



**TEXAS
INSTRUMENTS**
TMS34010
Math/Graphics Function Library
Reference Card

<code>double acos(x)</code> double x;
<code>void add_text_space(n)</code> int n;
<code>double asin(x)</code> double x;
<code>double atan(x)</code> double x;
<code>double atan2(u,v)</code> double u,v;
<code>void bit_expand(srcbits, srcpitch, w, h, xleft, ytop)</code> short srcbits[]; long srcpitch; int w, h, xleft, ytop;
<code>void bound_fill(x, y, buffer, size, b_color)</code> int x, y, size; char buffer[]; unsigned long b_color;
<code>void bound_patnfill(x, y, buffer, size, b_color)</code> int x, y, size; char buffer[]; unsigned long b_color;
<code>double ceil(x)</code> double x;
<code>int char_high()</code>
<code>int char_wide_max()</code>
<code>void clear_screen(pixval)</code> long pixval;
<code>int close_vuport(index)</code> int index;
<code>void color_blend(px1val, y1, y2, red1, grn1, blu1, red2, grn2, blu2)</code> int px1val, y1, y2; int red1, grn1, blu1; int red2, grn2, blu2;
<code>typedef long FIX</code> <code>void copy_matrix(matrixin, matrixout)</code> FIX matrixin[16]; FIX matrixout[16];
<code>void copy_vertex(n, vertexin, vertexout)</code> typedef long FIX; int n; FIX vertexin[], vertexout[];
<code>int copy_vuport(index1, index2)</code> int index1, index2;
<code>double cos(x)</code> double x;
<code>double cosh(x)</code> double x;
<code>double cotan(x)</code> double x;
<code>int cpw(x, y)</code> int x, y;

<code>void delay(n)</code> int n;
<code>int draw_char(x, y, c)</code> int x, y; char c;
<code>void draw_line(x1, y1, x2, y2)</code> int x1, y1, x2, y2;
<code>void draw_oval(w, h, xleft, ytop)</code> int w, h, xleft, ytop;
<code>void draw_ovalarc(w, h, xleft, ytop, theta, arc)</code> int w, h, xleft, ytop; int theta, arc;
<code>void draw_piearc(w, h, xleft, ytop, theta, arc)</code> int w, h, xleft, ytop; int theta, arc;
<code>void draw_point(x, y)</code> int x, y;
<code>void draw_polyline(n, linelist, ptlist)</code> int n; short linelist[], ptlist[];
<code>void draw_rect(w, h, xleft, ytop)</code> int w, h, xleft, ytop;
<code>int draw_string(x, y, s)</code> int x, y; char *s;
<code>double exp(x)</code> double x;
<code>double fabs(x)</code> double x;
<code>int fill_convex(n, edgelist, ptlist)</code> int n; short edgelist[], ptlist[];
<code>void fill_oval(w, h, xleft, ytop)</code> int w, h, xleft, ytop;
<code>void fill_piearc(w, h, xleft, ytop, theta, arc)</code> int w, h, xleft, ytop; int theta, arc;
<code>void fill_polygon(n, linelist, ptlist)</code> int n; short linelist[], ptlist[];
<code>void fill_rect(w, h, xleft, ytop)</code> int w, h, xleft, ytop;
<code>float *fix_to_float(n, in_array, out_array)</code> typedef long FIX; int n; FIX in_array[]; float out_array[];
<code>long *fix_to_long(n, in_array, out_array)</code> typedef long FIX; int n; FIX in_array[]; long out_array[];
<code>short *fix_to_short(n, in_array, out_array)</code> typedef long FIX; int n; FIX in_array[]; short out_array[];
<code>global FIX2FL</code>
<code>global FL2FIX</code>
<code>global FL_ADD</code>
<code>global FL_COS</code>
<code>global FL_MULT</code>

<code>global FL_SIN</code>
<code>FIX *float_to_fix(n, in_array, out_array)</code> typedef long FIX; int n; float in_array[]; FIX out_array[];
<code>double floor(x)</code> double x;
<code>double fmod(x, y)</code> double x, y;
<code>void frame_oval(w, h, xleft, ytop, dx, dy)</code> int w, h, xleft, ytop; int dx, dy;
<code>void frame_rect(w, h, xleft, ytop, dx, dy)</code> int w, h, xleft, ytop; int dx, dy;
<code>double frexp(value, exp)</code> double value; int *exp;
<code>void getall_palet(palet_array, reg_mask, y)</code> short palet_array[16]; int reg_mask, y;
<code>int get_ascent()</code>
<code>int get_descent()</code>
<code>int get_first_ch()</code>
<code>int get_font_max()</code>
<code>int get_last_ch()</code>
<code>int get_leading()</code>
<code>int get_patn_max()</code>
<code>int get_pixel(x, y)</code> int x, y;
<code>long get_pmask()</code>
<code>long get_ppop()</code>
<code>int get_psize()</code>
<code>void get_rect(w, h, xleft, ytop, darray, dpitch)</code> int w, h, xleft, ytop; short darray[] long dpitch;
<code>int get_transp()</code>
<code>int get_vuport_max()</code>
<code>int get_width(s)</code> char *s;
<code>void init_grafix()</code>
<code>void init_matrix(matrix)</code> typedef long FIX; FIX matrix[16];
<code>void init_palet()</code>
<code>void init_screen()</code>
<code>void init_text()</code>
<code>int init_video(monitor_val)</code> int monitor_val;
<code>void init_vuport()</code>
<code>int install_font(index, fontname)</code> int index; FONT *fontname;
<code>int install_patn(index, pattern)</code> int index; short pattern[16];

double ldexp (value, exp) double value; int exp;
char *lib_id ()
int lmo (n) long n;
double log (x) double x;
double log10 (x) double x;
FIX *long_to_fix (n, in_array, out_array) typedef long FIX; int n; long in_array[]; FIX out_array[];
double modf (value, exp) double value; int *exp;
void move_pixel (xs, ys, xd, yd) int xs, ys, xd, yd;
void move_rect (w, h, xs, ys, xd, yd) int w, h; int xs, ys, xd, yd;
void move_vuport (xleft, ytop) int xleft, ytop;
void new_screen (pixel, palet) long pixel; short palet[16];
int open_vuport ()
int patnfill_convex (n, edgelist, ptlist) int n; short edgelist[], ptlist[];
void patnfill_oval (w, h, xleft, ytop) int w, h, xleft, ytop;
void patnfill_piearc (w, h, xleft, ytop, theta, arc) int w, h, xleft, ytop; int theta, arc;
void patnfill_polygon (n, linelist, ptlist) int n; short linelist[], ptlist[];
void patnfill_rect (w, h, xleft, ytop) int w, h, xleft, ytop;
void patnframe_oval (w, h, xleft, ytop, dx, dy) int w, h, xleft, ytop; int dx, dy;
void patnframe_rect (w, h, xleft, ytop, dx, dy) int w, h, xleft, ytop; int dx, dy;
void patnpen_line (x1, y1, x2, y2) int x1, y1, x2, y2;
void patnpen_ovalarc (w, h, xleft, ytop, theta, arc) int w, h, xleft, ytop; int theta, arc;
void patnpen_piearc (w, h, xleft, ytop, theta, arc) int w, h, xleft, ytop; int theta, arc;
void patnpen_point (x, y) int x, y;
void patnpen_polyline (n, linelist, ptlist) int n; short linelist[], ptlist[];
int peek (address) long address;

long peek_breg (breg) int breg;
void pen_line (x1, y1, x2, y2) int x1, y1, x2, y2;
void pen_ovalarc (w, h, xleft, ytop, theta, arc) int w, h, xleft, ytop; int theta, arc;
void pen_piearc (w, h, xleft, ytop, theta, arc) int w, h, xleft, ytop; int theta, arc;
void pen_point (x, y) int x, y;
void pen_polyline (n, linelist, ptlist) int n; short linelist[], ptlist[];
void perspec (n, vertlist, ptlist, xview, yview, zview) typedef long FIX; FIX vertlist[]; short ptlist[]; int n, xview, yview, zview;
void poke (address, value) long address; int value;
void poke_breg (breg, value) long breg; int value;
double pow (x, y) double x, y;
void put_pixel (val, x, y) int val, x, y;
void put_rect (sarray, spitch, w, h, xleft, ytop) short sarray[]; long spitch; int w, h, xleft, ytop;
long rep_pixel (val) int val;
int rmo (n) long n;
void rotate (matrix, angle) typedef long FIX; FIX matrix[16], angle[3];
void run_decode (xleft, ytop, image) int xleft, ytop; short image[];
int run_encode (w, h, xleft, ytop, image, maxbytes) int w, h, xleft, ytop, maxbytes; short image[];
void scale (matrix, factor) typedef long FIX; FIX matrix[16], factor[3];
void seed_fill (xseed, yseed, buffer, maxbytes) int xseed, yseed, maxbytes; char buffer[];
void seed_patnfill (xseed, yseed, buffer, maxbytes) int xseed, yseed, maxbytes; char buffer[];
int select_font (index) int index;
int select_patn (index) int index;
int select_vuport (index) int index;

void setall_palet (palet, reg_mask, n, y) short palet[16]; int reg_mask, n, y;
void set_cliprect (w, h, xleft, ytop) int w, h, xleft, ytop;
void set_color0 (pixel_val) long pixel_val;
void set_color1 (pixel_val) long pixel_val;
void set_origin (x0, y0) int x0, y0;
void set_palet (reg, red, grn, blu) int reg, red, grn, blu;
void set_pensize (w, h) int w, h;
void set_pmask (mask) long pmask;
void set_ppop (ppop_code) int ppop_code;
FIX *short_to_fix (n, in_array, out_array) typedef long FIX; int n; short in_array[]; FIX out_array[];
double sin (x) double x;
double sinh (x) double x;
int size_vuport (w, h) int w, h;
double sqrt (x) double x;
long styled_line (x1, y1, x2, y2, style, mode) int x1, y1, x2, y2, mode; long style;
double tan (x) double x;
double tanh (x) double x;
void transform (matrix, n, verts) typedef long FIX; FIX matrix[16], verts[]; int n;
void translate (matrix, disp) typedef long FIX; FIX matrix[16], disp[3];
void transp_off ()
void transp_on ()
void vertex_to_point (n, verts, ptlist) int n; FIX verts[]; short ptlist[];
void wait_scan (line) int line;
long xytoaddr (x, y) int x, y;
void zoom_rect (ws, hs, xs, ys, wd, hd, xd, yd, linebuf) int ws, hs, xs, ys; int wd, hd, xd, yd; short linebuf[];