Invert (65535, 0);

Invert (0, 65535);

| for (y = 0; y < 240; y++){  for (x = 0; x < 320; x++){  pixel\_ptr = FPGA\_ONCHIP\_BASE | (y << 10) | (x << 1);  temp = \*(short \*) pixel\_ptr;  R = (temp >> 6) & 0xF800  G = (temp >> 5) & 0x7E0  B = temp & 0x1F;  sum = R + G + B;  if (sum >= 64){  \*(short \*) pixel\_ptr = 65535;  }  else{  \*(short \*) pixel\_ptr = 0;  }    }  } |
| --- |

<https://abhijitnathwani.github.io/blog/2018/01/07/Black-and-White-Image-using-C>

<https://github.com/abhijitnathwani/image-processing/blob/master/black_white.c>

| int x2; Int pixel\_ptr; short t1; for(y = 0; y <= 240; y++){  for(x = 0; x <= 320; x++){  pixel\_ptr = FPGA\_ONCHIP\_BASE | (y << 10) | (x << 1);  t1= \*(short \*) pixel\_ptr;  x2 = 320 - x;  pixel\_ptr = FPGA\_ONCHIP\_BASE | (y << 10) | (x << 1);  \*(short \*) pixel\_ptr = t1; } } |
| --- |

<https://www.geeksforgeeks.org/pass-2d-array-parameter-c/>

void thisfunction(type \*name)

| #include <time.h> #include <stdio.h> #include <stdlib.h> #include <unistd.h> #define KEY\_BASE 0xFF200050 #define VIDEO\_IN\_BASE 0xFF203060 #define FPGA\_ONCHIP\_BASE 0xC8000000   /\* This program demonstrates the use of the D5M camera with the DE1-SoC Board  \* It performs the following:   \* 1. Capture one frame of video when any key is pressed.  \* 2. Display the captured frame when any key is pressed.  \*/ /\* Note: Set the switches SW1 and SW2 to high and rest of the switches to low for correct exposure timing while compiling and the loading the program in the Altera Monitor program. \*/  void printtext(char \*text\_ptr, int offset){  while ( \*(text\_ptr)){  \*((char\*) 0xC9000000 + offset) = \*(text\_ptr);  ++text\_ptr;  ++offset;  }  }  // printtext(&s,  int main(void) {  // Naming Variables  volatile int \* KEY\_ptr = (int \*) KEY\_BASE;  volatile int \* Video\_In\_DMA\_ptr = (int \*) VIDEO\_IN\_BASE;  volatile short \* Video\_Mem\_ptr = (short \*) FPGA\_ONCHIP\_BASE;   int x, y;  int count = 0;  char s[64];  int offset = (40 << 5) + 10;  char \*text\_ptr;  char \*cha;  setenv("TZ", "EST5EDT", 1); // Sets Timezone to EST    /\* Define temporary variables \*/   \*(Video\_In\_DMA\_ptr + 3) = 0x4; // Enable the video   while (1)  {  while (1)  {  if (\*KEY\_ptr != 0) // check if any KEY was pressed  {  \*(Video\_In\_DMA\_ptr + 3) = 0x0; // Disable the video to capture one frame  while (\*KEY\_ptr != 0); // wait for pushbutton KEY release  time\_t t = time(NULL);  struct tm \*tm = localtime(&t);  strftime(s, sizeof(s), "%c", tm);  break;  }  }  count++; // Keeps track of number of images captured   /\* Display the time in EST and UTC \*/  printf ("Count: %i, Time: %s", count, ctime(&rawtime));    \*text\_ptr = s;  printtext(&text\_ptr, offset);    cha= s;  snprintf(cha, 12, " Count %d", count);  text\_ptr = s;      while ( \*(text\_ptr)){  location = 0xC9000000;  volatile char \* address = (char\*) location;  \*(address+offset) = \*(text\_ptr);  ++text\_ptr;  ++offset;  }        for (y = 0; y < 240; y++) {  for (x = 0; x < 320; x++) {  short temp2 = \*(Video\_Mem\_ptr + (y << 9) + x);  \*(Video\_Mem\_ptr + (y << 9) + x) = temp2;  }  }    while (1)  {  if (\*KEY\_ptr != 0) // check if any KEY was pressed  {  \*(Video\_In\_DMA\_ptr + 3) = 0x4; // Enable the video in preparation for next image capture  while (\*KEY\_ptr != 0); // wait for pushbutton KEY release  break;  }  }    } } |
| --- |
|  |
|  |

