File structure:

|  |  |  |
| --- | --- | --- |
| **Position** | **Block** | **Additional info** |
| 1 | Header | photon = ctb. PHZ is different |
| 2 | Preview1 | Same format for photon/ctb/phz files. |
| 3 | Preview2 | Same format for photon/ctb/phz files. |
| 4 | Print Parameters | Not present in PHZ |
| 5 | Slicer parameters | Not present in PHZ |
| 6 | Machine name string | ASCII |
| 7 | Layer definition table | Same format for photon/ctb/phz files. |
| 8 | Layer images data | photon ≠ ctb ≠ phz |

Header type photon v2 & CTB (size=0x70):

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Type** | **Field** | **Additional info** |
| 0x00 | int | File magic ID | 0x12FD0019 = .photon0x12FD0086 = .ctb |
| 0x04 | int | File version | 2 |
| 0x08 | float | Bed X size in mm |  |
| 0x0C | float | Bed Y size in mm |  |
| 0x10 | float | Bed Z size in mm |  |
| 0x14 | int | unknown1 | 0 |
| 0x18 | int | unknown2 | 0 |
| 0x1C | int | unknown3 | 10.0 |
| 0x20 | float | Layer Height in mm |  |
| 0x24 | float | Exposure Time in seconds |  |
| 0x28 | float | Bottom Exposure Time in seconds |  |
| 0x2C | float | Off Time in seconds |  |
| 0x30 | int | Nr of Bottom Layers |  |
| 0x34 | int | LCD resolution X  |  |
| 0x38 | int | LCD resolution Y  |  |
| 0x3C | int | Preview One Offset Address  | Pointer to preview 1(large) |
| 0x40 | int | Layers Definition Offset Address | Pointer to layers |
| 0x44 | int | Nr of Layers  |  |
| 0x48 | int | Preview Two Offset Address  | Pointer to preview 2(small) |
| 0x4C | int | Print Time in seconds  |  |
| 0x50 | int | Project Type  | 1=LCD\_mirror |
| 0x54 | int | Print Parameters Offset Address | Pointer to print parameters |
| 0x58 | int | Print Parameters Size  | Size of print param. area0x3c |
| 0x5C | int | AntiAliasing Level  | 1, 2, 4, 8 |
| 0x60 | short | Light PWM  |  |
| 0x62 | short | Bottom Light PWM |  |
| 0x64 | int | padding1 | 0 (photon)0x94337afd (cbd) |
| 0x68 | int | Slicer parameters Offset Address | Pointer to Slicer parameters |
| 0x6C | int | Slicer parameters size | Size of Slicer param. area0x4c |

Header phrozen sonic mini (size=0xd8):

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Type** | **Field** | **Additional info** |
| 0x00 | int | File magic ID | 0x9fda83ae |
| 0x04 | int | File version | 2 |
| 0x08 | float | Layer Height in mm |  |
| 0x0C | float | Exposure Time in seconds |  |
| 0x10 | float | Bottom Exposure Time in seconds |  |
| 0x14 | int | Nr of Bottom Layers |  |
| 0x18 | int | LCD resolution X  |  |
| 0x1C | int | LCD resolution Y  |  |
| 0x20 | int | Preview One Offset Address  | Pointer to preview 1(large) |
| 0x24 | int | Layers Definition Offset Address | Pointer to layers |
| 0x28 | int | Nr of Layers  |  |
| 0x2C | int | Preview Two Offset Address  | Pointer to preview 2(small) |
| 0x30 | int | Print Time in seconds  |  |
| 0x34 | int | Project Type  | 1=LCD\_mirror |
| 0x38 | int | AntiAliasing Level  | 1 (set to 4 in chitubox) |
| 0x3C | short | Light PWM  |  |
| 0x3E | short | Bottom Light PWM |  |
| 0x40 | int | unknown1 | 0 |
| 0x44 | int | unknown2 | 0 |
| 0x48 | int | unknown3 | 10.0 |
| 0x4C | float | Bed X size in mm |  |
| 0x50 | float | Bed Y size in mm |  |
| 0x54 | float | Bed Z size in mm |  |
| 0x58 | int | padding1Random number using CurrentTime as seed? | from header data0x4f62153a |
| 0x5C | float | Bottom Light Off Delay in seconds |  |
| 0x60 | float | Off Time in seconds |  |
| 0x64 | int | Nr of Bottom Layers |  |
| 0x68 | float | P1  | from print parameters0 |
| 0x6C | float | Bottom Lift Distance in mm |  |
| 0x70 | float | Bottom Lift Speed in mm/min |  |
| 0x74 | float | Lift Distance in mm |  |
| 0x78 | float | Lift Speed in mm/min |  |
| 0x7C | float | Retract Speed in mm/min |  |
| 0x80 | float | Print Volume in ml |  |
| 0x84 | float | Pring Weight in grams |  |
| 0x88 | float | Print Cost in dollars |  |
| 0x8C |  |  | param\_4[6]0 |
| 0x90 | int | Machine Name String Address | param\_4[7] |
| 0x94 | int | Machine Name String Length | param\_4[8] |
| 0x98 |  |  | param\_4[0]0 |
| 0x9C |  |  | param\_4[1]0 |
| 0xA0 |  |  | param\_4[2] 0 |
| 0xA4 |  |  | param\_4[3] 0 |
| 0xA8 |  |  | param\_4[4] 0 |
| 0xAC |  |  | param\_4[5] 0 |
| 0xB0 |  |  | param\_4[9] 0x1c |
| 0xB4 | int | Epoch timestamp in seconds(chitubox started?) | param\_4[10] 0x0192c9a9 |
| 0xB8 |  |  | param\_4[11]0x4 |
| 0xBC | int | Chitubox version | param\_4[12]0x01060300 |
| 0xC0 | int | Padding | 0 |
| 0xC4 | int | Padding | 0 |
| 0xC8 | int | Padding | 0 |
| 0xCC | int | Padding | 0 |
| 0xD0 | int | Padding | 0 |
| 0xD4 | int | Padding | 0 |

