***Application Memory Structure***

The application uses an external 8GB SD card to store long-term application memory. Information is stored in sets of 512 contiguous bytes called blocks; the addresses of each of these blocks are called sectors. Each byte is ASCII-formatted, and each byte string is terminated by the string-terminating character ‘\0’. The application memory consists of three sections: the signature, the manifest, and the database.

1. The **signature** (Start Sector = 0, Size = 1) is a static block located in the first sector of application memory. It contains sub-signatures of the current MCU (3 bytes) and card (5 bytes). On detection of new card, the MCU rewrites the signature and wipes the new card, severing the link with its previous card. The signature is formatted as follows:

{ *MCU[2]* } { *MCU[1]* } { *MCU[0]* } { *CRD[3]* } { *CRD[2]* } { *CRD [1]* } { *CRD [0]* }

1. The **manifest** (Start Sector = 1, Size = 255) is a dynamic block chain located in the first 128 kB of application memory. It contains block-sized markers that contain the address and integrity of files in the database. The structure of the markers is as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parameter** | **Block Offset** | **Data String** | | | |
| Type | 0 | T |
| Start | 2 | S[7] | … | S[0] |
| End | 11 | E[7] | … | E[0] |

* *Type*: Data type
* *Start*: File starting sector
* *End*: At marker write, this is empty indicating that the entry has NOT been completed. Upon file completion, this is updated to contain the ending sector

1. The **database** (Start Sector = 256, Size = ~ 8 GB) is a serialized list of files following the manifest.

The file types are as followed:

* ***Trace (Entry Type 0)***

A trace contains a serialized collection of route-descriptive coordinates called nodes following a statically-sized bitmap of routers. The bitmap is a dynamically updated 512 kB (32 x 32 quadrant-size) string of routers. Routers contain addresses of quadrant-starting /continuing coordinates. Consequentially, the application cannot store information pass 16 quadrants in any direction from the start location, and can only hold, at most, 56 quadrant-starting/continuing nodes. The structure of routers is as followed:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Param** | **Block Off.** | **Data String** | | | |
| Type | 0 | T |
| Entry ID | 2 | N[2] | N[1] | N[0] |
| Addr0 | 6 | A0[7] | … | A0[0] |
| … | … | … | … | … |
| AddrN | … | AN[7] | … | AN[0] |

* *Entry* *ID*: File number
* *Type*: Data type of node
* *Addr0*: Address of a quadrant-starting node
* *Addr1*-N: Addresses of quadrant-continuing nodes

The actual data in nodes vary with node type, but the generic structure of nodes is as followed:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parameter** | **Block Off.** | **Data String** | | | |
| Type | 0 | T |
| Entry ID | 2 | N[2] | N[1] | N[0] |
| X | 6 | SX | X[9] | … | X[0] |
| Y | 18 | SY | Y[8] | … | Y[0] |
| Time | 29 | UTCT[11] | … | UTCT[0] |
| Date | 52 | UTCD[7] | … | UTCD[0] |
| QuadCol | 61 | SC | C[1] | C[0] |
| QuadRow | 65 | SR | R[1] | R[0] |

* *Entry* *ID*: File number
* *Type*: Data type of node
* *X*: Either the coordinate’s absolute longitude or relative longitude
* *Y*: Either the coordinate’s absolute latitude or relative latitude
* *Time*: The time at which the coordinate is written
* *Date*: The date at which the coordinate is written
* *Quad*: Quadrant node belongs to (SR/SC are *Sign-of-Row* and *Sign-of-Column* respectively)

The structures of each type of node is as follows:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Node** | **Type** | **Entry ID** | **X** | **Y** | **UTC** | **Quad** |
| Normal | YES | YES | OFFSET | OFFSET | INVISBLE | YES |
| Super | YES | YES | OFFSET | OFFSET | VISIBLE | YES |
| Origin | YES | YES | ABSOLUTE | ABSOLUTE | VISIBLE | YES |
| Reference | YES | YES | ABSOLUTE | ABSOLUTE | INVISIBLE | YES |

* *Normal* – A node that links other nodes visually. The recorded position of this node is an offset by an origin/reference node.
* *Super* – A normal node with UTC data. The user can examine this type of node.
* *Origin* – A super node, except the recorded position is the absolute position of the user. This node is created once at the beginning of a trace.
* *Reference* – A node that is NOT represented visually, but anchors normal and super nodes.
* ***Singular Coordinate (Entry Type 1)***

A singular coordinate is simply a coordinate with an origin-node-payload structure, but with NO quadrant information. The user can create a singular coordinate by selecting the corresponding action in the main menu. It occupies one block in the manifest and in the database.

The structure of application memory with an incomplete trace file following two singular coordinate files can be visually represented as the following:

