JPEG

Generated by Doxygen 1.6.3

Thu Sep 2 15:59:48 2010

Contents

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 3

File Index

3.1 File List

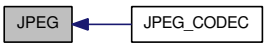
Here is a list of all files with brief descriptions:

Chapter 4

Module Documentation

4.1 JPEG

Collaboration diagram for JPEG:



Modules

- JPEG_CODEC

4.2 JPEG_CODEC

Collaboration diagram for JPEG_CODEC:

Data Structures

- struct DCT_data
  DCT FF structure definition.
- struct JOB_msg
  message structure definition.
- struct JOB_msgQueue
  messageQueue structure definition.

Defines

- #define DCT_8_X_8_BLOCK (8)
  dct macroblock 8∗8
- #define DCT_QUALITY (1)
  dct quality
- #define DCT_FIRST_PIX_QUALITY (1)
  first pixel in the 8∗8 quality
- #define DCT_1__SQRT2 (0.70710678)
  coefficients definition (dct matrix)
- #define DCT_PI__16 (0.98078528)
  second coefficient
- #define DCT_2PI__16 (0.92387953)
  third coefficient
- #define DCT_3PI__16 (0.83146961)
  forth coefficient
- #define DCT_4PI__16 (0.70710678)
  fifth coefficient
- #define DCT_5PI__16 (0.55557023)
  sixth coefficient
- #define DCT_6PI__16 (0.38268343)

seventh coefficient

- #define DCT_7PI__16 (0.19509032)
  eigth coefficient
- #define DCT_FFDCT_PRIORITY (25)
  dct task priority
- #define DCT_IDCT_PRIORITY (25)
  idct task priority
- #define ERROR_1 (-1)
  ERROR read / write definition.
- #define ERROR_2 (-2)
  ERROR_2 invalid file name (path).
- #define ERROR_3 (-3)
  ERROR_3 memory allocation error.
- #define ERROR_4 (-4)
  ERROR_4 reading file error.
- #define ERROR_5 (-5)
  ERROR_5 saving file error.
- #define ERROR_6 (-6)
  ERROR_6 null pointer error.
- #define OK (0)
  OK !!!
- #define ERROR_7 (-7)
  ERROR_7 null pointer (thread creation) error.
- #define ERROR_8 (-8)
  ERROR_8 thread creation error.
- #define ERROR_9 (-9)
  ERROR_9 msg queue init error.
- #define ERROR_10 (-10)
  ERROR_10 msg queue error : full msg Queue.
- #define JOB_MSGQUEUE_SIZE (20)
  msgQueue size
- #define DCT_JOB_ID (0x1)
  dct job ID

- #define DCT_JOB_COMPUTE (0x001)

  dct job ID

- #define JOB_EXIT (0x000)
- #define PGM_MAX_PATH_LENGTH (100)

  Max path length.

Typedefs

- typedef struct DCT_data DCT_data_ts

  DCT FF structure definition.

- typedef struct JOB_msg JOB_msg_ts

  message structure definition.

- typedef struct JOB_msgQueue JOB_msgQueue_ts

  messageQueue structure definition.

Functions

- void * DCT_forwardDct_f (void *params)

  flag for testing idct

- void * DCT_iDct_f (void *params)

  inverse dct.

- void DCT_transfomMatrixInit_f (float **dctMatrix)

  initialize the dct matrix.

- void DCT_quantumMatrixInit_f (int **quantumMatrix)

  initialize the quantum matrix DCT_8_X_8_BLOCK.

- void DCT_quantifyMatrix_f (int **quantifiedMatrix, float **inputMatrix, int **quantumMatrix)

  quantify Matrix DCT_8_X_8_BLOCK.

- void DCT_dequantifyMatrix_f (float **dequantifiedMatrix, int **inputMatrix, int **quantumMatrix)

  dequantify Matrix DCT_8_X_8_BLOCK.

- int JTOOLS_createJob (pthread_t *threadId, int threadPriority, void *jobFunction, void *threadData)

  create a job.

- int JTOOLS_exitJob (pthread_t *threadId)

  create a job.

- int JTOOLS_msgQueueInit (JOB_msgQueue_ts *msgQueue)

init a msg queue .

- int JTOOLS_msgQueuePush (JOB_msgQueue_ts *msgQueue, long msgId, long data1, long data2)
  push a message in the msg queue .
- int JTOOLS_msgQueueWait (JOB_msgQueue_ts *msgQueue, JOB_msg_ts *msg)
  push a message in the msg queue .
- int ** MTOOLS_matrixAllocInt_f (int nb_rows, int nb_cols)
  allocates 2D Matrix.
- void MTOOLS_matrixFreeInt_f (int **matrix)
  deallocates 2D Matrix.
- float ** MTOOLS_matrixAllocFloat_f (int nb_rows, int nb_cols)
  allocates 2D Matrix.
- void MTOOLS_matrixFreeFloat_f (float **matrix)
  deallocates 2D Matrix.
- void MTOOLS_matrixShowInt_f (int **matrix, int rows, int cols, unsigned int lineWidth)
  shows 2D Matrix.
- void MTOOLS_matrixShowFloat_f (float **matrix, int rows, int cols, unsigned int lineWidth)
  shows 2D Matrix.
- void MTOOLS_matrixTransposer (float **inputMatrix, float **outputMatrix, int rows, int cols)
  transpose Mtrix
- void MTOOLS_multiplyMatrix (float **inputMatrix_1, float **inputMatrix_2, float **outputMatrix, int rows, int cols)
  transpose Matrix
- void MTOOLS_multiplyMatrixZeros (float **inputMatrix_1, float **inputMatrix_2, float **outputMatrix, int rows, int cols, int zeroRow, int zeroCol)
  transpose Matrix
- void MTOOLS_matrixConvInt2Float (int **matrixInt, float **matrixFloat, int rows, int cols)
  converts Matrix from int to float
- void MTOOLS_matrixConvFloat2Int (float **matrixFloat, int **matrixInt, int rows, int cols)
  converts Matrix from float to int
- void MTOOLS_matrixCopyInt1 (int **inputMatrix, int **outputMatrix, int rows, int cols, int zeroRowsInput, int zeroColsOutput)
  transpose Mtrix
- void MTOOLS_matrixCopyInt2 (int **inputMatrix, int **outputMatrix, int rows, int cols, int zeroRowsOutput, int zeroColsOutput)
  transpose Matrix

- void MTOOLS_zigzagMatrixCollector (int **inputMatrix, int matrixWidth, int *outputStream)
  zigzag matrix collector
- int PGM_readPicture (char *pathPicture, int *row, int *col)
  read pgm picture.
- int PGM_writePicture (char *pathPicture, int **pictureMatrix, int row, int col)
  write pgm picture.

Variables

- JOB_msgQueue_ts dctMsgQueue
- JOB_msgQueue_ts iDctMsgQueue

4.2.1 Define Documentation

4.2.1.1 #define DCT_1__SQRT2 (0.70710678)

first coefficient

4.2.1.2 #define DCT_2PI__16 (0.92387953)

4.2.1.3 #define DCT_3PI__16 (0.83146961)

4.2.1.4 #define DCT_4PI__16 (0.70710678)

4.2.1.5 #define DCT_5PI__16 (0.55557023)

4.2.1.6 #define DCT_6PI__16 (0.38268343)

4.2.1.7 #define DCT_7PI__16 (0.19509032)

4.2.1.8 #define DCT_8_X_8_BLOCK (8)

Referenced by main().

4.2.1.9 #define DCT_FFDCT_PRIORITY (25)

Referenced by main().

4.2.1.10 #define DCT_FIRST_PIX_QUALITY (1)

4.2.1.11 #define DCT_IDCT_PRIORITY (25)

Referenced by main().

4.2.1.12 #define DCT_JOB_COMPUTE (0x001)

Referenced by main().

4.2.1.13 #define DCT_JOB_ID (0x1)

4.2.1.14 #define DCT_PI__16 (0.98078528)

4.2.1.15 #define DCT_QUALITY (1)

4.2.1.16 #define ERROR_1 (-1)

ERROR_1 opening file error : file does not exist

Referenced by ecrire_image_pgm(), and lire_image_pgm().

4.2.1.17 #define ERROR_10 (-10)

Referenced by main().

4.2.1.18 #define ERROR_2 (-2)

Referenced by lire_image_pgm().

4.2.1.19 #define ERROR_3 (-3)

Referenced by lire_image_pgm().

4.2.1.20 #define ERROR_4 (-4)

Referenced by lire_image_pgm().

4.2.1.21 #define ERROR_5 (-5)

Referenced by ecrire_image_pgm().

4.2.1.22 #define ERROR_6 (-6)

Referenced by PGM_writePicture().

4.2.1.23 #define ERROR_7 (-7)

4.2.1.24 #define ERROR_8 (-8)

4.2.1.25 #define ERROR_9 (-9)

4.2.1.26 #define JOB_EXIT (0x000)

Referenced by main().

4.2.1.27 #define JOB_MSGQUEUE_SIZE (20)

4.2.1.28 #define OK (0)

Referenced by ecrire_image_pgm(), main(), and PGM_writePicture().

4.2.1.29 #define PGM_MAX_PATH_LENGTH (100)

Referenced by ecrire_image_pgm(), lire_image_pgm(), and main().

4.2.2 Typedef Documentation

4.2.2.1 typedef struct DCT_data DCT_data_ts

4.2.2.2 typedef struct JOB_msg JOB_msg_ts

4.2.2.3 typedef struct JOB_msgQueue JOB_msgQueue_ts

4.2.3 Function Documentation

4.2.3.1 void DCT_dequantifyMatrix_f (float ** *dequantifiedMatrix*, int ** *inputMatrix*, int ** *quantumMatrix*)

Parameters

int** dequantifiedMatrix : output matrix.

float** inputMatrix : input matrix.

int** quantumMatrix : quantum matrix.

Returns

void

Author

mohamet.jaafar@gmail.com

Date

4.2.3.2 void * DCT_forwardDct_f (void * *params*)

flag for testing block 8x8 processing

forward dct.

Parameters

void *params : argument to pass (DCT_data_ts).

Returns

dct job id

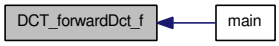
Author

mohamet.jaafar@gmail.com

Date

Referenced by main().

Here is the caller graph for this function:



4.2.3.3 void * DCT_iDct_f (void * *params*)

Parameters

void *params : argument to pass (DCT_data_ts).

Returns

dct job id

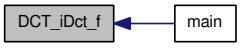
Author

mohamet.jaafar@gmail.com

Date

Referenced by main().

Here is the caller graph for this function:



4.2.3.4 void DCT_quantifyMatrix_f (int ** *quantifiedMatrix*, float ** *inputMatrix*, int ** *quantumMatrix*)

Parameters

int** quantifiedMatrix : output matrix.

float** inputMatrix : input matrix.

int** quantumMatrix : quantum matrix.

Returns

void

Author

mohamet.jaafar@gmail.com

Date

4.2.3.5 void DCT_quantumMatrixInit_f (int ** *quantumMatrix*)

Parameters

int **quantumMatrix : the quantum matrix.

