**Interface Requirement Specifications**

**IRS**

**ZipThaw 202**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Name** | **Position** | **Date** | **Signature** |
| Composed | Moshe Gershony | System engineer |  |  |
| Reviewed |  |  |  |  |
| Approved |  |  |  |  |
| Approved |  |  |  |  |
| Approved |  |  |  |  |

Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **version** | **Change details** | **Chapters** | **Editor** |
| 01.04.18 | 1.0 | First edition - draft |  | Moshe Gershony |
| 07.05.18 | 4.0 | Communication details, Pre-heating Op Code, Thawing Op Code, Status parameters | 3.1.11, 8.1, 8.2, 9 | Moshe Gershony |
| 03.07.18 | 5.0 | System shut OFF Opcode  Sensor and Data table – additional sensors and data.  Error status table – additional sensors  Cancellation of door lock sensor | 6.1, 8.2  9.3  9.5  9.2 | Moshe Gershony |
| 24.10.18 | 6.0 | Cancelled - temperature sensors Highlighted yellow | 9.3 | Moshe Gershony |
| 28.10.18 | 7.0 | Text changes Highlighted blue | 9.3, 9.4, 9.5 | Moshe Gershony |
| 28.10.18  5.11.18 | 7.0 | Opcode length field  Opcode: Turn OFF System | 8.1  8.2 | Moshe Gershony |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Table of Contents:

1. General
   1. Identification

This document contains the interface requirement specifications for the communication protocol between the processor of the System On Module (SOM) and the embedded Micro Controller Unit (MCU) of the ZipThaw system.

* 1. System description  
     TBD
  2. Abbreviations
     1. BIT – Built in Test.
     2. LOG – Log file.
     3. CRC – Cyclic Redundancy Check.
     4. SBC – Single Board Computer.
     5. SOM – System on Module.
     6. MCU – Micro Controller Unit.
     7. Software (SW) – computer program that is running in the SOM.
     8. Firmware – computer program that is running in the MCU.
     9. NVM – Non-Volatile Memory.
     10. LED – Light Emitting Diode.

1. Applicable Documents
   1. System documents
      1. MRD.
      2. DID.
      3. Risk Analysis.
   2. Applicable Standards
      1. Add …
2. Communication interface – SOM / MCU
   1. Communication channel functions  
      The communication channel is used for communicating bi-directional messages between the SOM and the MCU. The SOM sends Opcode messages and the MCU responds by sending various status and data messages. This document describes the communication protocol for the ZipThaw model 202 including two thawing chambers. The framework of this protocol may support larger number of thawing chambers by minor modifications.
   2. Interface type  
      Serial communication RS485, Multi-Drop (optional), Half Duplex:
      1. Baud rate – 115,200 bits per second.
      2. 8 Data bits, 1 Start bit, 1 Stop bit, NO Parity.
      3. MSB of every byte is transmitted first.
      4. Most Significant Byte of every Word (Word – 2 bytes) is transmitted first.
   3. Protocol
      1. Master / Slave communication type. SOM – master, MCUs – slaves.
      2. The SOM will send several (TBD) Opcode messages per second to each of the MCUs with a specific ID for each Opcode message.
      3. The addressed MCU will respond immediately to each Opcode by a relevant status or data message. This message will include the same ID of the Opcode that was just received from the SOM.
      4. For “keep alive” purpose, when there is no request for any specific operation, the SOM will send the Opcode – “Read system status” to each MCU repeatedly.
      5. The SOM is defined as the Master for communication purpose only. The control of the ZipThaw system will be managed by the embedded MCUs firmware.
      6. This protocol supports communication between a single SOM and several MCUs. A configuration of a single MCU is also supported.
      7. CRC8 will be use in every message. An example for CRC8 calculation is shown at

Appendix A.



