	2
Time interval between receptions	2 hours
Start time of 1st reception	3:00 am
Reception activation mode	Auto activated
Reception comparison	All compare

2. Table of Functions and input/output parameters

Command	R/W	Command	Data Included in command								Data returned to sender
			Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
RTC_SET_SEC	W	0x60	N/A	S40	S20	S10	S8	S4	S2	S1	Bit 0 : SER
RTC_SET_MIN	W	0x61	N/A	Min40	Min20	Min10	Min8	Min4	Min2	Min1	Bit 0 : MiER
RTC_SET_HR	W	0x62	DST	N/A	H20	H10	H8	H4	H2	H1	Bit 0 : HrER
RTC_SET_DATE	W	0x63	N/A	N/A	D20	D10	D8	D4	D2	D1	Bit 0 : DER
RTC_SET_MTH	W	0x64	N/A	N/A	N/A	Mo10	Mo8	Mo4	Mo2	Mo1	Bit 0: MER
RTC_SET_YR	W	0x65	N/A	Y40	Y20	Y10	Y8	Y4	Y2	Y1	Bit 0 : YER
RD_RTC_STATE	R	0x66	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Bit 0 : Once Bit 1: Curr-sync Bit 2: 27-Sync
RD_TIME	R	0x70 – 0x78	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	[0]=Board Address [1]=Second information [2]=minute information [3]=hour (in 24 hr) & DST information [4]=date information [5]=month information [6]=year information [7]=day of week information
SET_RX_CTRL	W	0x80	C1	C0	CANCEL	FCE	AUTO	000=RC OIFF 001=JJY60 010=WWVB 011=MSF 100=DCF 101=JJY40/JJY60 110=MSF/DCF 111=JJY40			N/A
SET_RX_TIME	W	0x81	T2	T1	A20	A10	A8	A4	A2	A1	N/A
SET_RX_DUR	W	0x82	N/A	N/A	Int20	Int10	Int8	Int4	Int2	Int1	N/A
SET_NO_CMP	W	0x83	N/A	SB	DOW	Yr	Mth	Date	Hr	Min	N/A
RD_RX_STATE	R	0x84	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Bits 0 -2 : 000=RCC OFF 001=JJY60 010=WWVB 011=MSF 100=DCF 101=JJY40/JJY60 110=MSF/DCF 111=JJY40 Bit 3: BUSY Bit 4: OFF Bit 5-6: BS0 – BS1

3. Description of Functions

3.1 RTC Commands

3.1.1 RTC_SET_SEC (Default : 00h)

This command sets the second data in the RTC. SER is a bit (as 1) which is returned to the master if the set second is not within 0 to 59. In this case, the previous second data will be restored.
S1 to S40 represents BCD values of the second. Range is 00h to 59h

3.1.2 RTC_SET_MIN (Default : 00h)

This command sets the minute data in the RTC. MiER is a bit (as 1) which is returned to the master if the set minute is not within 0 to 59. In this case, the previous minute data will be restored.
Min1 to Min40 represents BCD values of the minute. Range is 00h to 59h

3.1.3 RTC_SET_HR (Default: 00h)

This command sets the hour date in 24 h system in the RTC. HrER is a bit (as 1) which is returned to the master if the set minute is not within 0 to 23. In this case, the previous hour data will be restored.
H1 to H20 represents BCD values of the hour. Range is 00h to 23h
The bit 7 represents the DST status to be manually set by the user with the following definition:

Let Signal DST (or DST from date+month calculation) = A
Let RTC_SET_HR DST bit = B
Let RD_Time DST bit = C
Then the following relationship is applied.

A	B	C
1	1	0
0	1	1
1	0	1
0	0	0

That is, if RTC_SET_HR DST bit is 0, RD_Time DST bit shows the signal DST (or calculation DST) status.
If RTC_SET_HR DST bit is 1, RD_Time DST bit shows the overidden DST status which is manually inverted by the user.

3.1.4 RTC_SET_DATE (Default: 01h)

This command sets the date of the month in the RTC.
D1 to D20 represent BCD values of the date. Range is 01h to 31h.
DER is a bit (as 1) which is returned to the master if the set date does not comply with the month and year inside the RTC. In this case, the date is automatically reset to the last date of the set month and year.

