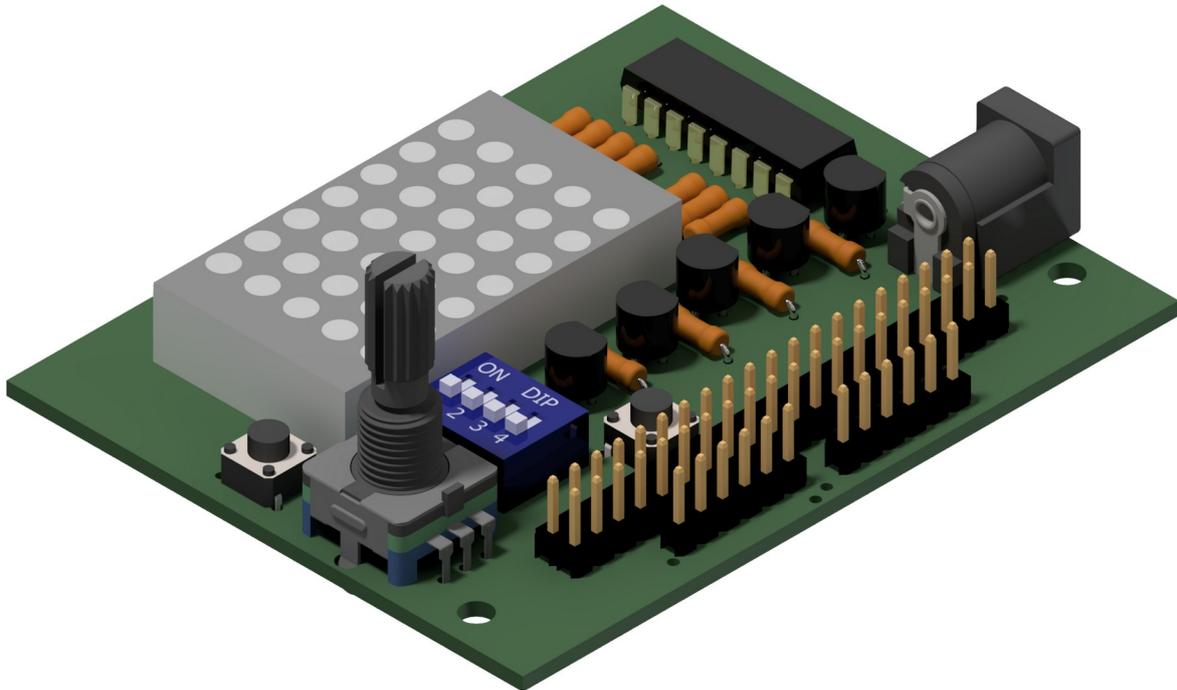


Visera HX-8K

(HX-8K Visor Hat)

Versión 1.0 - Quarantine edition



Placa de extensión de entradas y salidas digitales para prácticas de tecnología FPGA con la iCE40HX-8K Breakout Board de Lattice



Esquemático, trazado del PCB, información, circuitos sintetizables y modelos 3D disponibles para descarga libre y gratuita en: <https://miguelgrassi.com.ar/visera>

Miguel Grassi – 2020
(migrassi@gmail.com)



Material publicado bajo licencia Creative Commons BY NC. Permitida la distribución, copia y modificación libre y gratuita, con fines no comerciales y atribución al autor.

CARACTERÍSTICAS PRINCIPALES

Entradas

- 2 Tact switch
- 4 DIP switch
- 1 Encoder rotativo

Salidas

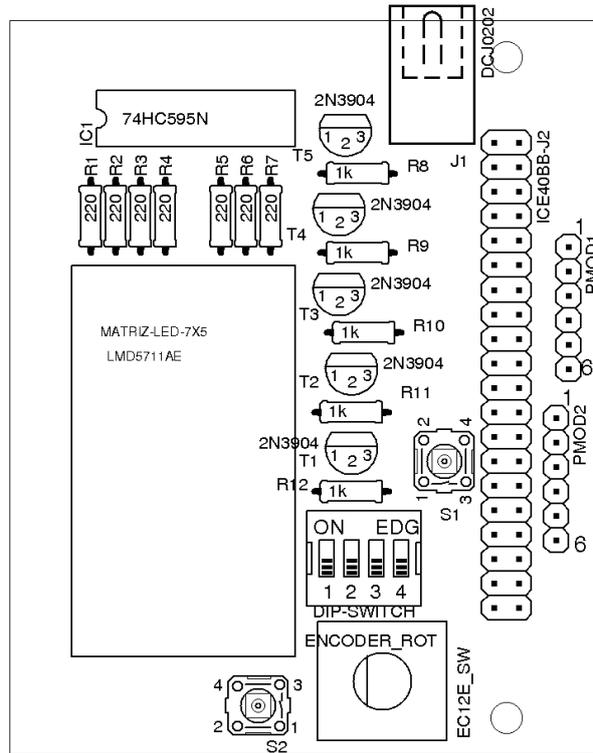
- 1 Matriz de 35 LEDs (7X5)
- 1 Réplica completa Header J2
- 2 PMOD Digilent 1x6

Alimentación

- 3.2 V CC (5 V Tolerant*)

Consumo

- 10 a 350 mA*



*Ver Consideraciones de Compatibilidad

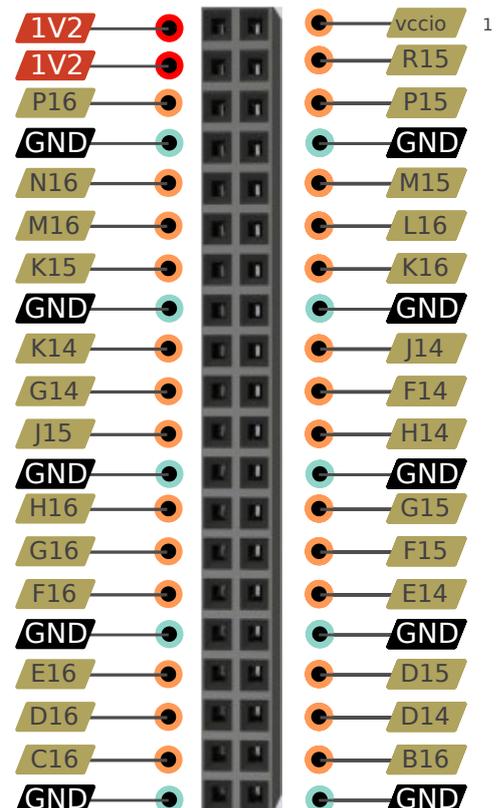
PINOUT

MATRIZ DE LEDS 7 X 5								
Columna	C1	C2	C3	C4	C5			
Pin	N16	M16	K15	K14	G14			
Pin 74HC595	QA	QB	QC	QD	QE	QF	QG	QH
Fila	1	2	3	4	-	5	6	7
Bits Salida	7	6	5	4	-	2	1	0
Data	M15							
Clock	R15							
Latch	P15							

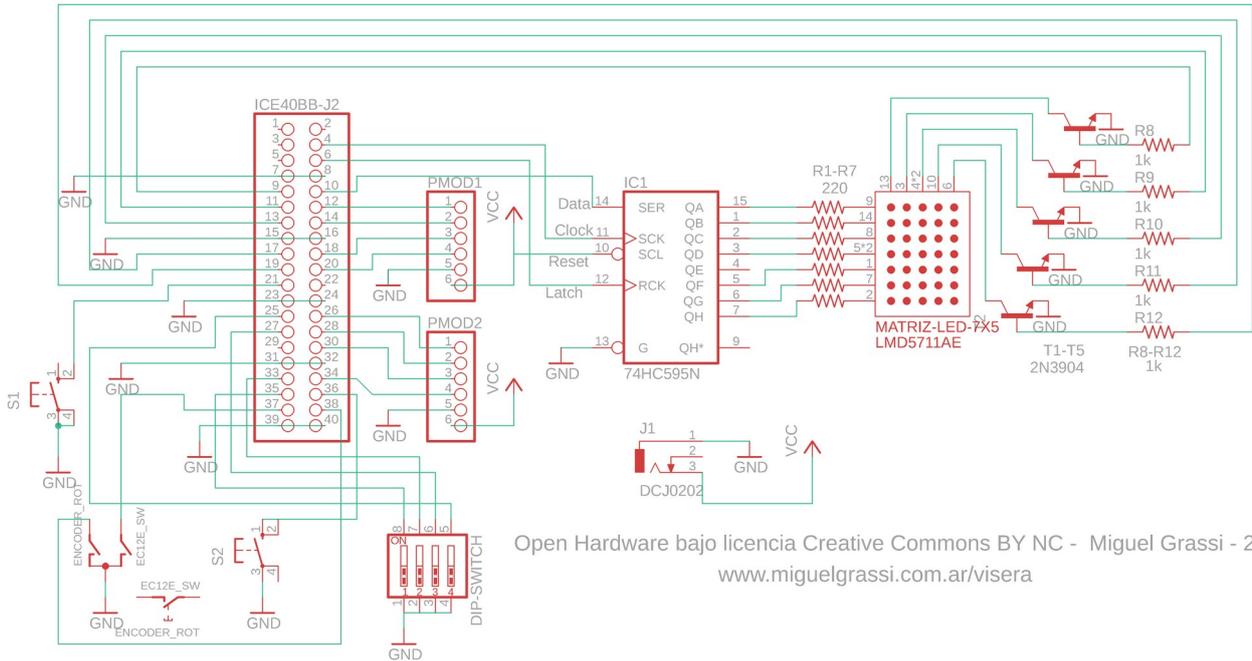
INPUTS				M A S A ES NECESARIO HABILITAR LAS RESISTENCIAS DE PULL-UP EN TODAS LAS ENTRADAS	
Tact Switch 1	J15				
Tact Switch 2	D14				
Dip Switch	D16	E16	G16		H16
Encoder	C16		B16		

P M O D 1	1	L16
	2	K16
	3	J14
	4	F14
	5	GND
	6	VCC

P M O D 2	1	G15
	2	F15
	3	E14
	4	D15
	5	GND
	6	VCC

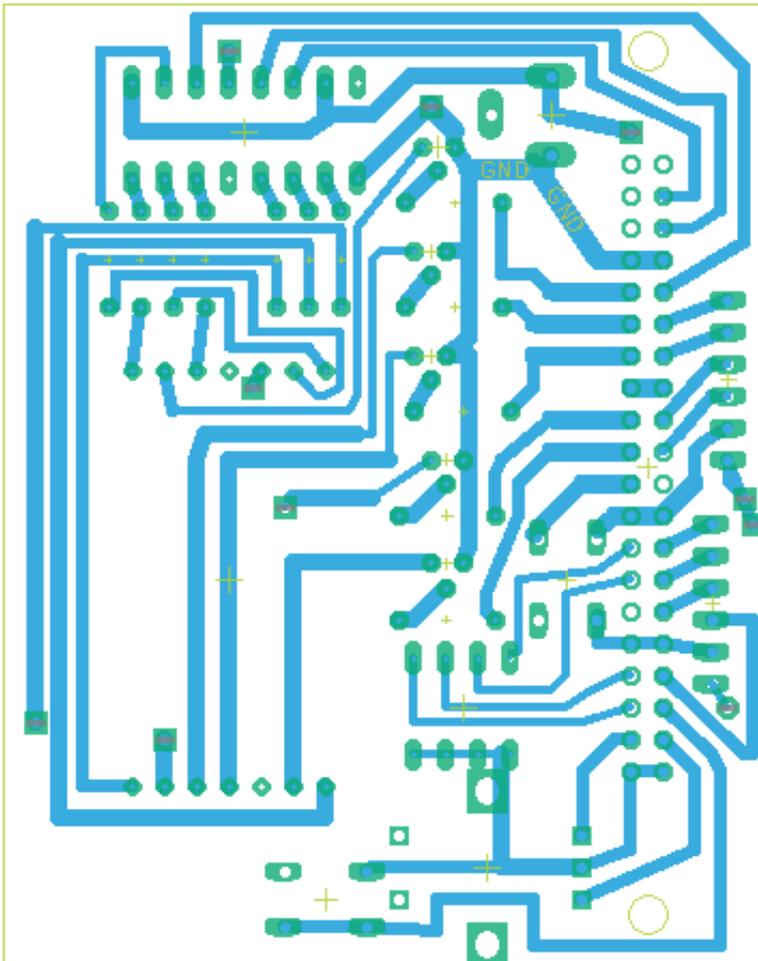


ESQUEMATICO



Open Hardware bajo licencia Creative Commons BY NC - Miguel Grassi - 2020
www.miguelgrassi.com.ar/visera

PCB y BOM



Elemento	Valor	Package
DIP-SWITCH	x4 DIP	EDG-04
ENCODER_ROT	EC12E	EC12E
IC1	74HC595N	DIL16
ICE40BB-J2	2X20 pin	XXlarge
J1	Jack Barrel	DCJ0202
MATRIZ-LED-7X5	LMD5711AE	LMD5711AE
PMOD1-PMOD2	1X06	pinhead
R1-R7	220	0207/7mm
R8-R12	1k	0207/7mm
S1-S2	tact-switch	B3F-10XX
T1-T5	2N3904	T092

Trazado optimizado para ruteo por CNC o producción artesanal mediante placa presensibilizada u otro método manual similar

Visera HX-8K – (HX-8K Visor Hat)

Versión 1.0 - Quarantine edition

Extensión I/O para la iCE40HX-8K Breakout Board Evaluation Kit de Lattice

La presente placa se diseñó como un accesorio de conexión directa a la placa de entrenamiento en tecnología FPGA “iCE40HX-8K Breakout Board” de Lattice Semiconductors, un modelo basado en la conocida familia FPGA iCE40, de esa marca.

La placa se inserta directamente en el único conector de 40 pines que viene poblado de fabrica en la Breakout Board, quedando “en voladizo” hacia afuera, dejando los otros tres conectores de 40 pines, jumpers, LEDs y demás partes de la placa original completamente descubiertos y libres para otras aplicaciones, como puede verse en la Figura 1. Esta disposición flotante, sujeta sólo por un lado, es la que le da el nombre de Visera o *Visor Hat*. La placa replica hacia arriba todos los pines del conector ocupado, de modo que si se deja la Visera sin alimentar, es posible acceder a todos los pines del conector original tal como si esta no existiera, sin necesidad de desmontarla¹.

La Visera provee además 2 conectores de extensión en formato Digilent Pmod™ de 6 pines para la conexión de cualquiera de los múltiples periféricos adicionales disponibles con esa especificación, que es prácticamente un estándar de la industria hoy.