1. BIT – Built In Test
   1. Power-up BIT – on every power-up, the system will perform a general built in test to check system sub-systems.
   2. Several types of built in tests will be available to be performed on demand, for example after some corrective maintenance. The GUI access to initiate operation of such tests will be from the technician screen. Each specific BIT will be initiated by sending a specific Opcode message from the SOM to the MCU – TBD.
2. Communication problem management
   1. Detection of communication problem will be performed mainly by verification of message CRC by the receiving side. Another type of problem may be timeout failure in receiving messages.
   2. In case the MCU detects a communication error it will notify the SOM by sending a status message with indication for communication error. The SOM will then re-send the same message again.
   3. In case the SOM detects an error, it will re-send the last Opcode again and wait for the MCU response.
   4. In case of communication error, five retries will be performed. If communication retries are continuously unsuccessful, an error message will be displayed on the screen indicating communication malfunction and the system will stop activities.
   5. In case that a communication malfunction is declared in a middle of thawing process the thawing process will be continued to its completion, indication for the user – TBD.
3. Main system operations
   1. General operations
      1. Power-up BIT.
      2. Pre-heat.
      3. Stand by.
      4. Select chamber.
      5. Start thawing.
      6. Stop thawing.
      7. Turn OFF system.
   2. Technician main operations
      1. Enter technician mode
      2. Turn ON/OFF a selected heating element.
      3. Turn ON/OFF a selected agitation motor. For each operation the motor will perform a complete cycle.
      4. Turn ON/OFF a selected door lock.
      5. Turn ON/OFF a selected indication LED.
      6. Turn ON/OFF a buzzer.
      7. Technician BITs – TBD.
      8. Exit technician mode.
   3. – Parameter setting
      1. Set parameters of pre-heating process.
      2. Set parameters of thawing process.
4. Boot Loader (check…………..)
   1. The system enables upgrading of firmware of the MCU – boot loading. Boot loading process:
      1. The SOM will send Opcode “Prepare for boot loading”.
      2. The MCU will respond by status indication if ready or not.
      3. If the MCU is not ready it means that a process is still in progress and that SOM will be notified by the process status. The SOM will wait for completion of the process.
      4. When the MCU is ready it will respond with a ready status indication. The SOM will send the new version to the MCU.

1. Structure of Opcode messages
   1. Each Opcode message is constructed of a series of fields as follows:

|  |  |  |
| --- | --- | --- |
| **Field #** | **Field description** | **Length in bytes** |
| 1 | Preamble - $$ | 2 |
| 2 | Opcode length – including all Opcode fields | 1 |
| 3 | Message ID – any number for ID matching with status message | 2 |
| 4 | Sending station ID – for multidrop configuration | 1 |
| 5 | Receiving station ID – for multidrop configuration | 1 |
| 6 | Opcode Type – see Opcode table below | 1 |
| 7 | Opcode Sub-Type – see Opcode table below | 1 |
| 8 | Data – Depends on Opcode Type and Sub-Type | 21 |
| 9 | CRC 8 bit (see appendix A) | 1 |

* 1. Opcode table - fields # 5, 6, 7 in Opcode massage:

| **Opcode Name** | **Opcode Type** | **Opcode Sub Type** | **Data length** | **Data values** | **Remarks** |
| --- | --- | --- | --- | --- | --- |
| **Status Opcodes** | | | | | |
| Read system status | 1 | 10 | 0 | - | All sensors, including keep alive |
| **BIT Opcodes** | | | | | |
| Power-up BIT | 2 | 10 | 0 | - |  |
| Technician BIT | 2 | 20 |  |  | TBD |
| **Parameter setting Opcodes** | | | | | |
| Pre-heating Parameter | 3 | 10 |  | Over- heat temp. (2 bytes), time (2 bytes), PID (6 bytes) X 3 for 3 different bags | Different for chambers? |
| Thawing Parameter | 3 | 20 |  | PID (6 bytes) X 3 for 3 different bags | Different for chambers? |
| **Operation Opcodes** | | | | | |
| Pre-heat | 4 | 10 |  |  |  |
| Stand by | 4 | 20 |  |  |  |
| Start thawing | 4 | 40 |  |  | Per chamber |
| Stop thawing | 4 | 50 |  |  | Per chamber |
| Turn OFF system | 4 | 60 |  |  |  |
| **Service, maintenance, development, research** | | | | | |
| Enter technician mode | 5 | 10 |  |  |  |
| Exit technician mode | 5 | 20 |  |  |  |
| Turn ON heating element | 6 | 10 |  | Heating element # |  |
| Turn OFF heating element | 6 | 20 |  | Heating element # |  |
| Turn ON agitation motor | 6 | 30 |  |  |  |
| Turn OFF agitation motor | 6 | 40 |  |  | Stop at home position |
| Turn ON door lock | 7 | 10 |  |  |  |
| Turn OFF door lock | 7 | 20 |  |  |  |
| Turn ON LED | 8 | 30 |  | LED color # |  |
| Turn OFF LED | 8 | 10 |  | LED color # |  |
| Turn ON buzzer | 8 | 20 |  |  |  |
| Turn OFF buzzer | 8 | 30 |  |  |  |