3.1.5 RTC_SET_MTH (Default: 01h)

This command sets the month data in the RTC.
Mo1 to Mo10 represent BCD value of the month. Range is 01h to 12h.
MER is a bit (as 1) which is returned to the master if the set month does not comply with the date inside the RTC. In this case, the date is automatically set to the last date of the set month and year.

3.1.6 RTC_SET_YR (Default: 00h)

This command sets the last two digits of the year data in the RTC
Y1 to Y80 represent BCD value of the year. Range is 00h to 99h.
If the set year does not comply with the date and month inside the RTC, the date is automatically set to the last date of the set month and year.

3.1.7 RD_RTC_STATE

This command reads out the update status of the RTC

ONCE: a "1" indicates RTC data has been updated at least once

CURR-SYNC: a "1" indicates RTC data has just been correctly synchronized with RC time, once this bit has been read, this bit will reset to "0".

27-SYNC: a "1" indicates RTC data has been synchronized with RC time within 27 hours.

3.1.8 RD_TIME

This command returns to the requestor unit the full time information in the RTC as follows:

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Data 0	RC Board address							
Data 1	X	S40	S20	S10	S8	S4	S2	S1
Data 2	X	Min40	Min20	Min10	Min8	Min4	Min2	Min1
Data 3	DST	x	H20	H10	H8	H4	H2	H1
Data 4	X	x	D20	D10	D8	D4	D2	D1
Data 5	X	x	x	Mo10	Mo8	Mo4	Mo2	Mo1
Data 6	Y80	Y40	Y20	Y10	Y8	Y4	Y2	Y1
Data 7	X	x	x	x	x	W4	W2	W1

3.2 Radio control reception commands

3.2.1 SET_RX_CTRL

This command sets the receiver module to a predefined state of reception during auto and manual reception: either single band, dual band or triple band reception.
The mode of reception is defined by the bits 0 to 2.

AUTO : If this bit is set as 1, the module will go into reception automatically according to auto reception time set in SET_RX_DUR for duration CMD_RX_INT, and mode as set in bits 0 to 2. If bit is 0, there will be no auto radio control reception as if radio control function is off.

FCE : If this bit is set as 1, the module will go into reception immediately according to mode as set in bits 0 to 2 for maximum duration as set in SET_RX_DUR.

CANCEL: If this bits is set as 1, the module will stop reception immediately if reception is in process.

C0,C1 : sets the number of auto receptions per day .
- 00 – there will be no auto receptions
- 01 – there will be 1 auto reception per day
- 10 – there will be 2 auto receptions per day
- 11 – there will be 3 auto receptions per day

3.2.2 SET_RX_TIME

This command sets the auto reception time of the day in hours.

Bits 0 to 5 : Sets in binary form the hour of the day in 24 hours system when the auto reception starts.

T1, T2: Sets the interval between the number of auto receptions per day as set in SET_RX_CTRL. For example, if T1, T2 = 10, and C0,C1 = 11, then there will be 3 auto receptions per day in intervals of 2 hours apart.

3.2.3 SET_RX_DUR

This command sets the maximum time of duration for each reception trial in minutes. The maximum duration that can be set is 30 minutes, the minimum duration that must be set is 3 minutes. This applies for dual and multi-frequency reception as well, for example, dual JJY40 and JJY60 together has maximum reception duration of 30 minutes, the minimum duration that must be set is 3 minutes.

3.2.4 SET_NO_CMP

Upon reception of a time stream, comparisons with previous data, both from 24hours before or from the minute before if any will be performed. If the recent minute's data coincides with 2 of the previous data, the reception is considered as successful and the time data will be updated to the RTC. This comparison algorithm usually compares all data available including hour, minute, date, month, day of the week, year and special bits (that is daylight saving bits) information. To speed up reception in weak signal, in some exceptional case like for a product which only shows the hour and minute, other time data can be neglected.

This command sets the time information **<u>not to be compared</u>** from the received time stream. Upon setting of the specified bit, for example setting bit 4 means the year data will not be analysed and compared in the comparison algorithm of reception.