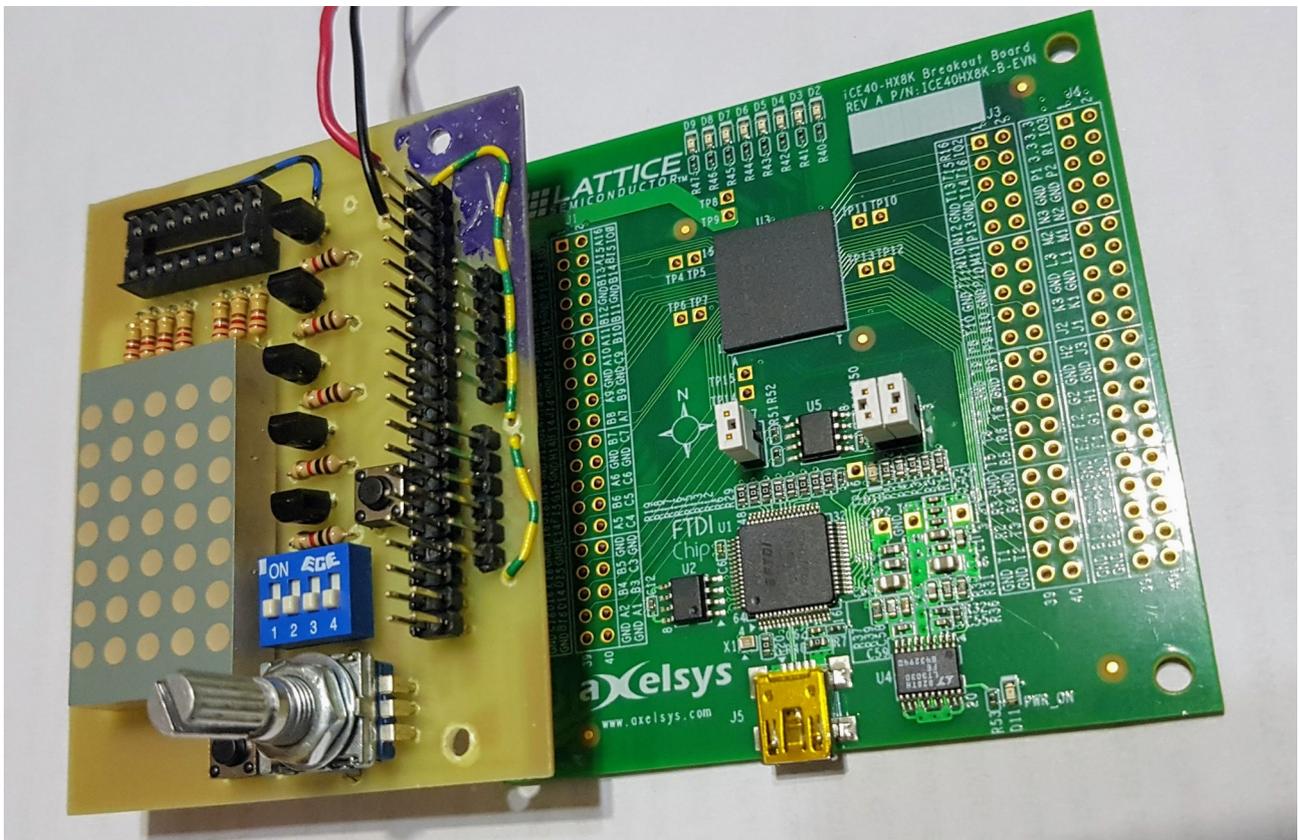


Figura 1: Prototipo de la Visera 1.0, montado en la Breakout Board

1 Las implicancias de uso del conector original sin desconectar la Visera, se tratan en detalle en Consideraciones de Compatibilidad

Origen y propósito

La Visera fue desarrollada en Mayo de 2020, en el marco de la pandemia mundial del COVID19 y particularmente la cuarentena estricta de la Ciudad de Buenos Aires. Fue resuelta, por lo tanto, con el material del cajón de rezagos del autor, dadas las dificultades para adquirir componentes nuevos. Esta importante restricción de diseño constituye una de sus fortalezas, ya que prácticamente cualquier aficionado o profesional de la electrónica tendrá disponibles esos mismos componente en su propio cajón, o los podrá adquirir con facilidad en negocios de cercanías.

Con el mismo criterio fue diseñado también el PCB, cuyo trazado se muestra en la Figura 2. Se trata de una plaqueta simple faz, con trazos generosos, para poder ser fabricada en forma artesanal, y que utiliza sólo componentes TH, que permiten su fácil montaje y soldadura manual.