Returns

Author

mohamet.jaafar@gmail.com

Date

4.2.3.6 void DCT_transfomMatrixInit_f (float ** *dctMatrix*)

Parameters

int **dctMatrix : the dct matrix.

Returns

Author

mohamet.jaafar@gmail.com

Date

4.2.3.7 int JTOOLS_createJob (pthread_t * *threadId*, int *threadPriority*, void * *jobFunction*, void * *threadData*)

Parameters

pthread_t *threadId : job thread id.

int threadPriority : job thread priority.

void *jobFunction : job fuction to execute.

void *threadData : job data.

Returns

error code

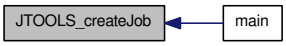
Author

mohamet.jaafar@gmail.com

Date

Referenced by main().

Here is the caller graph for this function:



4.2.3.8 int JTOOLS_exitJob (pthread_t * *threadId*)

Parameters

pthread_t *threadId : job thread id.

Returns

error code

Author

mohamet.jaafar@gmail.com

Date

Referenced by main().

Here is the caller graph for this function:

4.2.3.9 int JTOOLS_msgQueueInit (JOB_msgQueue_ts * *msgQueue*)

Parameters

JOB_msgQueue *msgQueue : message queue.

Returns

error code

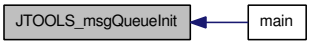
Author

mohamet.jaafar@gmail.com

Date

Referenced by main().

Here is the caller graph for this function:



4.2.3.10 int JTOOLS_msgQueuePush (JOB_msgQueue_ts * *msgQueue*, long *msgId*, long *data1*, long *data2*)

Parameters

JOB_msgQueue *msgQueue : message queue.

long msgId : message id.

long data1 : message 1st data.

long data2 : message 2nd data.

Returns

error code

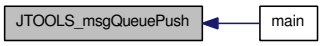
Author

mohamet.jaafar@gmail.com

Date

Referenced by main().

Here is the caller graph for this function:



4.2.3.11 int JTOOLS_msgQueueWait (JOB_msgQueue_ts * *msgQueue*, JOB_msg_ts * *msg*)

Parameters

JOB_msgQueue *msgQueue : message queue.

JOB_msg msg : message read.

Returns

error code

Author

mohamet.jaafar@gmail.com

Date

4.2.3.12 float** MTOOLS_matrixAllocFloat_f (int *nb_rows*, int *nb_cols*)

Parameters

int nb_rows : number of rows.

int nb_cols : number of colums.

Returns

address in memory of the allocated matrix

Author

sonia.zaibi@enit.rnu.edu

Date

4.2.3.13 int** MTOOLS_matrixAllocInt_f (int *nb_rows*, int *nb_cols*)

Parameters

int nb_rows : number of rows.

int nb_cols : number of colums.

Returns

address in memory of the allocated matrix

Author

sonia.zaibi@enit.rnu.edu

Date

Referenced by lire_image_pgm(), and main().

Here is the caller graph for this function:

4.2.3.14 void MTOOLS_matrixConvFloat2Int (float ** *matrixFloat*, int ** *matrixInt*, int *rows*, int *cols*)

Parameters

int* matrixFloat : input matrix.

float* matrixInt : output matrix.

int cols : number of columns.

int rows : matrix of rows.

Returns

void

Author

mohamet.jaafar@gmail.com

Date

4.2.3.15 void MTOOLS_matrixConvInt2Float (int ** *matrixInt*, float ** *matrixFloat*, int *rows*, int *cols*)

Parameters

int* matrixInt : input matrix.

float* matrixFloat : output matrix.

int cols : number of columns.

int rows : matrix of rows.

Returns

void

Author

mohamet.jaafar@gmail.com

Date

4.2.3.16 void MTOOLS_matrixCopyInt1 (int ** *inputMatrix*, int ** *outputMatrix*, int *rows*, int *cols*, int *zeroRowsInput*, int *zeroColsOutput*)

Parameters

int* inputMatrix : input matrix.

int* outputMatrix : output matrix.

int cols : number of columns.

int rows : matrix of rows.

int zeroCols : zero of columns(input).

int zeroRow : zero of rows(input).

Returns

void

Author

mohamet.jaafar@gmail.com

Date

4.2.3.17 void MTOOLS_matrixCopyInt2 (int ** *inputMatrix*, int ** *outputMatrix*, int *rows*, int *cols*, int *zeroRowsOutput*, int *zeroColsOutput*)

Parameters

int** inputMatrix : input matrix.

int** outputMatrix : output matrix.

int cols : number of columns.

int rows : matrix of rows.

int zeroCols : zero of columns(output).

int zeroRow : zero of rows(output).

Returns

void

Author

mohamet.jaafar@gmail.com

Date

4.2.3.18 void MTOOLS_matrixFreeFloat_f (float ** *matrix*)

Parameters

float** matrix : address in memory of the allocated matrix.

Returns

void

Author

sonia.zaibi@enit.rnu.edu

Date

4.2.3.19 void MTOOLS_matrixFreeInt_f (int ** *matrix*)

Parameters

int* matrix : address in memory of the allocated matrix.

Returns

void

Author

sonia.zaibi@enit.rnu.edu

Date

Referenced by main().

Here is the caller graph for this function:

4.2.3.20 void MTOOLS_matrixShowFloat_f (float ** *matrix*, int *rows*, int *cols*, unsigned int *lineWidth*)

Parameters

float* matrix : matrix to show.

int col : number of columns.

int row : matrix of rows.

unsigned int : the width of the line to show on the screen.

Returns

void

Author

mohamet.jaafar@gmail.com

Date

4.2.3.21 void MTOOLS_matrixShowInt_f (int ** *matrix*, int *rows*, int *cols*, unsigned int *lineWidth*)

Parameters

int* matrix : matrix to show.

int col : number of columns.

int row : matrix of rows.

unsigned int : the width of the line to show on the screen.