3.2.5 RD_RX_STATUS

This command allows the requestor to know the reception mode that the unit is working under, for example, whether it is under dual JJY or single band DCF etc. It also tells the requestor whether :

i) the unit is BUSY (set to 1) i.e. under reception, then at the same time also showing the BSI value (refreshed every second), or

ii) the unit is not under reception (OFF set to 1), or

4. Communications and Collision Detection Protocol

The communications subsystem will use a collision detection scheme to determine when the communications port is free or in use and when a collision has occurred. A collision occurs when two or more masters attempt to use the communications bus at the same time. This can be detected because the transmit and receive ports on the micro controller are connected to the same bus wire, albeit through additional circuitry. Careful control of the interrupts associated with the communications port will allow this to be an interrupt driven activity. Logically, this is a byte-oriented protocol. A higher level protocol determines the length and content of packets comprising messages.

The following state table describes the function of the collision detection protocol.

Table 1. Physical Communication Protocol State Table

State	Action	Result	Next State
1	Are there bytes to send?	Yes	2
		No	1
2	Check bus activity	Busy	6
		Free	3
3	Send Byte		4
4	Does byte sent = byte received?	Yes	1
		No	5
5	Delay 3 to 8 byte times	Complete	1
6	Delay 5 byte times	Complete	1

In the two delay states, 5 and 6, a byte time is defined as the amount of time required to transmit a single byte on the communications bus. This time is determined by the baud rate, number of data bits, number of stop bits, and a parity bit, if used.

The variable delay period shown in state five is intended to make the restart delay time random. If a collision does occur, the two bus masters will not delay the same amount of time before retrying transmission. This will reduce the possibility of subsequent collisions by the two masters. The variable delay period could be determined by a pseudo random number process, or by board function.

5. Data-Link Layer

The data-link layer of the software defines the information that moves across the bus in any given packet. The bytes defined in the data-link layer do not necessarily have a one-to-one correlation with the bytes in the physical layer. Many physical devices, such as I^2C, have bits in the physical layer that implement the functions of some of the bytes in the data-link layer. This data-link layer is intended to be generic so that it and the application layer of the software will not need to change even if the physical device must be redesigned. This layer of the communication system is appropriate for such technologies as a UART multi-drop environment.

The protocol is designed for use in a master/slave environment. There is no constraint that says a system must have one master and several slaves. Rather, this protocol is intended to be implemented in a small network type of environment where one device can be a master through one communication cycle and then become a slave for another communication cycle.

5.1 Data Word

The data will be transmitted in a 8-N-1 format as show in Figure 2. The transmitting device will send data with 1 start bit, followed by 8 data bits, ending with 1 stop bit. The sequencing of the data bits will be from LSB (bit 0) to MSB (bit 7).

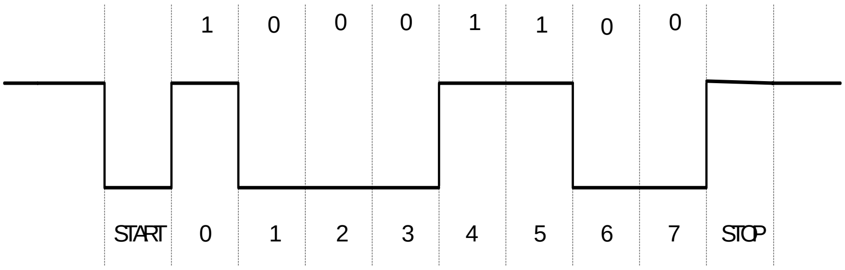


Figure 2. Serial Data Word Example.

5.2 Command Packet

A command is used by a master device to request action from a slave. The command packet has the structure shown in Table 2.

Table 2. Structure of Command Packet.