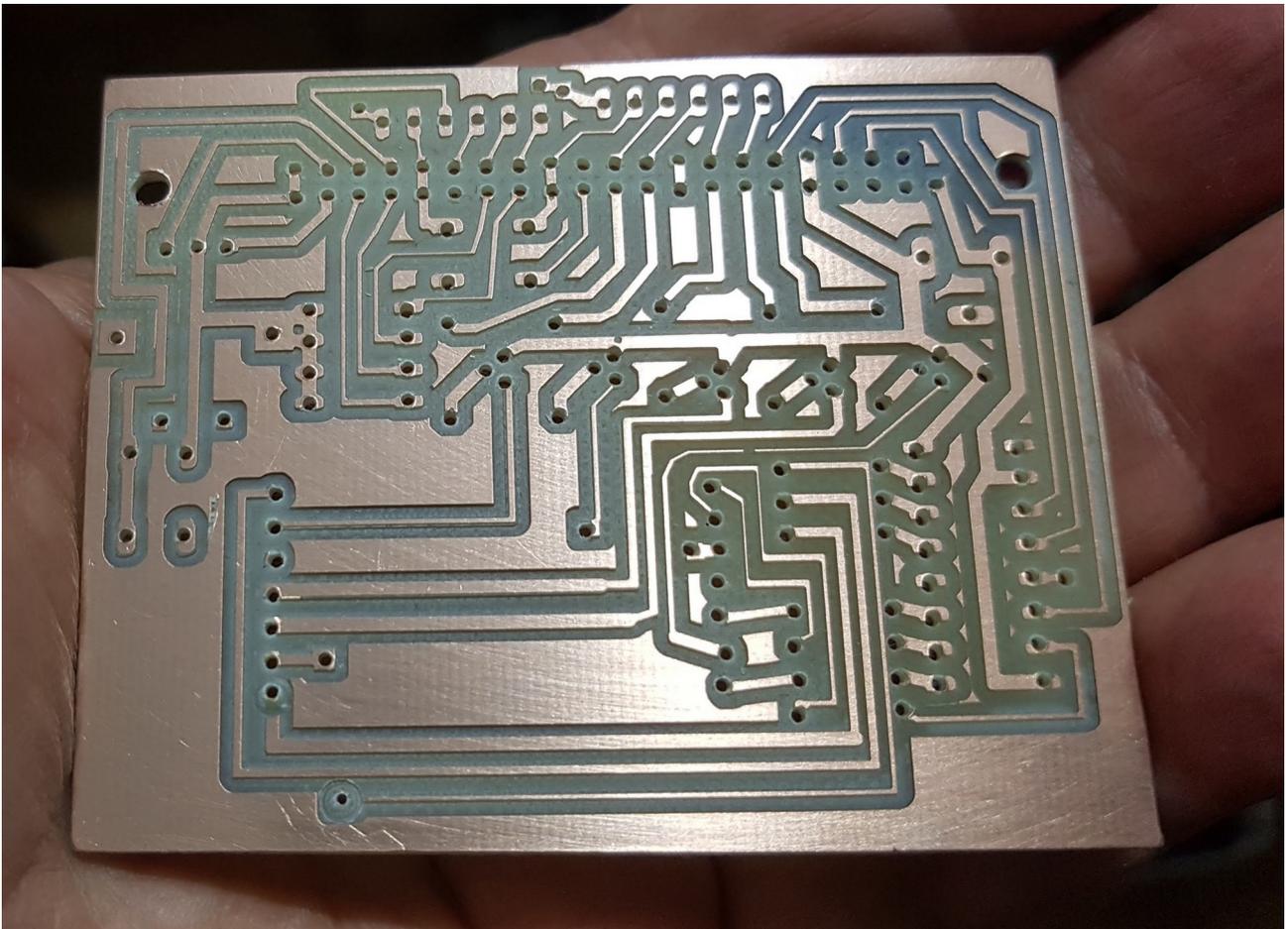


Figura 2: Trazado del PCB del prototipo, resuelto por CNC en placa de cobre simple faz. La desaparición de algunas islas pequeñas se debe al uso de una mecha más gruesa que la requerida, a causa de las restricciones mencionadas.

La condición “cuarenténica” está presente también en el propósito del prototipo: permitir desarrollar prácticas intensivas de FPGA y HDL, tema que el autor venía -como muchos- postergando desde siempre, frente a urgencias laborales o de estudio más perentorias que consumían el tiempo que la cuarentena -al menos en teoría- prometía otorgar.

La elección de la familia iCE40 como tecnología para un primer acercamiento al tema obedece principalmente a la existencia de una completa *toolchain* libre para esa familia, que permite el desarrollo del proceso completo, desde la programación en Verilog o VHDL, hasta la síntesis y el testbench con herramientas totalmente open source. Esta disponibilidad de herramientas libres -de no tan larga data, por cierto- generó la aparición de muchos tutoriales, ejercicios, módulos y recursos educativos, también libres, y muy valiosos, que facilitan una rápida entrada en tema. Al final de ese documento se incluye una lista y links a varios de estos recursos.

Dentro de la amplia gama de placas experimentales basadas en la FPGA iCE40, se escogió particularmente la *iCE40HX-8K Breakout Board* en lugar del popular *iCEstick40HX-1K* dado que, por un precio apenas un poco más alto que aquel, ofrece capacidades bastante mayores, incluyendo 7680 *logic cells* y 4 salidas de 40 pines cada una, frente a las 1280 *logic cells* y 2 salidas de 10 pines cada una del Stick, entre las más importantes diferencias. Más allá de eso, la compatibilidad entre ambos modelos es muy alta, de modo que la gran cantidad de recursos educativos, módulos, etc., disponibles para el *iCEstick* pueden aplicarse casi sin modificación alguna en la *Breakout Board*. Por ejemplo, es soportada directamente por el software *iceStudio*² una herramienta open source ideal para dar los primeros pasos en este campo, que puede ser ejecutada tanto en Linux como Windows o macOS y que se encargará de instalar los drivers necesarios, y de adaptar automáticamente los ejemplos al pinout y nomenclatura de la placa.

