COMPUTER-AIDED DESIGN OF SYSTEMS (E-CAD)

The purpose of this semester's project is to design a Keyboard Reading System, Control Robotic Arm and Screen Display.

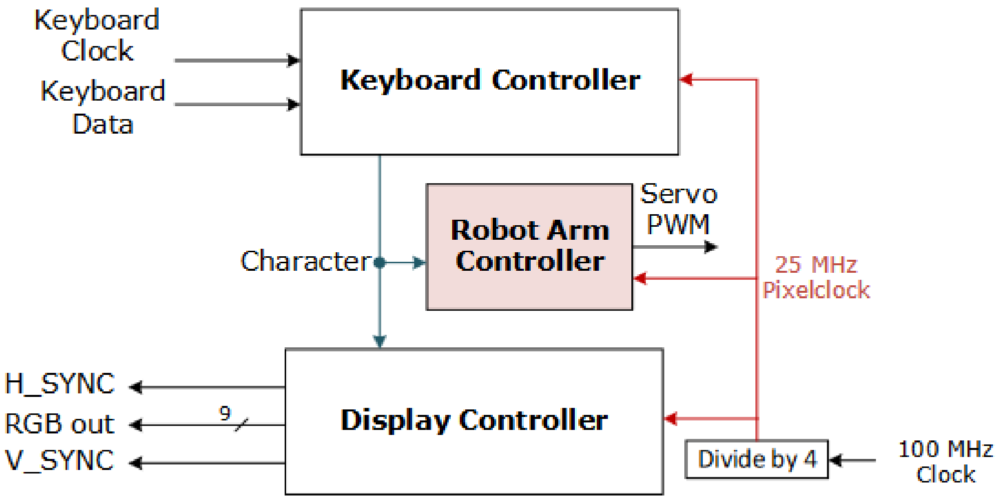
A. General Architecture of the System

The figure below shows an overview of the target system. The sub-designs that make up the system are:

♦ The Keyboard controller, which is responsible for reading specific keys from the keyboard.

♦ The Robot Arm controller, which controls the position of the robotic arm; and

♦ The Display controller, which is responsible for the display on a monitor via VGA protocol.



B. Brief description of the operation

Its operation consists of reading specific keys from the keyboard and moving of the robotic arm and displaying it on a VGA screen, depending on the key read. Consider that the

keyboard controller accepts only F, Q, H, X as possible keys while ignoring any other key. The table below lists how these keys determine the position of the robotic arm.

Key	Meaning	Position of the robotic arm
F	F(olded)	Fully folded
Q	Q(uarterly) extended	Extended by ¼
H	Half extended	Extended by ½
X	eXtended	In full extension

On the screen, the read character is always displayed in the centre of the screen, in white and has a size of 16x8 pixels.

C. Keyboard Controller sub-design

The scan codes that you should scan in this task are:

Key	Scancode
F	2B
Q	15
H	33
X	22

D. Data display on the VGA screen

The main parameters that determine the display on a VGA monitor are the frame refresh rate (Refresh Rate), the number of horizontal lines that make up an image (Number Of Rows) and the number of pixels each row contains (Number Of Pixels Per Row).

The product of the lines and pixels of each line determine the resolution of the screen. The target resolution and refresh rate tell us the frequency of the display clock, Fp (often referred to as the single pixel display clock, pixel clock), according to the relationship :

$$Fp = (\text{Refresh Rate}) \times (\text{Number Of Rows}) \times (\text{Number Of Pixels Per Row})$$

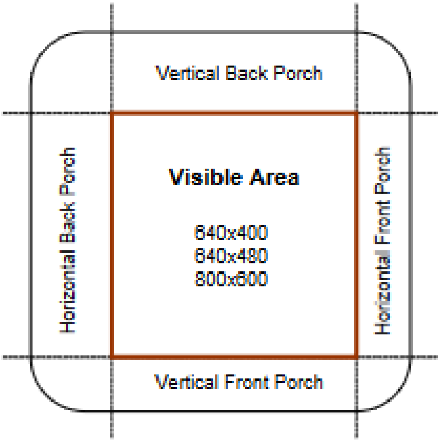
Obviously, for a given screen resolution, an increase in Refresh Rate implies the required increase in frequency of the pixel clock.