Returns

void

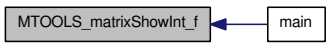
Author

mohamet.jaafar@gmail.com

Date

Referenced by main().

Here is the caller graph for this function:



4.2.3.22 void MTOOLS_matrixTransposer (float ** *inputMatrix*, float ** *outputMatrix*, int *rows*, int *cols*)

Parameters

float** inputMatrix : input matrix.

float** outputMatrix : output matrix.

int col : number of columns.

int row : matrix of rows.

Returns

void

Author

mohamet.jaafar@gmail.com

Date

4.2.3.23 void MTOOLS_multiplyMatrix (float ** *inputMatrix_1*, float ** *inputMatrix_2*, float ** *outputMatrix*, int *rows*, int *cols*)

Parameters

float** inputMatrix_1 : input matrix 1.

float** inputMatrix_2 : input matrix 2.

float** outputMatrix : output matrix.

int col : number of columns.

int row : matrix of rows.

Returns

void

Author

mohamet.jaafar@gmail.com

Date

4.2.3.24 void MTOOLS_multiplyMatrixZeros (float ∗∗ *inputMatrix_1*, float ∗∗ *inputMatrix_2*, float ∗∗ *outputMatrix*, int *rows*, int *cols*, int *zeroRow*, int *zeroCol*)

Parameters

float∗∗ inputMatrix_1 : input matrix 1.

float∗∗ inputMatrix_2 : input matrix 2.

float∗∗ outputMatrix : output matrix.

int col : number of columns.

int row : matrix of rows.

Returns

void

Author

mohamet.jaafar@gmail.com

Date

4.2.3.25 void MTOOLS_zigzagMatrixCollector (int ∗∗ *inputMatrix*, int *matrixWidth*, int ∗ *outputStream*)

Parameters

int∗∗ inputMatrix : input matrix.

int matrixWidth : matrix width.

int∗ outputStream : output tabular.

Returns

void

Author

mohamet.jaafar@gmail.com

Date

Referenced by main().

Here is the caller graph for this function:

4.2.3.26 int PGM_readPicture (char * *pathPicture*, int * *row*, int * *col*)

Parameters

char *∗pathPicture : path of the pgm picture to read.

int *∗row : output number of rows.

int *∗col : output number of columns.

Returns

pointer on the allocated matrix in the heap

Author

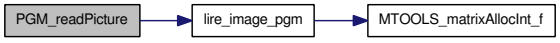
mohamet.jaafar@gmail.com

Date

References lire_image_pgm().

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:

4.2.3.27 int PGM_writePicture (char * *pathPicture*, int ** *pictureMatrix*, int *row*, int *col*)

Parameters

char *∗pathPicture : path of the pgm picture to write.

int *∗∗pictureMatrix : matrix holding pixel values.

int row : input number of rows.

int col : input number of columns.

Returns

error code

Author

mohamet.jaafar@gmail.com

Date

References ecrire_image_pgm(), ERROR_6, and OK.

Referenced by main().

Here is the call graph for this function:

Here is the caller graph for this function:

4.2.4 Variable Documentation

4.2.4.1 JOB_msgQueue_ts dctMsgQueue

4.2.4.2 JOB_msgQueue_ts iDctMsgQueue

Chapter 5

Data Structure Documentation

5.1 DCT_data Struct Reference

DCT FF structure definition.

`#include <dct_tools.h>`

Data Fields

- int ** inputPictureMatrix
- int ** outputPictureMatrix
- int col
- int row

5.1.1 Field Documentation

5.1.1.1 int DCT_data::col

5.1.1.2 int** DCT_data::inputPictureMatrix

5.1.1.3 int** DCT_data::outputPictureMatrix

5.1.1.4 int DCT_data::row

The documentation for this struct was generated from the following file:

- dct_tools.h

5.2 JOB_msg Struct Reference

message structure definition.

`#include <job_tools.h>`

Data Fields

- int msgId
- int data1
- int data2

5.2.1 Field Documentation

5.2.1.1 int JOB_msg::data1

1st message parameter

5.2.1.2 int JOB_msg::data2

2nd message parameter

5.2.1.3 int JOB_msg::msgId

Message ID

The documentation for this struct was generated from the following file:

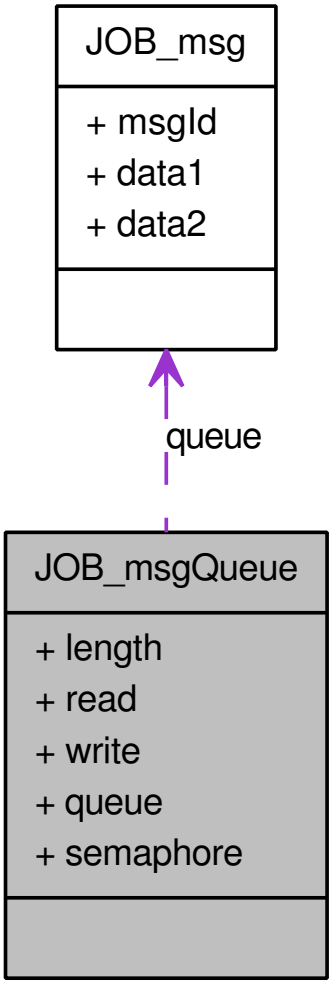
- job_tools.h

5.3 JOB_msgQueue Struct Reference

messageQueue structure definition.

`#include <job_tools.h>`

Collaboration diagram for JOB_msgQueue:



Data Fields

- int length
- int read
- int write
- JOB_msg_ts queue [JOB_MSGQUEUE_SIZE]
- sem_t semaphore

5.3.1 Field Documentation

5.3.1.1 int JOB_msgQueue::length

Queue length.