Montaje

La Visera tiene un header de 2x20 pines, identificado como "ICE40BB-J2", que debe poblarse con una tira doble de 20 pines extra-largos, usualmente conocidos como double-sided, como los que se muestran en la Figura 3. Esta componente se monta con los pines largos hacia el lado cobre y se suelda como una tira de pines normal, por ese mismo lado. De este modo, los pines inferiores sobresalen largamente y posibilitan una fácil inserción de toda la placa, una vez terminada, en el conector J2 de la Breakout Board, es decir, el único que viene instalado de fábrica en la misma.



Figura 3: Tira de 2x20 pines double-sided con la que se puebla el conector "ICE40BB-J2" de la Visera

2 <https://icestudio.io/>

No es conveniente que la Visera quede sujeta sólo por el conector. Si bien el contacto es muy firme, un conector de este tipo no está previsto para actuar como único soporte mecánico de una placa adicional de tamaño considerable, sobre la cual se actuará manualmente, y con frecuencia, para operar los switches, el encoder rotativo y conectar otros módulos o componentes. Se diseñó entonces un gabinete a medida, presentado en la Figura 4, para alojar la Breakout Board y sostener la Visera, de modo de evitar ejercer presión mecánica excesiva sobre el conector J2 de la placa FPGA.

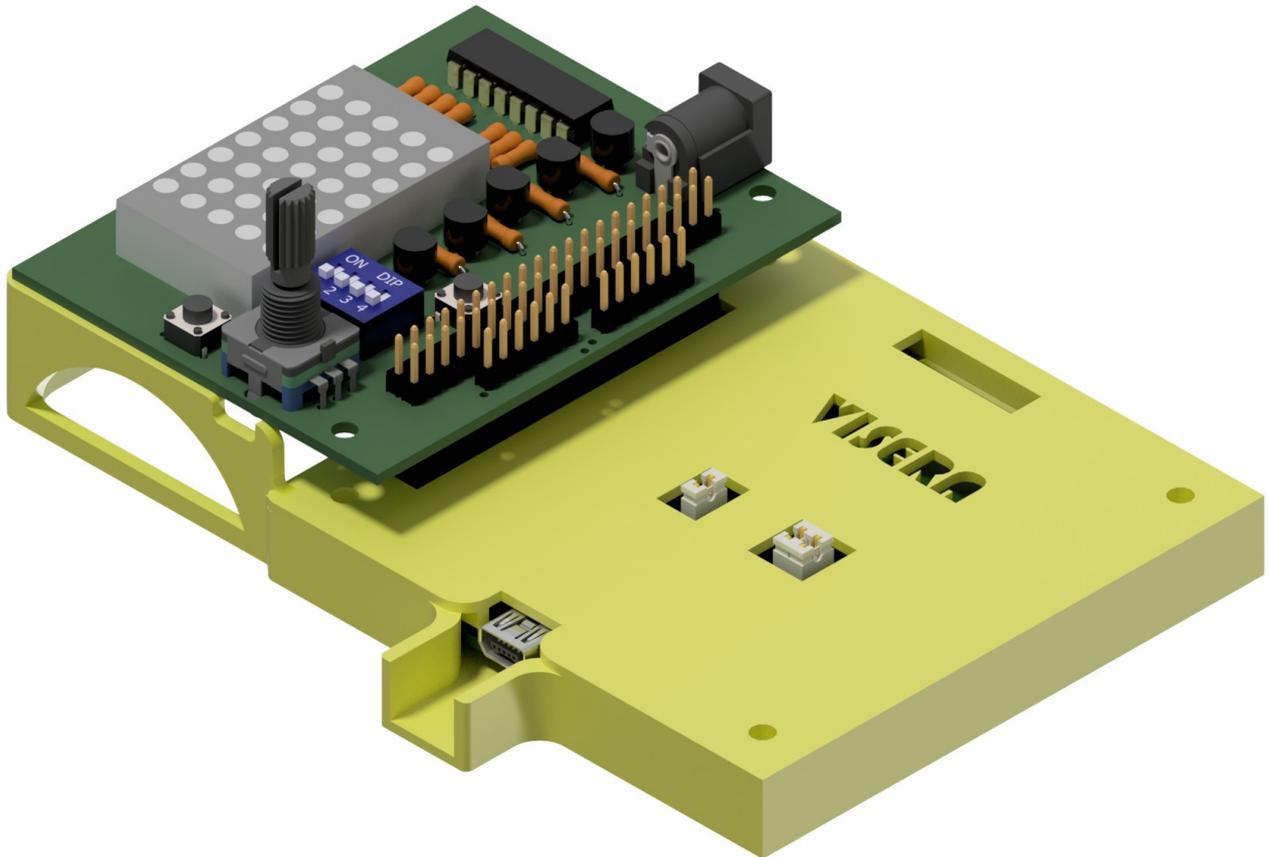


Figura 4: Modelo de gabinete propuesto para contener la Visera montada en la Breakout Board

La tapa que protege la placa principal permite visualizar los LEDs integrados en la misma y operar sobre los *jumpers* de configuración sin ser quitada. Un marco perimetral lateral da firme soporte a la Visera, permitiendo accionar manualmente sobre sus controles y conectar otros componentes o módulos Pmod con comodidad. Se adicionó un apéndice lateral que contiene y protege el conector USB de la Breakout Board, dado que se observó que la manipulación propia del uso somete al conector hembra montado en la placa a un esfuerzo considerable, presentando cierta tendencia a aflojarse y producir falsos contactos, riesgos que esta protección disminuye considerablemente.