STX	*Address*	*Packet Length*	*Comman d*	*Data*	*CRC*	*ETX*
1 byte	1 byte	1 byte	1 byte	n bytes	2 bytes	1 byte

Where:

- STX is one byte with a value of 0x02. To determine if you have a valid STX, check whether an ACK byte follows. If so, the 0x02 is in the middle of a transmission and should not be interpreted as an STX.
- Address is one byte. The radio control module will have an address defined as **0x59**.
- The packet length is the number of bytes in the packet including STX, address, packet length, command, data, CRC, and ETX. The packet length value is equivalent to 7 + n, where n is the number of data bytes.
- The command or request is one byte defined by the application layer.
- The data may be zero, one, or multiple bytes as defined in the application layer, except for the case of a request. In this case, the first data byte will be the master's address so the slave will know which device to respond to.
- The CRC is a 16-bit CRC.[1] The CRC is computed on all bytes of a packet except the STX, the CRC byte pair and the ETX. See the Appendix for the CRC source code.
- ETX is one byte with a value of 0x03.

5.3 ACK and NAK

The protocol is shown in Tables 3,4, and 5. It is demonstrated with all ACKs in the sequence. Note that an ACK is typically sent in response to a transmitted byte but is also sent in response to a complete packet transmission and represents a validation of the packet's length and CRC. At any point in the process where an ACK can be sent a NAK may be sent instead. If a NAK is transmitted, the communication sequence is aborted at that point. The master then has the option of re-starting the sequence (application dependent). A NAK shall be transmitted only in response to an overrun or framing error detected on (or in lieu of) a received byte or in response to the reception of a complete packet when the computed packet CRC does not match the transmitted packet CRC.

An ACK is one byte with a value of 0x06.

A NAK is one byte with a value of 0x15.

5.4 Protocol

The following protocol tables assume that time increments with each row of the table. Up to 200 milliseconds of delay should be tolerated for any expected event (row); an ACK response to a transmitted byte or the reception of the next byte of an incomplete packet. The arrows indicate the data direction. Table 3 is for a command sequence, Table 4 is for a request sequence, and Table 5 is for a response to request sequence. For the request sequence and response to request sequence, the first data byte is the transmitter's address.

Table 3. Command Sequence.

Command Sequence			
Master		*Slave*	
STX ☞	[0x02]		
Slave Address ☞	1 byte		
		[0x06]	☜ ACK
Packet Length ☞	1 byte		
		[0x06]	☜ ACK
Command ☞	1 byte		
		[0x06]	☜ ACK
Data Byte 1 ☞	1 byte		
		[0x06]	☜ ACK
Data Byte 2 ☞	1 byte		
		[0x06]	☜ ACK
Data Byte n ☞	1 byte		
		[0x06]	☜ ACK
CRC MSB ☞	1 byte		
		[0x06]	☜ ACK
CRC LSB ☞	1 byte		
		[0x06]	☜ ACK
ETX ☞	[0x03]		
		[0x06]	☜ ACK
		[0x06]	☜ ACK

Table 4. Request Sequence.

Request Sequence			
Requestor		*Requestee*	
STX ☞	[0x02]		
Requestee Address ☞	1 byte		
		[0x06]	☜ ACK
Packet Length ☞	1 byte		
		[0x06]	☜ ACK
Request Command ☞	1 byte		
		[0x06]	☜ ACK
Data Byte 1 – Requestor's Address (Transmitter) ☞	1 byte		
		[0x06]	☜ ACK
Data Byte 2 ☞	1 byte		
		[0x06]	☜ ACK
Data Byte n ☞	1 byte		
		[0x06]	☜ ACK
CRC MSB ☞	1 byte		
		[0x06]	☜ ACK
CRC LSB ☞	1 byte		
		[0x06]	☜ ACK
ETX ☞	[0x03]		
		[0x06]	☜ ACK
		[0x06]	☜ ACK

Table 5. Response to Request Sequence.