Referenced by main().

5.3.1.2 JOB_msg_ts JOB_msgQueue::queue[JOB_MSGQUEUE_SIZE]

Message queue.

5.3.1.3 int JOB_msgQueue::read

Read posision.

5.3.1.4 sem_t JOB_msgQueue::semaphore

Waiting semaphore.

5.3.1.5 int JOB_msgQueue::write

Write position.

The documentation for this struct was generated from the following file:

- job_tools.h

Chapter 6

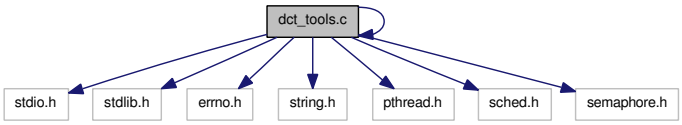
File Documentation

6.1 dct_tools.c File Reference

dct / idct

```
#include "dct_tools.h"
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <pthread.h>
#include <sched.h>
#include <semaphore.h>
```

Include dependency graph for dct_tools.c:



This graph shows which files directly or indirectly include this file:

Functions

- void * DCT_forwardDct_f (void *params)

  flag for testing idct

- void * DCT_iDct_f (void *params)

inverse dct.

- void DCT_transfomMatrixInit_f (float **dctMatrix)
  initialize the dct matrix.
- void DCT_quantumMatrixInit_f (int **quantumMatrix)
  initialize the quantum matrix DCT_8_X_8_BLOCK.
- void DCT_quantifyMatrix_f (int **quantifiedMatrix, float **inputMatrix, int **quantumMatrix)
  quantify Matrix DCT_8_X_8_BLOCK.
- void DCT_dequantifyMatrix_f (float **dequantifiedMatrix, int **inputMatrix, int **quantumMatrix)
  dequantify Matrix DCT_8_X_8_BLOCK.

Variables

- JOB_msgQueue_ts dctMsgQueue
- JOB_msgQueue_ts iDctMsgQueue

6.1.1 Detailed Description

Author

mohamet.jaafar@gmail.com

Version

0.8

Date

first release
modification mohamet.jaafar@gmail.com 2008

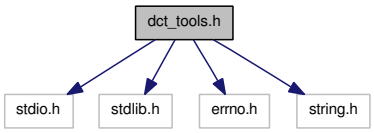
This file describe public PGM Picture reader / writer structure and functions.

6.2 dct_tools.h File Reference

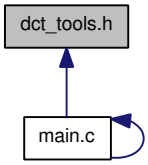
dct / idct

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
```

Include dependency graph for dct_tools.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct DCT_data
  DCT FF structure definition.

Defines

- #define DCT_8_X_8_BLOCK (8)
  *dct macroblock 8*8*
- #define DCT_QUALITY (1)
  dct quality
- #define DCT_FIRST_PIX_QUALITY (1)
  *first pixel in the 8*8 quality*
- #define DCT_1__SQRT2 (0.70710678)
  coefficients definition (dct matrix)
- #define DCT_PI__16 (0.98078528)
  second coefficient

- #define DCT_2PI__16 (0.92387953)
  third coefficient
- #define DCT_3PI__16 (0.83146961)
  forth coefficient
- #define DCT_4PI__16 (0.70710678)
  fifth coefficient
- #define DCT_5PI__16 (0.55557023)
  sixth coefficient
- #define DCT_6PI__16 (0.38268343)
  seventh coefficient
- #define DCT_7PI__16 (0.19509032)
  eigth coefficient
- #define DCT_FFDCT_PRIORITY (25)
  dct task priority
- #define DCT_IDCT_PRIORITY (25)
  idct task priority

Typedefs

- typedef struct DCT_data DCT_data_ts
  DCT FF structure definition.

Functions

- void * DCT_forwardDct_f (void *params)
  flag for testing idct
- void DCT_quantumMatrixInit_f (int **quantumMatrix)
  initialize the quantum matrix DCT_8_X_8_BLOCK.
- void DCT_transfomMatrixInit_f (float **dctMatrix)
  initialize the dct matrix.
- void DCT_quantifyMatrix_f (int **quantifiedMatrix, float **inputMatrix, int **quantumMatrix)
  quantify Matrix DCT_8_X_8_BLOCK.
- void DCT_dequantifyMatrix_f (float **dequantifiedMatrix, int **inputMatrix, int **quantumMatrix)
  dequantify Matrix DCT_8_X_8_BLOCK.

- void * DCT_iDct_f (void *params)
  inverse dct.

6.2.1 Detailed Description

Author

mohamet.jaafar@gmail.com

Version

0.8

Date

first release
modification mohamet.jaafar@gmail.com 2008

This file describe public DCT /IDCT structure and functions.

---------- Copyright (C) 2004 Mohamed JAAFAR

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

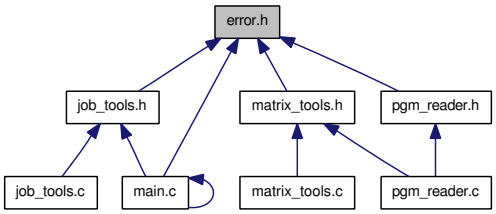
This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA 02110-1301, USA.

6.3 error.h File Reference

error type

This graph shows which files directly or indirectly include this file:



Defines

- #define ERROR_1 (-1)

  ERROR read / write definition.

- #define ERROR_2 (-2)

  ERROR_2 invalid file name (path).

- #define ERROR_3 (-3)

  ERROR_3 memory allocation error.

- #define ERROR_4 (-4)

  ERROR_4 reading file error.

- #define ERROR_5 (-5)

  ERROR_5 saving file error.

- #define ERROR_6 (-6)

  ERROR_6 null pointer error.

- #define OK (0)

  OK !!!

- #define ERROR_7 (-7)

  ERROR_7 null pointer (thread creation) error.

- #define ERROR_8 (-8)

  ERROR_8 thread creation error.

- #define ERROR_9 (-9)

  ERROR_9 msg queue init error.

- #define ERROR_10 (-10)

  ERROR_10 msg queue error : full msg Queue.

6.3.1 Detailed Description

Author

mohamet.jaafar@gmail.com

Version

0.8

Date

first release
modification mohamed.jaafar@gmail.com 2008

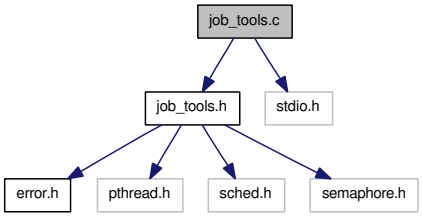
This file describe public error code structure and functions.

6.4 job_tools.c File Reference

job tools (multithreading)