El prototipo realizado fue impreso mediante el método de Modelado por Deposición Fundida (impresión 3D) en Acronitrilo Butadieno Estireno (ABS) con excelentes resultados. Puede accederse a los archivos STL necesarios para imprimir las dos piezas de este gabinete (base y tapa) en los enlaces presentados al final de este documento.

Prestaciones y uso

La Visera presenta un conjunto de entradas y salidas destinados a facilitar un variado tipo de pruebas y ejercicios de síntesis de circuitos digitales en la FPGA sin necesidad de elementos adicionales de hardware. Se detallan a continuación cada uno de esos elementos, con una breve explicación de su propósito y posibilidades elementales.

Entradas digitales

La Tabla 1 presenta un resumen de las entradas digitales, identificando los ocho pines de la FPGA iCE 40 a la que se encuentran respectivamente conectados. Estas entradas son:

- Dos pulsadores sin retención, S1 y S2, conectados a los pines J15 y D14 de la iCE40 respectivamente. Son adecuados para practicar con la captura de eventos momentáneos y como interfaz interactiva simple con el usuario.
- Cuatro DIP Switchs, es decir llaves con retención, identificados con los números 1 a 4 y conectados respectivamente a los pines D16, E16, G16 y H16 de la iCE40. Permiten 16 combinaciones diferentes, útiles para practicar con números binarios hasta 1111 (0xF) o seleccionar distintas configuraciones de parámetros o módulos sin necesidad de sintetizar nuevamente el circuito, entre muchas otras aplicaciones.
- Un encoder rotativo de cuadratura (o encoder incremental) de 24 pasos por vuelta, con sus dos canales conectados a los pines B16 y C16 de la iCE40. Es útil para detectar movimientos de rotación, medir ángulos, velocidades y todo tipo de ejercicios con secuencias de pulsos, código grey, BCD, etc. Útil también para usar como interfaz de usuario para el *setting* de parámetros numéricos discretos o continuos, valores incrementales, escalas, etc.

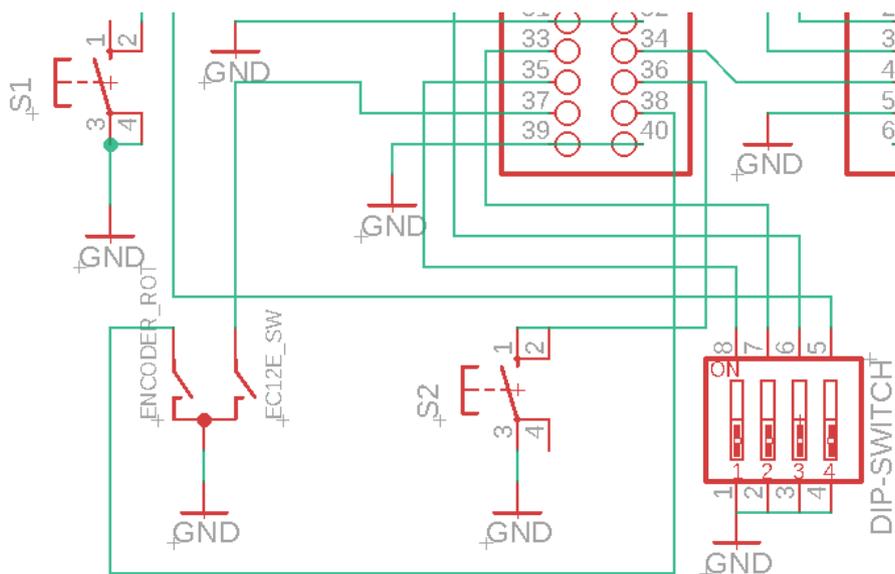


Figura 5: Detalle de conexionado de las entradas

ELEMENTO		PINES iCE40				M A S A	HABILITAR PULL-UPS EN TODAS LAS ENTRADAS
Tact Switch 1		J15					
Tact Switch 2		D14					
Dip Switches 1 a 4		D16	E16	G16	H16		
Encoder. Canales 1 y 2		C16		B16			

Tabla 1: Entradas digitales y pines de la FPGA correspondientes a cada una.

Lógica de las entradas

Debe tenerse en cuenta que, como se observa en la Figura 5, todas las entradas cierran su circuito a GND cuando son accionadas (estado ON) quedando, en cambio, el respectivo pin de entrada de la FPGA al que estén conectas en estado flotante cuando el elemento está en posición normal o de reposo (estado OFF).

Es decir, la placa no dispone de resistencias de pull-up propias, con el objetivo de dejar su respectiva entrada liberada y sin interferencia, para el caso en que se quiera conectar otro elemento a la misma entrada cuando la Visera está sin alimentación. Como consecuencia de esto, **para el uso normal de cualquier entrada de la Visera es imprescindible habilitar en el circuito a sintetizar la resistencia de pull-up correspondiente al pin de la FPGA conectado a esa entrada.** La lógica resultante en esa entrada será entonces de tipo negativo: un '0' en la entrada indica el interruptor en 'ON' y un '1' indica el estado normal, no activado o interruptor en 'OFF'.