<https://stackoverflow.com/questions/10279718/append-char-to-string-in-c>

<https://www.tutorialspoint.com/c_standard_library/c_function_strftime.htm>

<https://www.educative.io/edpresso/what-is-the-snprintf-function-in-c>

D5\_computer: onchip char buffer

<https://homepages.inf.ed.ac.uk/rbf/BOOKS/PHILLIPS/cips2ed.pdf>

<https://iq.opengenus.org/character-in-c/>

Source: <https://people.ece.cornell.edu/land/courses/ece5760/DE1_SOC/DE1_soc_computer/VGA_mandelbrot/mandelbrot_video_2.c>

C:/Users/michaelchan/Downloads/LAB2/Lab2\_ECE332\_2021/capture\_image/capture\_image.c:56:26: error: 'address' undeclared (first use in this function)

C:/Users/michaelchan/Downloads/LAB2/Lab2\_ECE332\_2021/capture\_image/capture\_image.c:56:26: note: each undeclared identifier is reported only once for each function it appears in

C:/Users/michaelchan/Downloads/LAB2/Lab2\_ECE332\_2021/capture\_image/capture\_image.c:52:17: warning: variable 'location' set but not used [-Wunused-but-set-variable]

Compilation stopped.

| #include <time.h> #include <stdio.h> #include <stdlib.h> #include <unistd.h> #define KEY\_BASE 0xFF200050 #define VIDEO\_IN\_BASE 0xFF203060 #define FPGA\_ONCHIP\_BASE 0xC8000000   /\* This program demonstrates the use of the D5M camera with the DE1-SoC Board  \* It performs the following:   \* 1. Capture one frame of video when any key is pressed.  \* 2. Display the captured frame when any key is pressed.  \*/ /\* Note: Set the switches SW1 and SW2 to high and rest of the switches to low for correct exposure timing while compiling and the loading the program in the Altera Monitor program. \*/ int main(void) {  volatile int \* KEY\_ptr = (int \*) KEY\_BASE;  volatile int \* Video\_In\_DMA\_ptr = (int \*) VIDEO\_IN\_BASE;  volatile short \* Video\_Mem\_ptr = (short \*) FPGA\_ONCHIP\_BASE;   int x, y;  int count = 0;  char s[64];  setenv("TZ", "EST5EDT", 1); // Sets Timezone to EST    /\* Define temporary variables \*/   \*(Video\_In\_DMA\_ptr + 3) = 0x4; // Enable the video   while (1)  {  while (1)  {  if (\*KEY\_ptr != 0) // check if any KEY was pressed  {  \*(Video\_In\_DMA\_ptr + 3) = 0x0; // Disable the video to capture one frame  while (\*KEY\_ptr != 0); // wait for pushbutton KEY release  time\_t t = time(NULL);  struct tm \*tm = localtime(&t);  strftime(s, sizeof(s), “%c”, tm);  break;  }  }  count++; // Keeps track of number of images captured  /\* Display the time in EST and UTC \*/  printf ("Count: %i, Time: %s", count, ctime(&rawtime));  int offset = (40 << 5) + 10;  int location;    char \*text\_ptr = s;  while ( \*(text\_ptr)){  location = 0xC9000000;  volatile char \* address = (char\*) location;  \*(address+offset) = \*(text\_ptr);  ++text\_ptr;  ++offset;  }  char \*cha = s;  snprintf(cha, 12, “ Count %d”, count);  text\_ptr = s;  offset = (40 << 5) + 10;  while ( \*(text\_ptr)){  location = 0xC9000000;  volatile char \* address = (char\*) location;  \*(address+offset) = \*(text\_ptr);  ++text\_ptr;  ++offset;  }    for (y = 0; y < 240; y++) {  for (x = 0; x < 320; x++) {  short temp2 = \*(Video\_Mem\_ptr + (y << 9) + x);  \*(Video\_Mem\_ptr + (y << 9) + x) = temp2;  }  }    while (1)  {  if (\*KEY\_ptr != 0) // check if any KEY was pressed  {  \*(Video\_In\_DMA\_ptr + 3) = 0x4; // Enable the video in preparation for next image capture  while (\*KEY\_ptr != 0); // wait for pushbutton KEY release  break;  }  }    } } |
| --- |