1. Structure of Status message
   1. Each Status Message is constructed of a series of fields as follows:

|  |  |  |
| --- | --- | --- |
| **Field #** | **Field description** | **Length in bytes** |
| 1 | Preamble - $$ | 2 |
| 2 | Status Message length | 1 |
| 3 | Message ID | 2 |
| 4 | Sending station ID – for multidrop configuration | 1 |
| 5 | Receiving station ID – for multidrop configuration | 1 |
| 6 | Single bit sensors – 1 bit per sensor, up to 40 sensors – see table | 5 |
| 7 | Data sensors – 2 bytes per sensor, up to 29 sensors – see table | 58 |
| 8 | Process status – 51 processes, 1-byte progress per process (%) | 51 |
| 9 | Errors – 48 different error types, 1 bit per error type | 6 |
| 10 | CRC 8 bit | 1 |

1. 1. Single bit sensors and status table - Field # 7 in Status Message:

| **Byte #** | **Bit #** | **Sensor Description** |
| --- | --- | --- |
| 1 (LSB) | 0 | Door closed sensor |
| 1 | 1 | Spare |
| 1 | 2 | Spare |
| 1 | 3 | Spare |
| 1 | 4 | Spare |
| 1 | 5 | Spare |
| 1 | 6 | Spare |
| 1 | 7 | Spare |
| 2 | 0 | Agitation motor home position |
| 2 | 1 | Spare |
| 2 | 2 | Spare |
| 2 | 3 | Spare |
| 2 | 4 | Spare |
| 2 | 5 | Spare |
| 2 | 6 | Spare |
| 2 | 7 | Spare |
| 3 | 0 | Ready for Boot Loading, 0 – not ready, 1 - ready |
| 3 | 1 | Spare |
| 3 | 2 | Spare |
| 3 | 3 | Spare |
| 3 | 4 | Spare |
| 3 | 5 | Spare |
| 3 | 6 | Spare |
| 3 | 7 | Spare |
| 4 | 0 - 7 | Spares |
| 5 | 0 - 7 | Spares |

* 1. Sensor and data table - Field # 8 in Status Message:

| **Byte #** | **Sensor and Data Description** | **Measurement unit** |
| --- | --- | --- |
| 1 - 2 (LSB) | ~~Temp. sensor – heat element #~~1 Spare | Degrees x 10 (1 decimal point) |
| 3 – 4 | Temp. sensor #1 – cushion #1 - body | Degrees x 10 (1 decimal point) |
| 5 – 6 | ~~Temp. sensor #2 – cushion #1~~ Spare | Degrees x 10 (1 decimal point) |
| 7 – 8 | ~~Temp. sensor – heat element #~~2 Spare | Degrees x 10 (1 decimal point) |
| 9 – 10 | Temp. sensor #1 – cushion #2 - door | Degrees x 10 (1 decimal point) |
| 11 – 12 | ~~Temp. sensor #2 – cushion #2~~ Spare | Degrees x 10 (1 decimal point) |
| 13 – 14 | Temp. sensor ZipSleeve – RFID | Degrees x 10 (1 decimal point) |
| 15 – 16 | Temp. sensor ZipSleeve – IR | Degrees x 10 (1 decimal point) |
| 17 – 18 | Electronic board temperature sensor | Degrees x 10 (1 decimal point) |
| 19 – 20 | Weight of plasma bag | Grams |
| 21 – 22 | Time from End of Thawing process (time is transferred till detection of Door Open) | Seconds |
| 23 – 24 | Spare |  |
| 25 – 26 | Spare |  |
| 27 – 28 | Spare |  |
| 29 – 30 | Spare |  |
| 31 – 32 | Spare |  |
| 33 – 34 | 3.3V | mV |
| 35 – 36 | 24V | mV |
| 37 – 38 | Spare |  |
| 39 – 40 | Spare |  |
| 41 – 42 | Spare |  |
| 43 – 44 | Spare |  |
| 45 - 46 | Spare |  |
| 47 – 48 | Spare |  |
| 49 - 50 | Spare |  |
| 51 – 52 | Spare |  |
| 53 - 54 | Spare |  |
| 55 – 56 | Spare |  |
| 57 - 58 | Spare |  |

* 1. Process status table:

Description of process status value: 0 to 100 – completion percentage, FF – inactive process.

| **Byte #** | **Process Description** | **Remarks** |
| --- | --- | --- |
| 1 (LSB) | Pre- heating chamber | Without plasma bag |
| 2 | Stand by chamber | Without plasma bag |
| 3 | Thawing chamber |  |
| 4 | Power-up BIT |  |
| 5 | Technician BIT |  |
| 6 | Spare |  |
| 7 | Spare |  |
| 8 | Spare |  |
| 9 | Spare |  |
| 10 | Spare |  |
| 11 | Spare |  |
| 12 | Spare |  |
| 13 | Spare |  |
| 14 | Spare |  |
| 15 | Spare |  |
| 16 | Spare |  |
| 17 | Spare |  |
| 18 | Spare |  |
| 19 | Spare |  |
| 20 - 49 | Spare |  |
| 50 | Firmware version MSB |  |
| 51 | Firmware version LSB |  |