The desired display controller has the responsibilities of timing the display and sending it to information to be displayed at specific times. These two processes are discussed in detail below.

D.1. Screen timing

The timing of the screen is achieved by the controller driving the horizontal (HSYNC) and vertical (VSYNC) synchronization. The first one is activated with every scan frame needs to change line while the second one is activated when the viewing of an entire frame is completed. For the proper operation of the screen it is required the scanning of some additional lines (Vertical Porch) and pixels in each line (Horizontal Porch) interval during which blackening, i.e. removal of the load from the pixels of the screen.

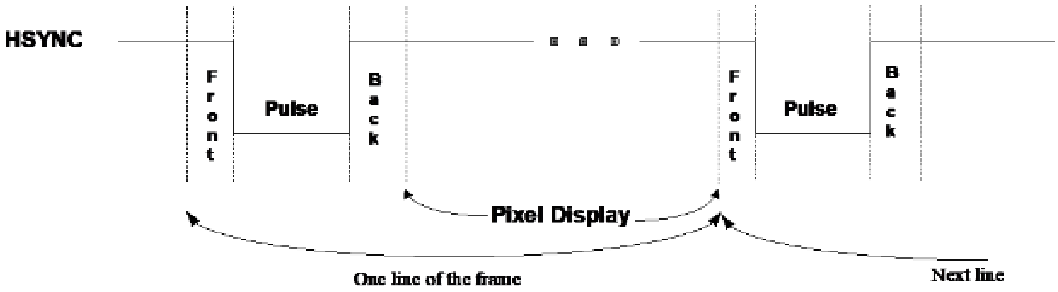
When scanning these extra lines and pixels on each line is not sent any information to be displayed but rather the information sent must be the black color. In the below figure shows the screen and the respective areas into which it is divided. The visible area is shown in the center along with some typical resolutions.



Note that the resolution, e.g., 640x400, refers only to the visible area, while in reality both the number of lines in each frame and the number of pixels per line are more. The lines contained in the areas Vertical Front Porch and Vertical Back Porch do not contain any information as is the case with the pixels in the Horizontal Back and Horizontal Front Porch regions.

D.1.1. Synchronisation of lines

The synchronization of the lines is performed using the HSYNC signal. The completaion of a period of this signal implies the projection of a line including areas not carrying information, i.e. the Horizontal Back Porch and Horizontal Front Porch. A period of the HSYNC signal is shown in figure below.

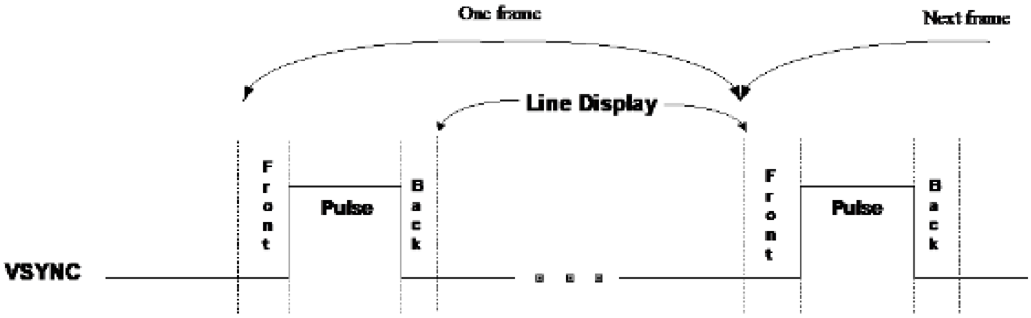


The time that each region lasts is determined by the required resolution and refresh rate and is usually measured in a number of pixel clock cycles. An HSYNC period is on the order of tens of microseconds.

D.1.2. Synchronisation of frames

The synchronization of the frames is performed using the VSYNC signal. The completion of a period of this signal implies the display of an entire frame, i.e. the set of lines that make it up.

The VSYNC period also includes the regions that do not carry information i.e. Vertical Back Porch and Front Porch, respectively. A period of the VSYNC signal is shown in the figure.



Unlike line synchronization, the time that each region in a VSYNC signal period lasts is usually measured in a number of lines, and a VSYNC period is on the order of tens of mseconds.

For your project, we will use a resolution of 640x400 and a refresh rate equal to 70Hz. So the parameters (most commercial monitors support a tolerance of ±5% in terms of pixel clock) of the target system are the following:

♦ Pixel Clock frequency = 25 MHz. This frequency should be generated with sufficient

accuracy

♦ HorizontalFrontPorch = 16 pixels

♦ HSYNCPulse = 96 pixels

♦ HorizontalBackPorch = 48 pixels

♦ VisiblePixels = 640 pixels

♦ VerticalFrontPorch = 12 rows

♦ VSYNCPulse = 2 rows

♦ VerticalBackPorch = 35 rows

♦ VisibleRows = 400 rows

For this particular analysis, the Pulse for the HSYNC signal is negative logic, while for VSYNC it is positive logic (see also the figures above). More generally, however, the polarity of these signals along with their timing characteristics determine the desired resolution of the display.

D.2. Image display

Each image is represented by the pixels that make up the image. Each pixel can be black, i.e. indistinguishable from the background, or have one of 512 different color shades depending on the intensity of each color component (R, G, B). Our board has the signals RED [2:0], GREEN [2:0] and BLUE[2:0], which are mapped to the following FPGA pins:

Signal	Terminal of FPGA
VGA-RED2	B1
VGA-RED1	D6
VGA-RED0	C8
VGA-GREEN2	C3
VGA-GREEN1	A5
VGA-GREEN0	A8
VGA-BLUE2	D5
VGA-BLUE1	E7
VGA-BLUE0	C9
VGA-HSYNC#	B7
VGA-VSYNC#	D8

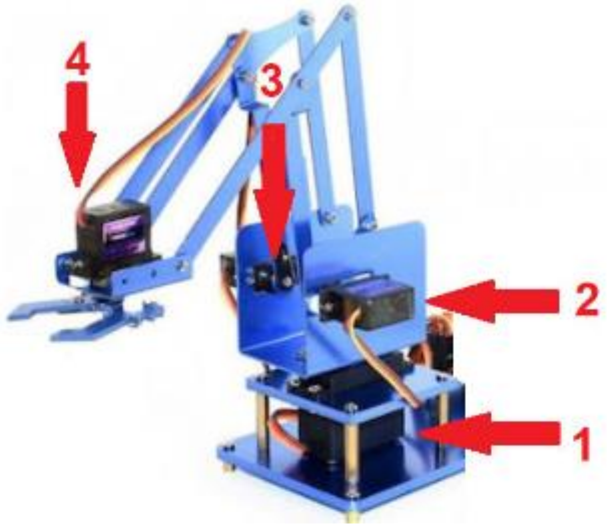
In the present task, a black and white image is requested, which means that it is sufficient that the pixels of the displayed character is white and the rest of the screen is black.

So every time you want to display a white pixel of the character you should set RED [2:0] = GREEN [2:0] = BLUE[2:0] = 111_2 = 7_{10} while you should set all color components to 0 for every other pixel in the display area.

Recall that during the blackening time, i.e. when the extra lines and pixels in each line outside the display area are scanned, the information sent must be in black.

E. Robot Arm Controller Sub-design

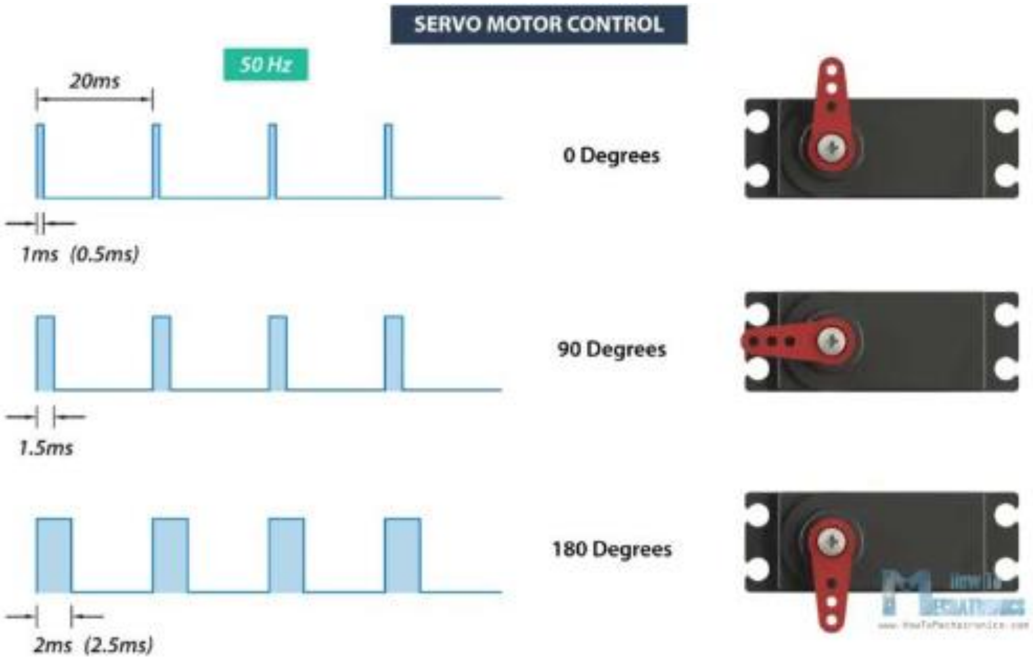
The position as well as the movements of a robotic arm depend on servo motors (servos for short). The particular robotic arm that we will use in this work has 4 servos which are shown in the picture below:



Servo 1 controls the rotation of the robotic arm, servo 2 controls the movement of the robotic arm up and down, servo 3 controls the movement of the robotic arm forward and backward and finally servo 4 controls the opening and closing of the caliper. In this paper, only servo 3 is requested to be dealt with, although what is described below applies to the other servos and almost all servos on the market.

Each servo is controlled by a clock signal, with a period of 20ms, whose duty cycle determines the position the servo will move the arm. Recall that by duty cycle we mean the percentage of cycle time in which the signal is at 1 in each period. Because we are essentially embedding the motion information in the pulse width the signals we use to control the servos can be thought of as PWM (Pulse Width Modulated) signals.

The servo you are asked to test in this task requires a PWM signal with a period of 20ms and recognizes duty cycle 5 to 10% (1ms - 2ms) for the various requested positions. See the figure below showing the various servo positions depending on the duty cycle of the PWM signal (the values in parentheses are not supported in the lab servo):



So if we assume that all the subsystems in our task are timed with the Pixel Clock which should be 25MHz according to the previous, you can easily calculate that your pulse period (20ms) corresponds to 500,000 cycles of the Pixel Clock, while the various positions requested in your task are as follows:

Key	Position of the arm	Duty Cycle of PWM signal	Successive cycles that the signal must be 1
F	Fully folded	5.00%	25,000
Q	Extended by ¼	6.25%	31,250
H	Extended by ½	7.50%	37,500
X	In full extension	10.00%	50,000

The FPGA pin on which the PWM signal is mapped is:

Signal	Terminal of FPGA
PWM	M3