C:/Users/michaelchan/Downloads/LAB2/Lab2\_ECE332\_2021/capture\_image/capture\_image.c:57:13: error: stray '\223' in program

C:/Users/michaelchan/Downloads/LAB2/Lab2\_ECE332\_2021/capture\_image/capture\_image.c:57:32: error: 'Count' undeclared (first use in this function)

C:/Users/michaelchan/Downloads/LAB2/Lab2\_ECE332\_2021/capture\_image/capture\_image.c:57:32: note: each undeclared identifier is reported only once for each function it appears in

C:/Users/michaelchan/Downloads/LAB2/Lab2\_ECE332\_2021/capture\_image/capture\_image.c:57:13: error: stray '\224' in program

C:/Users/michaelchan/Downloads/LAB2/Lab2\_ECE332\_2021/capture\_image/capture\_image.c:57:39: error: 'd' undeclared (first use in this function)

Compilation stopped.

Code currently working as of Mar 8, 20222:45:

| // Michael was here // #include "address\_map\_arm.h"  #include <time.h> #include <stdio.h> #include <stdlib.h> #include <unistd.h> #define KEY\_BASE 0xFF200050 #define VIDEO\_IN\_BASE 0xFF203060 #define FPGA\_ONCHIP\_BASE 0xC8000000   /\* This program demonstrates the use of the D5M camera with the DE1-SoC Board  \* It performs the following:   \* 1. Capture one frame of video when any key is pressed.  \* 2. Display the captured frame when any key is pressed.  \*/ /\* Note: Set the switches SW1 and SW2 to high and rest of the switches to low for correct exposure timing while compiling and the loading the program in the Altera Monitor program. \*/ int main(void) {  volatile int \* KEY\_ptr = (int \*) KEY\_BASE;  volatile int \* Video\_In\_DMA\_ptr = (int \*) VIDEO\_IN\_BASE;  volatile short \* Video\_Mem\_ptr = (short \*) FPGA\_ONCHIP\_BASE;   int x, y;  int count = 0;  setenv("TZ", "EST5EDT", 1); // Sets Timezone to EST    /\* Define temporary variables \*/  time\_t rawtime; // Initialize time       \*(Video\_In\_DMA\_ptr + 3) = 0x4; // Enable the video   while (1)  {  while (1)  {  if (\*KEY\_ptr != 0) // check if any KEY was pressed  {  \*(Video\_In\_DMA\_ptr + 3) = 0x0; // Disable the video to capture one frame  while (\*KEY\_ptr != 0); // wait for pushbutton KEY release  break;  }  }  count++; // Keeps track of number of images captured  time(&rawtime); // Read the current system time   /\* Display the time in EST and UTC \*/  printf ("Count: %i, Time: %s", count, ctime(&rawtime));   short input[320][240];  // This saves  for (y = 0; y < 240; y++) {  for (x = 0; x < 320; x++) {  input[x][y] = \*(Video\_Mem\_ptr + (y << 9) + x);  //\*(Video\_Mem\_ptr + (y << 9) + x) = temp2;  }  }   // This writes  for (y = 0; y < 240; y++) {  for (x = 0; x < 320; x++) {  //input[x][y] = \*(Video\_Mem\_ptr + (y << 9) + x);  \*(Video\_Mem\_ptr + (y << 9) + x) = input[x][y];  }  }    while (1)  {  if (\*KEY\_ptr != 0) // check if any KEY was pressed  {  \*(Video\_In\_DMA\_ptr + 3) = 0x4; // Enable the video in preparation for next image capture  while (\*KEY\_ptr != 0); // wait for pushbutton KEY release  break;  }  }    } } |
| --- |

The time code needs to be changed to the one below.

