LifeGuard Data Transmission Protocol (CPOD-Base Station)

General Considerations:

CPOD can send ACK and REQ commands (It will ACK most of the time)
BS only sends REQ (the BS never ACK a request, it echoes that REQ back instead)

In general, when the BS sends a request (REQ with a SEQ#), the CPOD responds with the requested data, acknowledging that request with the corresponding ACK and repeating the request SEQ#.

On rare occasions the CPOD can send a request signaling that he needs the BS to echo that request right away, since the CPOD probably has new configuration settings that need to be requested by the BS.

Sequence numbering will be orchestrated by BS. CPOD is expected to ACK messages by repeating the provided SEQ in order to confirm that it has processed the request.

Limitations of CPOD: CPOD's command stack is 8 commands deep. CPOD's data buffer for receiving data from the BS is 40 bytes long. In addition, CPOD can only store data from ONE request. When it receives another request that contains data (SAMPLING_PARAMETERS or SET_TIME), it will overwrite the previous data in its data buffer. BS must ensure that it never sends another REQ that contains data without having received an acknowledgment for a previous request that contained data. Since SAMPLING_PARAMETERS and SET_TIME are the only requests that contain data, this case is important only during setup.

Note: All Opcodes are coded according to Appendix A.

Commands are coded on 4 bits (16 values). The complete command byte in the message frame is organized like this:

Upper 4 bits	Lower 4 bits
Request Commands (REQ)	Acknowledgement Commands (ACK)

Command codes (4bits for both ACK and REQ) are:

Code	Abbrev. Command
0x0	NO_OPERATION
0x1	START_DOWNLOAD
0x2	START_STREAMING
0x3	END_SESSION
0x4	AVAILABLE_OPCODES

0x5	SAMPLING_PARAMETERS
0x6	NEXT_PACKET_DOWNLOAD
0x7	NEXT_PACKET_STREAMING
0x8	NEXT_PACKET_LOGGING
0x9	SET_TIME
0xA	RESET
0xB	STATUS
0xC	HANDSHAKE
0xD	SIM
0xE	Not used
0xF	READ_TIMER

[Each field of 4 bits contains a valid command (START_DOWNLOAD, END_SEESION, etc…). The command is an acknowledgement (ACK) or a request (REQ) depending on which field it is in.]

Meaning of Commands:

Command Received	By	Cmd	Meaning
NO_OPERATION			✓ Used when either REQ or ACK field should be empty. For instance, when we only wish to acknowledge a command without further request.
START_ DOWNLOAD	BS	ACK 0x01	✓ CPOD is ready to start sending the data in download mode. ✓ DATA field contains information on the last memory storage location. ✓ CPOD expects a NEXT_PACKET_DOWNLOAD request to follow.
		REQ 0x10	✓ CPOD asks the BS to initiate a download. ✓ DATA filed is empty. ✓ CPOD expects a START_ DOWNLOAD request to follow.
	CPOD	REQ 0x10	✓ CPOD must switch to download mode. ✓ DATA field should be empty. SEQ# should be 0. ✓ CPOD must send a START_ DOWNLOAD ACK.
START_STREAMING	BS	ACK 0x02	✓ CPOD is ready to start sending the data in streaming mode. ✓ DATA field is empty. ✓ CPOD expects an AVAILABLE_OPCODES request to follow.
		REQ 0x20	✓ CPOD asks the BS to initiate a streaming. ✓ DATA field is empty. ✓ CPOD expects a START_ STREAMING request to follow.
	CPOD	REQ 0x20	✓ CPOD stops sampling and expects AVAILABLE_OPCODES request to follow. ✓ DATA field is empty. SEQ# will be 0. ✓ CPOD must send a START_ STREAMING ACK.

END_SESSION	BS	ACK 0x03	✓ CPOD has ended its transmission. ✓ DATA field is empty. ✓ No further specific command is expected.
		REQ 0x30	✓ CPOD must end its transmission. ✓ DATA field could contain values depending on the accompanying ACK. ✓ CPOD expects a END_SESSION REQ.
	CPOD	REQ 0x30	✓ CPOD terminates the session. ✓ DATA field is empty. ✓ CPOD must send an END_SESSION ACK.
AVAILABLE_OPCODES	BS	ACK 0x04	✓ CPOD sends the new list of available opcodes. ✓ DATA contains the list of available opcodes. ✓ CPOD logs this ACK in flash memory ✓ CPOD expects a SAMPLING_PARAMETERS request to follow.
		REQ 0x40	✓ CPOD has refreshed its list of available opcodes (connect/disconnect) and signals the BS to do so. ✓ DATA field is empty. ✓ BS should send a AVAILABLE_OPCODES request to inquire about new changes
	CPOD	REQ 0x40	✓ BS asks for list of available opcodes ✓ DATA field is empty. ✓ CPOD must compile list of current available opcodes, and send it in a AVAILABLE_OPCODES ACK ✓ CPOD expects a SAMPLING_PARAMETERS request to follow.

SAMPLING_PARAMETERS	BS	ACK 0x05	✓ CPOD sends the new sampling rates for each opcode that the BS required in the previous corresponding REQ message ✓ DATA contains the number of messages per second, and sampling periods, samples per message and offsets for each opcode ✓ CPOD logs this ACK in flash memory ✓ No further specific command is expected.
		REQ 0x50	✓ CPOD has changed some sampling rates and wants to notify BS. ✓ DATA field could contain values depending on the accompanying ACK ✓ BS should send a SAMPLING_PARAMETERS request to inquire about new changes
	CPOD	REQ 0x50	✓ BS sends the requested message sampling rate, and a list of sampling parameters for each opcode ✓ DATA contains the number of messages per second, and sampling periods, samples per message, and offsets for each opcode ✓ ✓ CPOD will resume logging with new sampling rates
NEXT_PACKET_DOWNLOAD	BS	ACK 0x06	✓ CPOD sends the next packet of logged samples ✓ DATA contains the next packet of logged samples ✓ No further specific command is expected.
		REQ 0x60	✓ SHOULD NOT BE USED

	CPOD	REQ 0x60	✓ CPOD is required to send the next packet of logged samples ✓ DATA should be empty ✓ CPOD will compile the data, and send it to BS in a NEXT_PACKET_DOWNLOAD ACK
NEXT_PACKET_ STREAMING	BS	ACK 0x07	✓ CPOD sends the latest packet of sampled data, acquired in streaming mode (logging + streaming) ✓ DATA contains the latest packet of sampled data ✓ No further specific command is expected.
		REQ 0x70	✓ SHOULD NOT BE USED
	CPOD	REQ 0x70	✓ CPOD is required to send the latest packet of sampled data ✓ DATA field is empty. ✓ CPOD will compile the data, and send it to BS in a NEXT_PACKET_ STREAMING ACK
NEXT_PACKET_ LOGGING	BS	ACK 0x08	✓ CPOD sends the latest packet of sampled data, acquired in logging-only mode (no streaming) ✓ DATA contains the latest packet of logged data ✓ No further specific command is expected – This command is expected in Download mode ONLY
		REQ 0x80	✓ SHOULD NOT BE USED, since we are not in streaming mode.
	CPOD	REQ 0x80	✓ SHOULD NOT BE USED, since we are not in streaming mode.
SET_TIME	BS	ACK 0x09	✓ CPOD sends the latest time ✓ DATA contains the latest values that were read from the RTC ✓ No further specific command is expected

		REQ 0x90	✓ SHOULD NOT BE USED
	CPOD	REQ 0x90	✓ CPOD is required to set the time of its RTC ✓ DATA contains the time and date values ✓ CPOD will send the new time and date values to BS in a SET_TIME ACK (Note: "seconds" sent back may be different from received value)
RESET	BS	ACK 0x0A	✓ CPOD has reset itself, flash memory pointers have been reset to beginning of memory ✓ DATA field is empty. ✓ No further specific command is expected.
		REQ 0xA0	✓ SHOULD NOT BE USED
	CPOD	REQ 0xA0	✓ CPOD is required to reset itself and enable overwriting of old data ✓ DATA field should be empty
STATUS	BS	ACK 0x0B	✓ CPOD is required to compile a list of register values for debugging and testing purposes ✓ DATA contains a list of register values ✓ No further specific command is expected.
		REQ 0xB0	✓ SHOULD NOT BE USED
	CPOD	REQ 0xB0	✓ CPOD is required to return a list of register values for debugging and testing purposes ✓ DATA field should be empty
HANDSHAKE	BS	ACK 0x0C	✓ CPOD sends handshake data ✓ DATA contains the device ID, connection type, and firmware version number ✓ No further specific command is expected.