* 1. Error status table:

| **Byte #** | **Bit #** | **Error Description** |
| --- | --- | --- |
| 1 (LSB) | 0 | Door closed sensor |
| 1 | 1 | Spare |
| 1 | 2 | Agitation motor home position #1 – timeout |
| 1 | 3 | 3.3V |
| 1 | 4 | 24V |
| 1 | 5 | Spare |
| 1 | 6 | Spare |
| 1 | 7 | Spare |
| 2 | 0 | Temp. sensor – chamber heat element #1 |
| 2 | 1 | Temp. sensor – cushion #1 (body) – 1st sensor |
| 2 | 2 | Temp. sensor – cushion #1 – (body) 2nd sensor - safety |
| 2 | 3 | Temp. sensor – chamber heat element #2 |
| 2 | 4 | Temp. sensor – cushion #2 (door) – 1st sensor |
| 2 | 5 | Temp. sensor – cushion #2 (door) – 2nd sensor - safety |
| 2 | 6 | Temp. sensor ZipSleeve – RFID |
| 2 | 7 | Temp. sensor ZipSleeve – IR |
| 3 | 0 | Spare |
| 3 | 1 | Spare |
| 3 | 2 | Spare |
| 3 | 3 | Spare |
| 3 | 4 | Spare |
| 3 | 5 | Spare |
| 3 | 6 | Spare |
| 3 | 7 | Spare |
| 4 | 0 | Spare |
| 4 | 1 | Spare |
| 4 | 2 | Spare |
| 4 | 3 | Spare |
| 4 | 4 | Spare |
| 4 | 5 | Spare |
| 4 | 6 | Spare |
| 4 | 7 | Spare |
| 5 - 6 | 0 - 7 | Spare |

1. Structure of Boot Loading Data – TBD.

Appendix A

CRC is calculated on all message bytes except the CRC itself.

CRC8 Calculation:

const uint8\_t crc8\_Table[] =

{

                0,  94, 188, 226,  97,  63, 221, 131, 194, 156, 126,  32, 163, 253,  31,  65,

                157, 195,  33, 127, 252, 162,  64,  30,  95,   1, 227, 189,  62,  96, 130, 220,

                35, 125, 159, 193,  66,  28, 254, 160, 225, 191,  93,   3, 128, 222,  60,  98,

                190, 224,   2,  92, 223, 129,  99,  61, 124,  34, 192, 158,  29,  67, 161, 255,

                70,  24, 250, 164,  39, 121, 155, 197, 132, 218,  56, 102, 229, 187,  89,   7,

                219, 133, 103,  57, 186, 228,   6,  88,  25,  71, 165, 251, 120,  38, 196, 154,

                101,  59, 217, 135,   4,  90, 184, 230, 167, 249,  27,  69, 198, 152, 122,  36,

                248, 166,  68,  26, 153, 199,  37, 123,  58, 100, 134, 216,  91,   5, 231, 185,

                140, 210,  48, 110, 237, 179,  81,  15,  78,  16, 242, 172,  47, 113, 147, 205,

                17,  79, 173, 243, 112,  46, 204, 146, 211, 141, 111,  49, 178, 236,  14,  80,

                175, 241,  19,  77, 206, 144, 114,  44, 109,  51, 209, 143,  12,  82, 176, 238,

                50, 108, 142, 208,  83,  13, 239, 177, 240, 174,  76,  18, 145, 207,  45, 115,

                202, 148, 118,  40, 171, 245,  23,  73,   8,  86, 180, 234, 105,  55, 213, 139,

                87,   9, 235, 181,  54, 104, 138, 212, 149, 203,  41, 119, 244, 170,  72,  22,

                233, 183,  85,  11, 136, 214,  52, 106,  43, 117, 151, 201,  74,  20, 246, 168,

                116,  42, 200, 150,  21,  75, 169, 247, 182, 232,  10,  84, 215, 137, 107,  53

};

uint8\_t Calculate\_crc8 (uint8\_t \*msg\_array, uint8\_t msg\_size)

{

                uint8\_t crc = 0x96; //initial\_Value

                for (uint8\_t idx = 0; idx < msg\_size; idx++)

                {

                                crc = crc8\_Table[crc ^ (uint8\_t)msg\_array[idx]];

                }

               return crc;

}