Print parameters (@Print Parameters Offset Address):

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Type** | **Field** | **Additional info** |
| 0x00 | float | Bottom Lift Distance in mm |  |
| 0x04 | float | Bottom Lift Speed in mm/min |  |
| 0x08 | float | Lift Distance in mm |  |
| 0x0C | float | Lift Speed in mm/min |  |
| 0x10 | float | Retract Speed in mm/min |  |
| 0x14 | float | Print Volume in ml |  |
| 0x18 | float | Pring Weight in grams |  |
| 0x1C | float | Print Cost in dollars |  |
| 0x20 | float | Bottom Light Off Delay in seconds |  |
| 0x24 | float | Light Off Delay in seconds |  |
| 0x28 | int | Nr of Bottom Layers |  |
| 0x2C | float | P1 | 0 |
| 0x30 | float | P2 | 0 (photon)0xc2c47a40 (cbd) |
| 0x34 | float | P3 | 0 (photon)0x1a6 (cbd) |
| 0x38 | float | P4 | 0 |

Slicer parameters (@Slicer Parameters Offset Address):

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Type** | **Field** | **Additional info** |
| 0x00 | int |  | 0 |
| 0x04 | int |  | 0 |
| 0x08 | int |  | 0 |
| 0x0C | int |  | 0 |
| 0x10 | int |  | 0 |
| 0x14 | int |  | 0 |
| 0x18 | int |  | 0 |
| 0x1C | int | Machine Name String Address |  |
| 0x20 | int | Machine Name String Length |  |
| 0x24 | int | Machine type? | 0x8=anycubic photon0xF=Mars Pro |
| 0x28 | int | Epoch timestamp in seconds(chitubox started?) |  |
| 0x2C | int | Machine type? | 0x4=anycubic photon0x8=Mars Pro |
| 0x30 | int | Chitubox version | 0x01060300 |
| 0x34 | int |  | 0 |
| 0x38 | int |  | 0 |
| 0x3C |  |  | 0 (cbd)0x379e01d0(photon) |
| 0x40 |  |  | 0 (cbd)0x02b6(photon) |
| 0x44 |  |  | 0 (cbd)0x040d(photon) |
| 0x48 |  |  | 0 (cbd)0x003a0066(photon) |

layer definition (@Layers Definition Offset Address):

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Type** | **Field** | **Additional info** |
| 0x00 | float | layerPositionZ |  |
| 0x04 | float | layerExposure |  |
| 0x08 | float | layerOffTimeSeconds |  |
| 0x0C | int | dataAddress |  |
| 0x10 | int | dataSize |  |
| 0x14 | int | unknown1 |  |
| 0x18 | int | unknown2 |  |
| 0x1C | int | unknown3 |  |
| 0x20 | int | unknown4 |  |

layer data (@Layer Data Address)

RLE for photon. Unknown for others.

 public void unpackLayerImage(byte[] packedLayerImage) {

 clear();

 int x = 0;

 int y = 0;

 for (int i = 0; i < packedLayerImage.length; i++) {

 byte rle = packedLayerImage[i];

 byte colorCode = (byte) ((rle & 0x60) >> 5);

 boolean extended = (rle & 0x80) == 0x80;

 int length = rle & 0x1F;

 if (extended) {

 i++;

 length = (length << 8) | packedLayerImage[i] & 0x00ff;

 }

 for(int xi = x; xi<(x+length); xi++) {

 switch (colorCode) {

 case SUPPORTED:

 supported(xi, y);

 break;

 case CONNECTED:

 unSupported(xi, y);

 break;

 case ISLAND:

 island(xi, y);

 break;

 }

 }

 x += length;

 if (x >= width) {

 y++;

 x = 0;

 }

 }

Preview data (@Preview One/Two Offset Address)

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Type** | **Field** | **Additional info** |
| 0x00 | int | Image Size X |  |
| 0x04 | int | Image Size Y |  |
| 0x08 | int | Image Data Address | Pointer to preview image data |
| 0x0C | int | Image Data Size |  |
| 0x10 | int | unknown1 |  |
| 0x14 | int | unknown2 |  |
| 0x18 | int | unknown3 |  |
| 0x1C | int | unknown4 |  |

Preview image data (@Image Data Address)

 private void decodeImageData() {

 imageData = new int[resolutionX \* resolutionY];

 int d = 0;

 for (int i = 0; i < dataSize; i++) {

 int dot = rawImageData[i] & 0xFF | ((rawImageData[++i] & 0xFF) << 8);

 int color = ((dot & 0xF800) << 8) | ((dot & 0x07C0) << 5) | ((dot & 0x001F) << 3);

// int red = ((dot >> 11) & 0x1F) << 3;

// int green = ((dot >> 6) & 0x1F) << 3;

// int blue = (dot & 0x1F) << 3;

// color = red<<16 | green<<8 | blue;

 int repeat = 1;

 if ((dot & 0x0020) == 0x0020) {

 repeat += rawImageData[++i] & 0xFF | ((rawImageData[++i] & 0x0F) << 8);

 }

 while (repeat > 0) {

 imageData[d++] = color;

 repeat--;

 }

 }

 }