		REQ 0xC0	✓ SHOULD NOT BE USED
	CPOD	REQ 0xC0	✓ CPOD is required to identify itself to the base station and send device ID, connection type, and firmware version number ✓ DATA field should be empty
SIM	BS	ACK 0x0D	✓ CPOD echoes the value of the simulation register ✓ DATA contains the simulation register byte ✓ No further specific command is expected.
		REQ 0xD0	✓ SHOULD NOT BE USED
	CPOD	REQ 0xD0	✓ CPOD is required enable real-data or simulated-data mode based on the value of the simulation register sent to it ✓ DATA contains the simulation register byte
READ_TIMER	BS	ACK 0x0F	✓ CPOD echoes the value of the simulation register ✓ DATA contains the simulation register byte ✓ No further specific command is expected.
		REQ 0xF0	✓ SHOULD NOT BE USED
	CPOD	REQ 0xF0	✓ CPOD is required enable real-data or simulated-data mode based on the value of the simulation register sent to it ✓ DATA contains the simulation register byte

Note: CPOD never receives ACK from the BS.

Message Format:

(SYNC)	0xFF	SIZE	CMD	DATA	SEQ	CRC

SYNC: *1 Byte,* 0x00, Synchronization byte for Base Station RX/TX control (only included in messages sent from BS to CPOD, CPOD will ignore this byte when received)

0xFF : *1 Byte* frame marker

SIZE: *1 Byte*, specifying size of (CMD + DATA + SEQ), the value 0xFF is reserved for EOF indication

CMD: *1 Byte*

REQ (4 bits)	ACK (4 bits)

DATA: *as many Bytes as necessary* (<= 252 bytes)

SEQ: *1 Byte* that contains the sequence number of the request (REQ), and will be repeated in the corresponding acknowledgement (ACK)

CRC: *2 Byte* CCITT-16 CRC of (CMD + DATA + SEQ) High byte first?

Data Format:

START_DOWNLOAD_ACK: [4 bytes]

CSA	PAGEH	PAGEL

CSA : DF_CS_ADDR - Data Flash Chip Select Address *(1byte)*
PAGEH : DF_PAGE_H - Data Flash Page Address High Byte *(1byte)*
PAGEL : DF_PAGE_L - Data Flash Page Address Low Byte *(1byte)*
MPP : MPP – Messages stored per Data Flash Page *(1byte),* **used in CPOD firmware versions 2.0 and higher**

AVAILABLE_OPCODES ACK: [N bytes]

OP_1	OP_2	……..	OP_N

OP_1, OP_2,…, OP_N: OpCodes (1 byte each)

Example:
BS-REQ: FF 02 40 01 00 E2 (SEQ=1)
CPOD-ACK: FF 0B 04 22 2B 08 31 32 33 06 01 03 01 2D 95

SAMPLING_PARAMETERS REQ, SAMPLING_PARAMETERS ACK: [1 + N * 3 bytes]

MPS	SP_1	NBS_1	OFFP_1	……..

MPS : Number of Messages per second, default = 8, (*1 byte*)
SP_1, SP_1, …, SP_N: Sampling period for each opcode, encoded according to Appendix B (*1 byte each*)
NBS_1, NBS_2…NBS_N: Number of Samples per message (*1 byte each*)
OFFP_1, OFFP_2…OFFP_N: Offset Position (number of bytes) of the First sample, taken from after the last byte of FLAG_DATA. A value of 0xFF means that the BS does not want to collect samples for this OpCode. (*1 byte each*).

Note: N is the same as in AVAILABLE_OPCODES ACK. Opcodes NOT wanted by the BS will have an OFFP of 0xFF, but must be present in this message anyways. In the first version of CPOD, N will always be 9.

Example:
BS-REQ: FF 1E 50 08 01 20 00 01 20 30 04 08 60 02 02 6C 02 02 6F 02 02 72 20 01 75 20 01 77 20 01 79 01 A7 E6

NEXT_PACKET_DOWNLOAD ACK: [SIZE-1 bytes]

LG_CMD	LG_DATA

LG_CMD: Command that was logged at the time of transmission (*1 byte*)
LG_DATA: Sample data that was logged at the time of transmission. Its SIZE and meaning depends on the previous commands [previous sampling rate commands, etc…. as well as the accompanying ACK in the LG_CMD] (*SIZE-2 Bytes*).

Note: CPOD logs every message that it transmits by only writing the (CMD + DATA) to the flash memory and its own SIZE tag to specify the size of CMD + DATA. This (CMD + DATA) will be read back to the BS, which will be able to set it parameters and decode stream of data directly from this sequence of logged data.

NEXT_PACKET_STREAMING ACK, NEXT_PACKET_LOGGING ACK:
[1 + SizeOf(FLAG_DATA) + NB_1 + NB_2 + … + NB_M bytes]

FLAG	FLAG_DATA	DATA_1	DATA_2	……..	DATA_M

FLAG: Information Flag for this particular packet (see below) (*1 byte*)
FLAG_DATA: Data corresponding to the FLAG byte, if needed (*0-? Bytes*)

DATA_1*:* Sample data for Opcode OP_1, encoded on NB_1 bytes (*NB_1 bytes*)
DATA_2: Sample data for Opcode OP_2, encoded on NB_1 bytes (*NB_2 bytes*)
…….
***DATA_M*:** Sample data for Opcode OP_M, encoded on NB_M bytes (*NB_M bytes*)

Note: *FLAG* is one byte ORed combination of the following:

- *EVENT_MARK* [0x01, bit0]: message is marked with an event. No influence on FLAG_DATA
- *LOST_DATA* [0x02, bit1]: CPOD acquired data faster than it could send it. FLAG_DATA contains number of messages lost (one byte). If this flag is set, the flag data byte will be the first in the sequence of flag data bytes.
- *ENCRYPTED* [0x04, bit2]: message is encrypted. No influence on *FLAG_DATA*
- *BLOOD_PRESSURE* [0x08, bit3]: message contains blood pressure sample values [Syst and then Diast], encoded in 2x2 = 4 bytes. These bytes are the second set of flag data bytes.
- *GPS* [0x10, bit4]: message contains GPS data encoded in 64 bytes according to the Garmin data structure below. These bytes are the third set of flag data bytes.

  GPS Data Structure (Garmin Format):

```
float alt; /* altitude above WGS 84 ellipsoid (meters) */
float epe; /* estimated position error, 2 sigma (meters) */
float eph; /* epe, but horizontal only (meters) */
float epv; /* epe, but vertical only (meters) */
int fix; /* type of position fix */
double tow; /* time of week (seconds) */
Radian_Type posn; /* latitude and longitude (radians) */
float east; /* velocity east (meters/second) */
float north; /* velocity north (meters/second) */
float up; /* velocity up (meters/second) */
float msl_hght; /* height of WGS 84 ellipsoid above MSL (meters)*/
int leap_scnds; /* difference between GPS and UTC (seconds) */
long wn_days; /* week number days */
```

  where
  float = 32 bit IEEE format floating point (1 sign bit, 8 exponent bits, 23 mantissa bits),
  long = 32 bit integer (signed unless the unsigned keyword is present)
  int = 16 bit integer (signed unless the unsigned keyword is present)
  double = 64 bit IEEE format floating point data (1 sign bit, 11 exponent bits, 52 mantissa bits),
  and Radian = two doubles (latitude and longitude in radians).

- *CO2* [0x20, bit5]: message contains CO2 data encoded in 40 bytes according to the data structure below. These bytes are the third set of flag data bytes.

  CO2 Data Structure:

  0x20
  Time

0x7C
EtCO2
0x7C
FiCO2
0x7C
RR
0x7C
SpO2
0x7C
PR

Where:
Time: Time since device was turned on (session started), hh:mm:ss, 8bytes, ASCII
EtCO2 – Endtidal CO2, amount of CO2 present at the end of exhalation, 7bytes, ASCII
FiCO2 – Fraction Inspired CO2, amount of CO2 present during inhalation, 7bytes, ASCII
RR – Respiration Rate, 4bytes, ASCII
SpO2 – Pulse Oximetry from Nellcor oximax oximeter, 4bytes, ASCII
PR – Pulse Rate from Nellcor oximax oximeter, 4bytes, ASCII

Example:
```
00:08:05|   41  |    2  |  21| 100|  75
```

2030 303A 3038 3A30 357C 2020 2034 3120 207C 2020 2020 3220 207C 2020 3231 7C20 3130 307C 2020 3735 0A0D

Size of *FLAG_DATA* depends on *FLAG*, and is only known as we are decoding *FLAG*. Bytes of data in FLAG_DATA must be read in the order of the above list: Lost Data byte first (1 Byte), then Blood Pressure bytes (4 bytes).