The issue is that it displays an incorrect time when the date is moved forward:

Local Time: 03/04/2022 -5:32:41 Counter:0

When the time is really:

Local Time: 03/03/2022 19:35:00 Counter:0

<https://www.techonthenet.com/c_language/standard_library_functions/time_h/gmtime.php>

<https://www.onlinegdb.com/>

Correct by using the code below:

| /\* localtime example \*/ #include <stdio.h> /\* puts, printf \*/ #include <time.h> /\* time\_t, struct tm, time, localtime \*/ #include <stdlib.h> #include <unistd.h> int main () {  setenv("TZ", "EST5EDT", 1);  time\_t rawtime;    time (&rawtime);  printf ("Current local time and date: %s", ctime(&rawtime));    sleep(2);  time (&rawtime);  printf ("Current local time and date: %s", ctime(&rawtime));    sleep(2);  time (&rawtime);  printf ("Current local time and date: %s", ctime(&rawtime));    return 0; } |
| --- |

| #include <stdio.h>  #include <time.h>  #include <string.h> #include <stdlib.h> #define BUF\_LEN 256  int main(void) {  setenv("TZ", "EST5EDT", 0);  char buf[BUF\_LEN] = {0};  time\_t rawtime = time(NULL);  if (rawtime == -1) {  puts("The time() function failed");  return 1;   }    struct tm \*ptm = localtime(&rawtime);    if (ptm == NULL) {  puts("The localtime() function failed");  return 1;  }   strftime(buf, BUF\_LEN, "Today is %A", ptm);  puts(buf);    memset(buf, 0, BUF\_LEN);  strftime(buf, BUF\_LEN, "The month is %B", ptm);  puts(buf);    memset(buf, 0, BUF\_LEN);  strftime(buf, BUF\_LEN, "Today is %-d day of %B", ptm);  puts(buf);   memset(buf, 0, BUF\_LEN);  strftime(buf, BUF\_LEN, "Today is %-j day of %G", ptm);  puts(buf);     memset(buf, 0, BUF\_LEN);  strftime(buf, BUF\_LEN, "Today is %-W week of %G", ptm);  puts(buf);     memset(buf, 0, BUF\_LEN);  strftime(buf, BUF\_LEN, "The time is %T", ptm);  puts(buf);     memset(buf, 0, BUF\_LEN);  strftime(buf, BUF\_LEN, "The date is %D", ptm);  puts(buf);     memset(buf, 0, BUF\_LEN);  strftime(buf, BUF\_LEN, "The timezone is %z", ptm);  puts(buf);    return 0; } |
| --- |

| #include <stdio.h> #include <time.h>  #define EST (-5)  int main(int argc, const char \* argv[]) {  /\* Define temporary variables \*/  struct tm \*estime;  time\_t now;   /\* Read the current system time \*/  time(&now);   /\* Convert the system time to GMT (now UTC) \*/  estime = gmtime(&now);   /\* Display the time in PDT and UTC \*/  printf ("Time Stamp: %1d/%1d/%2d %2d:%02d:%2d\n", estime->tm\_mon, estime->tm\_mday, estime->tm\_year,(estime->tm\_hour + EST) % 12, estime->tm\_min, estime->tm\_sec);   return 0; } |
| --- |

| struct tm {  int tm\_sec; /\* seconds, range 0 to 59 \*/  int tm\_min; /\* minutes, range 0 to 59 \*/  int tm\_hour; /\* hours, range 0 to 23 \*/  int tm\_mday; /\* day of the month, range 1 to 31 \*/  int tm\_mon; /\* month, range 0 to 11 \*/  int tm\_year; /\* The number of years since 1900 \*/  int tm\_wday; /\* day of the week, range 0 to 6 \*/  int tm\_yday; /\* day in the year, range 0 to 365 \*/  int tm\_isdst; /\* daylight saving time \*/  }; |
| --- |

| https://github.com/cmruderman/HW-SW-Compression-Decompression/blob/master/src/capture\_image.axf.objdump |
| --- |