```
#include "job_tools.h"
#include <stdio.h>
```

Include dependency graph for job_tools.c:



Functions

- int JTOOLS_createJob (pthread_t *threadId, int threadPriority, void *jobFunction, void *threadData)

  create a job.

- int JTOOLS_exitJob (pthread_t *threadId)

  create a job.

- int JTOOLS_msgQueueInit (JOB_msgQueue_ts *msgQueue)

  init a msg queue .

- int JTOOLS_msgQueuePush (JOB_msgQueue_ts *msgQueue, long msgId, long data1, long data2)

  push a message in the msg queue .

- int JTOOLS_msgQueueWait (JOB_msgQueue_ts *msgQueue, JOB_msg_ts *msg)

  push a message in the msg queue .

6.4.1 Detailed Description

Author

mohamet.jaafar@gmail.com

Version

0.8

Date

first release
modification mohamet.jaafar@gmail.com 2008

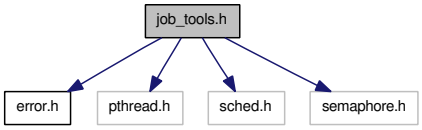
This file describe public multithreading approche structure and functions.

6.5 job_tools.h File Reference

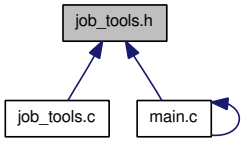
job tools (multithreading)

```
#include "error.h"
#include <pthread.h>
#include <sched.h>
#include <semaphore.h>
```

Include dependency graph for job_tools.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct JOB_msg

  message structure definition.

- struct JOB_msgQueue

  messageQueue structure definition.

Defines

- #define JOB_MSGQUEUE_SIZE (20)

  msgQueue size

- #define DCT_JOB_ID (0x1)

  dct job ID

- #define DCT_JOB_COMPUTE (0x001)

  dct job ID

- #define JOB_EXIT (0x000)

Typedefs

- typedef struct JOB_msg JOB_msg_ts
  message structure definition.
- typedef struct JOB_msgQueue JOB_msgQueue_ts
  messageQueue structure definition.

Functions

- int JTOOLS_createJob (pthread_t *threadId, int threadPriority, void *jobFunction, void *threadData)
  create a job.
- int JTOOLS_exitJob (pthread_t *threadId)
  create a job.
- int JTOOLS_msgQueueInit (JOB_msgQueue_ts *msgQueue)
  init a msg queue .
- int JTOOLS_msgQueuePush (JOB_msgQueue_ts *msgQueue, long msgId, long data1, long data2)
  push a message in the msg queue .
- int JTOOLS_msgQueueWait (JOB_msgQueue_ts *msgQueue, JOB_msg_ts *msg)
  push a message in the msg queue .

6.5.1 Detailed Description

Author

mohamet.jaafar@gmail.com

Version

0.8

Date

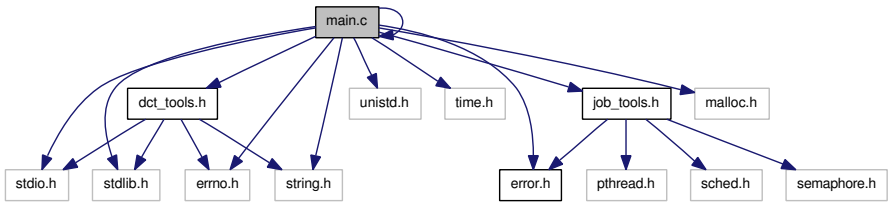
first release
modification mohamet.jaafar@gmail.com 2008

This file describe public multithreading approche structure and functions.

6.6 main.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <time.h>
#include "error.h"
#include "pgm_reader.h"
#include <malloc.h>
#include "job_tools.h"
#include "dct_tools.h"
```

Include dependency graph for main.c:



This graph shows which files directly or indirectly include this file:

Functions

- int main (int argc, char **argv)

Variables

- JOB_msgQueue_ts dctMsgQueue
- JOB_msgQueue_ts iDctMsgQueue

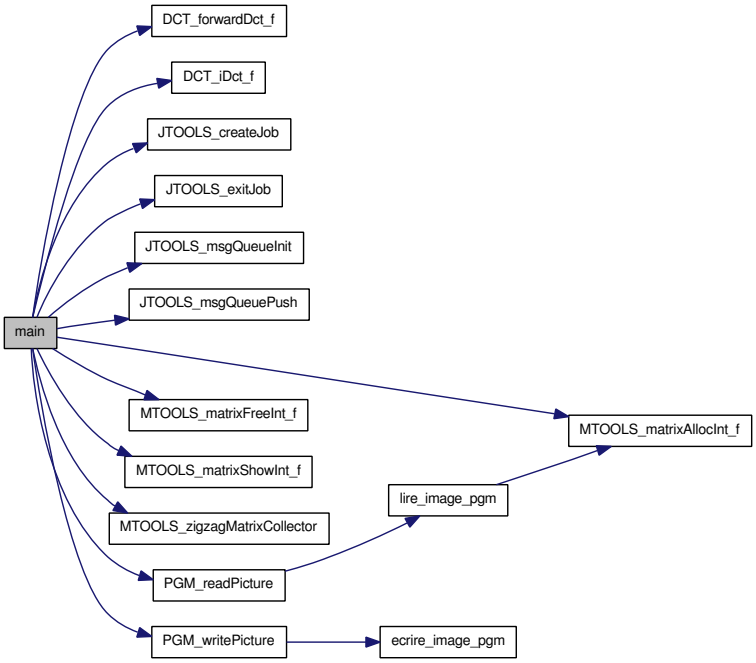
6.6.1 Function Documentation

6.6.1.1 int main (int *argc*, char ** *argv*)

References DCT_8_X_8_BLOCK, DCT_FFDCT_PRIORITY, DCT_forwardDct_f(), DCT_iDct_f(), DCT_IDCT_PRIORITY, DCT_JOB_COMPUTE, ERROR_10, JOB_EXIT, JTOOLS_createJob(), JTOOLS_exitJob(), JTOOLS_msgQueueInit(), JTOOLS_msgQueuePush(), JOB_msgQueue::length,

MTOOLS_matrixAllocInt_f(), MTOOLS_matrixFreeInt_f(), MTOOLS_matrixShowInt_f(), MTOOLS_-zigzagMatrixCollector(), OK, PGM_MAX_PATH_LENGTH, PGM_readPicture(), and PGM_-writePicture().

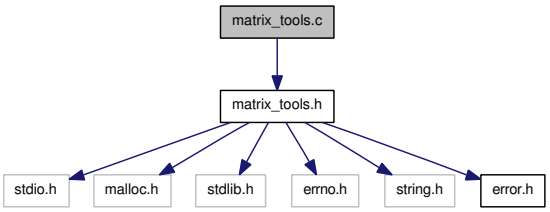
Here is the call graph for this function:



6.7 matrix_tools.c File Reference

```
#include "matrix_tools.h"
```

Include dependency graph for matrix_tools.c:



Functions

- int ** MTOOLS_matrixAllocInt_f (int nb_rows, int nb_cols)
  allocates 2D Matrix.
- void MTOOLS_matrixFreeInt_f (int **matrix)
  deallocates 2D Matrix.
- float ** MTOOLS_matrixAllocFloat_f (int nb_rows, int nb_cols)
  allocates 2D Matrix.
- void MTOOLS_matrixFreeFloat_f (float **matrix)
  deallocates 2D Matrix.
- void MTOOLS_matrixShowInt_f (int **matrix, int rows, int cols, unsigned int lineWidth)
  shows 2D Matrix.
- void MTOOLS_matrixShowFloat_f (float **matrix, int rows, int cols, unsigned int lineWidth)
  shows 2D Matrix.
- void MTOOLS_matrixTransposer (float **inputMatrix, float **outputMatrix, int rows, int cols)
  transpose Mtrix
- void MTOOLS_multiplyMatrix (float **inputMatrix_1, float **inputMatrix_2, float **outputMatrix, int rows, int cols)
  transpose Matrix
- void MTOOLS_multiplyMatrixZeros (float **inputMatrix_1, float **inputMatrix_2, float **outputMatrix, int rows, int cols, int zeroRow, int zeroCol)
  transpose Matrix
- void MTOOLS_matrixConvInt2Float (int **matrixInt, float **matrixFloat, int rows, int cols)
  converts Matrix from int to float
- void MTOOLS_matrixConvFloat2Int (float **matrixFloat, int **matrixInt, int rows, int cols)
  converts Matrix from float to int

- void MTOOLS_matrixCopyInt1 (int **inputMatrix, int **outputMatrix, int rows, int cols, int zeroRowsInput, int zeroColsInput)

  transpose Mtrix

- void MTOOLS_matrixCopyInt2 (int **inputMatrix, int **outputMatrix, int rows, int cols, int zeroRowsOutput, int zeroColsOutput)

  transpose Matrix

- void MTOOLS_zigzagMatrixCollector (int **inputMatrix, int matrixWidth, int *outputStream)

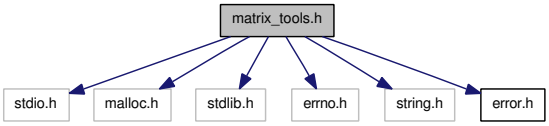
  zigzag matrix collector

6.8 matrix_tools.h File Reference

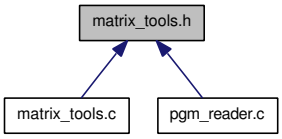
matrix tools

```
#include <stdio.h>
#include <malloc.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include "error.h"
```

Include dependency graph for matrix_tools.h:



This graph shows which files directly or indirectly include this file:



Functions

- int ** MTOOLS_matrixAllocInt_f (int nb_rows, int nb_cols)
  allocates 2D Matrix.
- void MTOOLS_matrixFreeInt_f (int **matrix)
  deallocates 2D Matrix.
- float ** MTOOLS_matrixAllocFloat_f (int nb_rows, int nb_cols)
  allocates 2D Matrix.
- void MTOOLS_matrixFreeFloat_f (float **matrix)
  deallocates 2D Matrix.
- void MTOOLS_matrixShowInt_f (int **matrix, int rows, int cols, unsigned int lineWidth)
  shows 2D Matrix.
- void MTOOLS_matrixShowFloat_f (float **matrix, int rows, int cols, unsigned int lineWidth)
  shows 2D Matrix.
- void MTOOLS_matrixTransposer (float **inputMatrix, float **outputMatrix, int rows, int cols)
  transpose Mtrix

- void MTOOLS_multiplyMatrix (float **inputMatrix_1, float **inputMatrix_2, float **outputMatrix, int rows, int cols)

  transpose Matrix

- void MTOOLS_multiplyMatrixZeros (float **inputMatrix_1, float **inputMatrix_2, float **outputMatrix, int rows, int cols, int zeroRow, int zeroCol)

  transpose Matrix

- void MTOOLS_matrixConvInt2Float (int **matrixInt, float **matrixFloat, int rows, int cols)

  converts Matrix from int to float

- void MTOOLS_matrixConvFloat2Int (float **matrixFloat, int **matrixInt, int rows, int cols)

  converts Matrix from float to int

- void MTOOLS_matrixCopyInt1 (int **inputMatrix, int **outputMatrix, int rows, int cols, int zeroRowsInput, int zeroColsOutput)

  transpose Mtrix

- void MTOOLS_matrixCopyInt2 (int **inputMatrix, int **outputMatrix, int rows, int cols, int zeroRowsOutput, int zeroColsOutput)

  transpose Matrix

- void MTOOLS_zigzagMatrixCollector (int **inputMatrix, int matrixWidth, int *outputStream)

  zigzag matrix collector

6.8.1 Detailed Description

Author

sonia.zaibi@enit.rnu.edu

Version

0.8

Date

first release
modification mohamet.jaafar@gmail.com 2008

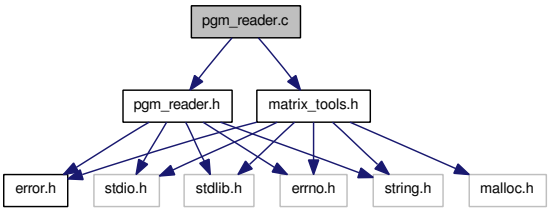
This file describe public PGM Picture reader / writer structure and functions.

6.9 pgm_reader.c File Reference

pgm picture read write file

```
#include "pgm_reader.h"
#include "matrix_tools.h"
```

Include dependency graph for pgm_reader.c:



Functions

- int lire_image_pgm (char *nom_brut, int *nb_col, int *nb_lin)
  read pgm picture.
- int ecrire_image_pgm (char *nom, int **ima_in, int taillex, int tailley)
  write pgm picture.
- int PGM_readPicture (char *pathPicture, int *row, int *col)
  read pgm picture.
- int PGM_writePicture (char *pathPicture, int **pictureMatrix, int row, int col)
  write pgm picture.

6.9.1 Detailed Description

Author

sonia.zaibi@enit.rnu.edu

Version

0.8

Date

first release
modification mohamet.jaafar@gmail.com 2008

This file describe an pgm picture read / write.

---------- Copyright (C) 2004 Mohamed JAAFAR

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA 02110-1301, USA.

6.9.2 Function Documentation

6.9.2.1 int ecrire_image_pgm (char * *nom*, int ** *ima_in*, int *taillex*, int *tailley*)

Parameters

char *nom : path of the pgm picture to write.

int **ima_originale : input matrix.

int nb_col : input number of columns.

int nb_lin : input number of rows.

Returns

error code

Author

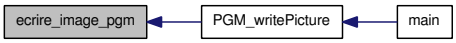
sonia.zraibi@rnu.edu

Date

References ERROR_1, ERROR_5, OK, and PGM_MAX_PATH_LENGTH.

Referenced by PGM_writePicture().

Here is the caller graph for this function:



6.9.2.2 int lire_image_pgm (char * *nom_brut*, int * *nb_col*, int * *nb_lin*)

Parameters

char *nom_brut : path of the pgm picture to read.

int **ima_originale : output matrix.

int *nb_col : output number of columns.

int *nb_lin : output number of rows.

Returns

error code

Author

sonia.zraibi@rnu.edu

Date

try to get filename from user

opening the file

the PGM file is already opened

reading the pgm file header

on trouve une nlle ligne de commentaires

lecture des dimensions

lecture des points

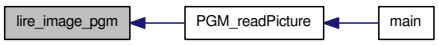
allocations

References ERROR_1, ERROR_2, ERROR_3, ERROR_4, MTOOLS_matrixAllocInt_f(), and PGM_-MAX_PATH_LENGTH.

Referenced by PGM_readPicture().

Here is the call graph for this function:

Here is the caller graph for this function:

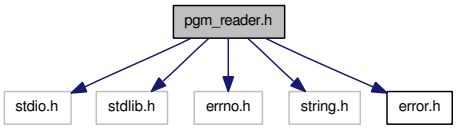


6.10 pgm_reader.h File Reference

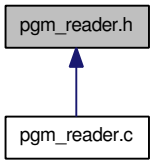
pgm picture reader / writer

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include "error.h"
```

Include dependency graph for pgm_reader.h:



This graph shows which files directly or indirectly include this file:



Defines

- #define PGM_MAX_PATH_LENGTH (100)
  Max path length.

Functions

- int PGM_readPicture (char *pathPicture, int *row, int *col)
  read pgm picture.
- int PGM_writePicture (char *pathPicture, int **pictureMatrix, int row, int col)
  write pgm picture.

6.10.1 Detailed Description

Author

sonia.zaibi@enit.rnu.edu

Version

0.8

Date

first release
modification mohamet.jaafar@gmail.com 2008

This file describe public PGM Picture reader / writer structure and functions.

Index