SET_TIME REQ, SET_TIME ACK: [7 bytes]

SEC	MIN	HRS	DAY	MONTH	WKDAY	YEAR

SEC : Seconds *(1byte)*
MIN : Minutes *(1byte)*
HRS : Hours *(1byte)*
DAY : Day of the Month *(1byte)*
MONTH : Month *(1byte)*
WKDAY : Weekday *(1byte)*
YEAR : Year *(1byte)*

STATUS ACK: [24 bytes]

CSA	PAGEH	PAGEL	CSAR	PAGERDH	PAGERDL	BUFORH	BUFORL
BUFN	HSZ	MLSZ	STKPTR	PORTA	PORTB	PORTC	PORTD
PORTE	HRH	HRL	SPO2H	SPO2L	BPMSG	SPMSG	BUFREG

CSA : DF_CS_ADDR - Data Flash Chip Select Address *(1byte)*

PAGEH : DF_PAGE_H - Data Flash Page Address High Byte *(1byte)*
PAGEL : DF_PAGE_L - Data Flash Page Address Low Byte *(1byte)*
CSAR : DF_CS_ADDR_RD - Data Flash Chip Select Address for Read *(1byte)*
PAGERDH : DF_PAGE_RD_H - Data Flash Page Address Read High Byte *(1byte)*
PAGERDL : DF_PAGE_RD_L - Data Flash Page Address Read Low Byte *(1byte)*
BUFORH : DF_BUF_OFF_RD_H - SRAM Buffer Offset for Read High Byte *(1byte)*
BUFORL : DF_BUF_OFF_RD_L - SRAM Buffer Offset for Read Low Byte *(1byte)*
BUFN : DF_BUF_OFF_N - Number of Messages already stored in SRAM buffer *(1byte)*
HSZ : HEADER_SIZE - Header length of current message *(1byte)*
MLSZ : MSGLOG_SIZE - Size of current message to be logged *(1byte)*
MODEREG : MODEREG – Mode Register *(1byte)*
MSGCOUNT : MSGCOUNT – Number of messages available for streaming *(1byte)*
PORTB : PORTB - I/O Port B *(1byte)*
PORTC : PORTC - I/O Port C *(1byte)*
PORTD : PORTD - I/O Port D *(1byte)*
PORTE : PORTE - I/O Port E *(1byte)*
HRH : HR_HI - Heart Rate High Byte *(1byte)*
HRL : HR_LO - Heart Rate Low Byte *(1byte)*
SPO2H : SPO2_HI - SpO2 High Byte *(1byte)*
SPO2L : SPO2_LO - SpO2 Low Byte *(1byte)*
BPMSG : BPMESG - Bytes per Message *(1byte)*
SPMSG : SPMESG - Samples per Message *(1byte)*
BUFREG : BUFREG - Samples currently stored in message buffer *(1byte)*

Example:
BS-REQ: FF 02 B0 01 13 23 (SEQ=1)
CPOD-ACK: FF 1A 0B 01 00 0B 00 81 7A 00 00 03 03 7E BA 14 E4 B1 DF 05 00 00 00 00 7B 51 0D 01 39 97

HANDSHAKE_ACK: [5 bytes]

SNH	SNM	SNL	CONN_TYPE

SNH : SERIAL_H – Serial Number High Byte *(1byte)*
SNM : SERIAL_M – Serial Number Middle Byte *(1byte)*
SNL : SERIAL_L – Serial Number Low Byte *(1byte)*
CONN_TYPE : CONN_TYPE – Connection Type, 0x01=hardwired, 0x02=Bluetooth, 0x04=916MHz, 0x00=unknown *(1byte)*
VERH: VERSION_H – Version Number High Byte *(1byte)*
VERL: VERSION_L – Version Number Low Byte *(1byte)*

SIM_ACK: [1 byte]

SIMREG

SIMREG : SIMREG – Simulation Register, 0x00 – real data, 0x01 and 0x02 – simulated stair case data *(1byte);* **(no more simulated ECG in firmware version V2.0 and higher, needed program memory space to add GPS support)**

READ_TIMER ACK: [17 bytes]