Response to Request Sequence			
Requestee		Requestor	
STX ☞	[0x02]		
Requestor's Address ☞	1 byte		
		[0x06]	☜ ACK
Packet Length ☞	1 byte		
		[0x06]	☜ ACK
Command to which Requestee is Responding ☞	1 byte		
		[0x06]	☜ ACK
Data Byte 1 – Requestee's Address (Transmitter) ☞	1 byte		
		[0x06]	☜ ACK
Data Byte 2 ☞	1 byte		
		[0x06]	☜ ACK
Data Byte n ☞	1 byte		
		[0x06]	☜ ACK
CRC MSB ☞	1 byte		
		[0x06]	☜ ACK
CRC LSB ☞	1 byte		
		[0x06]	☜ ACK
ETX ☞	[0x03]		
		[0x06]	☜ ACK
		[0x06]	☜ ACK

6. Application Layer

All products must incorporate a version number request and a version number reply into the application layer. This will enable the factory and service equipment to verify the version number and product type of each device for every product.

Table 6. Version Number Request Command.

Request Command	Value
Version Number Request	0x01

The Version Number Request will have one data byte, which is the requestor's address. This enables the receiver to respond to the correct device.

The Version Number Reply will have four data bytes. The first data byte will be the requestee's address. The requestor will then know which device is replying. Data byte 2 will be the product identifier (specified in the product application layer). The next two data bytes will be the encoded version number. Table 7 and 8 are an example of a version number request and version number reply sequence. For this example, please assume the following:

Requestor's Address	0x15
Requestee's Address	0x13
Product Identifier	5-burner model = 0x05
Version Number	5.8 = 0x05 0x08

Table 7. Example of Version Number Request Sequence.

Request Sequence			
Requestor		Requestee	
STX ☞	[0x02]		
Requestee Address ☞	[0x13]		
		[0x06]	☜ ACK
Packet Length ☞	[0x08]		
		[0x06]	☜ ACK
Request Command ☞	[0x01]		
		[0x06]	☜ ACK
Data Byte 1 – Requestor's Address (Transmitter) ☞	[0x15]		
		[0x06]	☜ ACK
CRC MSB ☞	[1 byte]		
		[0x06]	☜ ACK
CRC LSB ☞	[1 byte]		
		[0x06]	☜ ACK
ETX ☞	[0x03]		
		[0x06]	☜ ACK
		[0x06]	☜ ACK

Table 8. Example of Version Number Reply Sequence.

Response to Request Sequence			
Requestee		Requestor	
STX ☞	[0x02]		
Requestor's Address ☞	[0x15]		
		[0x06]	☜ ACK
Packet Length ☞	[0x0]		
		[0x06]	☜ ACK
Command to which Requestee is Responding ☞	[0x01]		
		[0x06]	☜ ACK
Data Byte 1 – Requestee's Address (Transmitter) ☞	[0x13]		
		[0x06]	☜ ACK
Data 2 – Product Identifier ☞	[0x05]		
		[0x06]	☜ ACK
Data 3 – Version Number 1 ☞	[0x05]		
		[0x06]	☜ ACK
Data 4 – Version Number 2 ☞	[0x08]		
		[0x06]	☜ ACK
CRC MSB ☞	[1 byte]		
		[0x06]	☜ ACK
CRC LSB ☞	[1 byte]		
		[0x06]	☜ ACK
ETX ☞	[0x03]		
		[0x06]	☜ ACK
		[0x06]	☜ ACK

Ordering information:

Interface	Communication	PCB with antenna	casing with edge connector	sealed casing (IP67) with 4m cat5 wire
serial bus	half duplex	CME8000-BUS-LP-01	CME8000-BUS-GS-01	CME8000-BUS-GV-01
RS232	full duplex	CME8000-BUS-LP-02	CME8000-BUS-GS-02	CME8000-BUS-GV-02
RS485	half duplex	CME8000-BUS-LP-03	--	CME8000-BUS-GV-03
RS485	full duplex	CME8000-BUS-LP-04	--	CME8000-BUS-GV-04

Dimensions:

Ordering no.	Description	Dimension
CME8000-BUS-LP-XX	CME8000-BUS module – PCB antenna included	20 x 60mm (without antenna)
CME8000-BUS-GS-XX	CME8000-BUS module with IP65 casing antenna included	40 x 70 x 15mm
CME8000-BUS-GV-XX	CME8000-BUS module with 4m Cat5 wire, with casing, sealed (IP67) antenna included	40 x 70 x 15mm

CME8000-BUS-LP

CME8000-BUS-GS

CME8000-BUS-GV