SEC	MIN	HRS	DAY	MONTH	WKDAY	YEAR
DA1	CSA1	PAGEH1	PAGEL1	MODE1	DA2	CSA2
PAGEH2	PAGEL2	MODE2				

SEC : Seconds *(1byte)*
MIN : Minutes *(1byte)*
HRS : Hours *(1byte)*
DAY : Day of the Month *(1byte)*
MONTH : Month *(1byte)*
WKDAY : Weekday *(1byte)*
DA1 : DATA_VALID – reads 0xDA if pointer backup No. 1 is valid *(1byte)*
CSA1 : DF_CS_ADDR - Data Flash Chip Select Address backup No. 1 *(1byte)*
PAGEH1 : DF_PAGE_H - Data Flash Page Address High Byte backup No. 1*(1byte)*
PAGEL1 : DF_PAGE_L - Data Flash Page Address Low Byte backup No. 1 *(1byte)*
MODE1 : MODEREG backup No. 1 *(1byte)*
DA2 : DATA_VALID – reads 0xDA if pointer backup No. 2 is valid *(1byte)*
CSA2 : DF_CS_ADDR - Data Flash Chip Select Address backup No. 2 *(1byte)*
PAGEH2 : DF_PAGE_H - Data Flash Page Address High Byte backup No. 2*(1byte)*
PAGEL2 : DF_PAGE_L - Data Flash Page Address Low Byte backup No. 2 *(1byte)*
MODE2 : MODEREG backup No. 2 *(1byte)*

This command reads the contents of the Real-Time-Clock. The first 7 bytes returned contain the time and date values, and the following 10 bytes contain the two backups of the flash memory pointers and the mode register. These backups allow CPOD to be turned off without losing its last storage location in the flash memory. Two backups are used in case of CPOD being turned off while a backup operation is performed. CPOD alternates between these locations every time it writes to flash (i.e. every second). Before each backup, register DATA_VALID is cleared. Then the pointer and mode register values are backed up, and then DATA_VALID is set to DA to signal that the backup was successful. On startup of CPOD, the first backup is reloaded, but if DATA_VALID is not equal to 0xDA, then the second backup set is used.

Appendix A – Opcodes

Pulse Oximetry	0x01
ECG – Lead I	0x21
ECG – Lead II	0x22
ECG – Lead III	0x23
ECG – Lead aVR	0x24
ECG – Lead aVL	0x25
ECG – Lead aVF	0x26
ECG – Lead V1	0x27
ECG – Lead V2	0x28
ECG – Lead V3	0x29
ECG – Lead V4	0x2A
ECG – Lead V5	0x2B
ECG – Lead V6	0x2C
Heart Rate	0x03
Blood Pressure Systolic	0x51
Blood Pressure Diastolic	0x52
Blood Pressure MAP	0x53
Skin Temperature	0x06
Respiration Rate	0x07
Respiration Raw	0x08
Acceleration X	0x31
Acceleration Y	0x32
Acceleration Z	0x33
Activity	0x34

Appendix B – Sampling Periods, Sampling Rates, Opcode Offsets, and Message Size Limitations

Sampling periods are coded on ONE byte, and they are calculated as multiples of $1/256^{th}$ of a second, with the exception of Zero (0) which is taken to be 1 second.

Example:

0 → 1 second
1 → $1/256^{th}$ of a second
2 → $2/256^{th}$ of a second
…
254 → $254/256^{th}$ of a second
255 → $255/256^{th}$ of a second

Default Sampling Rates (Periods). The default sampling rates (sampling periods), resolutions, and bytes per message (1/8 second) are as follows:

Parameter	Sampling Rate	Sampling Period	Resolution (fixed!)	Bytes per Message
ECG II	256 S/sec (3.9ms)	1	12 bits	48 bytes/message
ECG V5	256 S/sec (3.9ms)	1	12 bits	48 bytes/message
Respiration	64 S/sec (15.6ms)	4	12 bits	12bytes/message
Acceleration X	128 S/sec (7.8ms) [1]	2	12 bits	3 bytes/message
Acceleration Y	128 S/sec (7.8ms) [1]	2	12 bits	3 bytes/message
Acceleration Z	128 S/sec (7.8ms) [1]	2	12 bits	3 bytes/message
Temperature	1 S/sec (1000ms)	0 (256)	12 bits	2 bytes/message
SpO2	1 S/sec (1000ms)	0 (256)	12 bits	2 byte/message
Heart Rate	1 S/sec (1000ms)	0 (256)	12 bits	2 bytes/message
Syst. BP	4 S/h	N/A	12 bits	2 bytes (flag data)
Diast. BP	4 S/h	N/A	12 bits	2 bytes (flag data)

[1]averaged and stored at 16 S/sec

The resolution of all parameters is fixed to 12 bits and cannot be changed in CPOD's firmware. Blood Pressure is sampled only a few times per hour, and is therefore included as flag data. The flag byte indicates that blood pressure data follows when bit 3 is set (0x08).

Sampling periods are stored in CPOD in a 1-byte sampling period variable as multiples of the fastest sampling period (here ECG). Applying this to the above default values, the sampling period variable for ECG is therefore 1 by definition, for respiration it will be 4, for acceleration it will be 2, and for temperature it will be 256.
Choosing other sampling periods is possible but all sampling periods must be one of the following values: 1, 2, 4, 8, 16, 32, 0. It must also be ensured that the sampling periods chosen do not result in a message size greater than the maximum message size of 132 (including 3 frame bytes). See section "Message Size" for details.

Flash Memory Structure. The Flash memory in CPOD consists of four 64-Mbit flash chips. Each flash chip has 8192 pages of 1056 bytes each. The amount of data that can be stored per second is limited to 1 page or 1056 bytes. Four flash chips with 8192 pages each yield a total capacity of 32768 pages. That equals a total storage time of 32768 seconds, or 9.1 hours.

Message Size. The message size is fixed as follows: 1 Message = 1/8 seconds = 125ms = 132 bytes max. (1056bytes/8). Based on the default sampling parameters listed above, the default message size that is stored in flash is 123 data bytes plus 1 flag byte, plus 3 frame bytes (size, cmd, and seq) = 127 bytes/message. This leaves (1056-8*127) = 40 bytes for occasional flag data: time stamp, event, and BP data (none of these more frequent than once per second, so there will never be more than one long message with flag data per 1056-byte page).

Opcode Offsets and Opcode Selection. Whenever data is being acquired by CPOD, it is stored at a fixed location of the CPOD message *data* buffer (122 bytes default). After this buffer has been filled (this will take 125ms), it is copied into the 1056 byte RAM buffer of the Flash. This is being repeated 8 times, and then a full second of buffered data is transferred to flash.
Since CPOD is not capable of performing extensive calculations, it has to rely on the base station to compute the buffer location for each parameter byte. The base station needs to compute and send these offsets to CPOD during setup where they will be stored in CPOD's non-volatile EEPROM memory. There is a single-byte offset variable for each parameter specifying the offset counted from the first byte of the data buffer (excluding the flag data byte). For a typical setup where all parameters and default sampling rates are used, these offsets are as follows:

Parameter	**Opcode Offset**	**Byte Locations**
ECG II	0	(48 bytes: 0 ... 47)
ECG V5	48	(48 bytes: 48 ... 95)
Respiration	96	(12 bytes: 96 ... 107)
Acceleration X	108	(3 bytes: 108, 109, 110)
Acceleration Y	111	(3 bytes: 111, 112, 113)
Acceleration Z	114	(3 bytes: 114, 115, 116)
Temperature	117	(2 bytes: 117, 118)
SpO2	119	(2 bytes: 119, 120)
Heart Rate	121	(2 bytes: 121, 122)

Opcode Selection: Any combination of opcodes can be selected during setup (e.g. ECG V5, Acceleration Y, and Heart Rate). If an opcode is NOT included, CPOD needs to be told that by setting the opcode offset value for this opcode to FF. This means, CPOD expects an opcode offset value for each parameter. The opcode offsets need to be sent to CPOD in the parameter order shown above (ECG II, ECG V5, Resp., Accel. X, Y, Z, Temp., SpO2, HR). No opcode offset is necessary for Blood Pressure, since it is included as flag data.
If the default opcode offsets are changed, care must be taken to ensure that the difference between two offsets (or the number of bytes per parameter per message) are divisible by 3. This is due to the way of storing two 12-bit values in 3 bytes. The exception is the case where only one 12-bit parameter per message is stored. In this case, the number of byte locations will be 2 by necessity (e.g. default value for Temperature, see above table). The 12-bit value in these two bytes will be left aligned, i.e. the 4 least significant bits of the second byte will be unused.