Esta configuración permite, además, que el nivel de tensión alto (el '1' lógico) de cada entrada de la FPGA sea el nivel de tensión interno propio de la iCE40, en forma independiente de la alimentación que se le suministre a la Visera, permitiendo alimentar ésta tanto con 3.3 volts como con 5 volts, sin necesidad de adaptación de niveles lógicos.

Salidas digitales

La placa Visera presenta un conjunto de LEDs dispuestos en una matriz rectangular de 7 filas y 5 columnas, que se puede utilizar de diversas formas, combinables entre sí:

- 1) En conjunto para configurar un único dígito alfanumérico grande (7 pixeles de alto por 5 de ancho o viceversa) suficiente para exhibir todo tipo de números, letras mayúsculas o minúsculas y variados símbolos, de cierta complejidad, como se observa en la Figura 6.

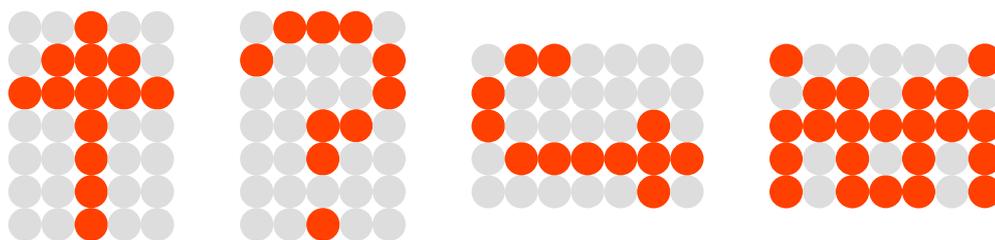


Figura 6: Ejemplo de símbolos y caracteres gigantes usando todos los LEDs de la matriz.

2) Como indicadores digitales individuales, adicionales a los 8 LEDs que ya presenta la placa original, llevando el total a 43 LEDs individuales.

3) En conjunto para emular un display numérico de dos dígitos de 7 segmentos, separados por una línea vertical, como se ejemplifica las representaciones del “39” y “40” de la Figura 7. Este modo es ideal para prácticas que emulen exactamente el comportamiento de los típicos dígitos de cátodo común o ánodo común, de uso muy habitual en electrónica. En el apartado de ejemplos se desarrollará un módulo que acepta 7 entradas digitales y emula en la matriz el dígito respectivo, como si se tratara de un display de este tipo para utilizar en prácticas de electrónica digital.

4) En conjunto para formar dos dígitos, de forma similar al modo de 7 segmentos, pero con mayor flexibilidad para configurar símbolos, ya que se dispone de más del doble de elementos individuales en cada dígito (exactamente 15 en lugar de 7). La Figura 7 muestra la distribución necesaria para generar las letras “ML” (por **M**atriz **L**ed) y la cifra “1.4” usando este modo, que denominamos “7 extendido” y que permite obtener prácticamente todo el alfabeto y varios símbolos, así como mostrar dos caracteres simultáneamente.

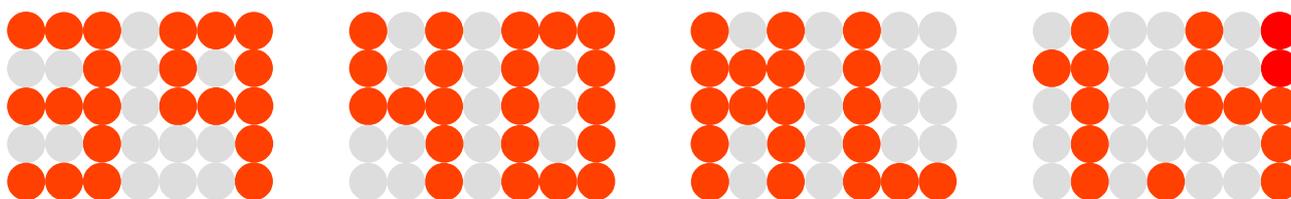


Figura 7: Ejemplo de configuración de dos dígitos. Las dos imágenes de la izquierda corresponden al modo 7 segmentos puro y las dos de la derecha al modo de mayor resolución (2 de 5x3 + 1 de 1x5)

La Figura 8 compone en una misma imagen un ejemplo del modo “7 segmentos” y otro del modo “7 extendido” fotografiados en el display del prototipo.



Figura 8: Muestra del modo 7 segmentos puro (cifra “39”) y del modo 7 extendido (letras “ML”) fotografiados en el prototipo construido.

Funcionamiento de la matriz de LEDs

El display de 1.2” de la Visera consiste en un LMD5711AE³, una matriz conformada por 35 LEDs de 3 mm de diámetro cada uno, dispuestos en 7 filas de 5 elementos cada una. En este modelo de display se conectan entre sí los cátodos de todos los LEDs de una

3 Hoja de datos de la matriz LMD5711AE: <http://www.sycelectronica.com.ar/optoelectronica/LMD5711AE.pdf>

misma columna, mientras sucede lo propio con los ánodos de los LEDs de una misma fila. Cada fila y columna se conecta a un terminal del componente, según el circuito esquemático presentado en la Figura 9.

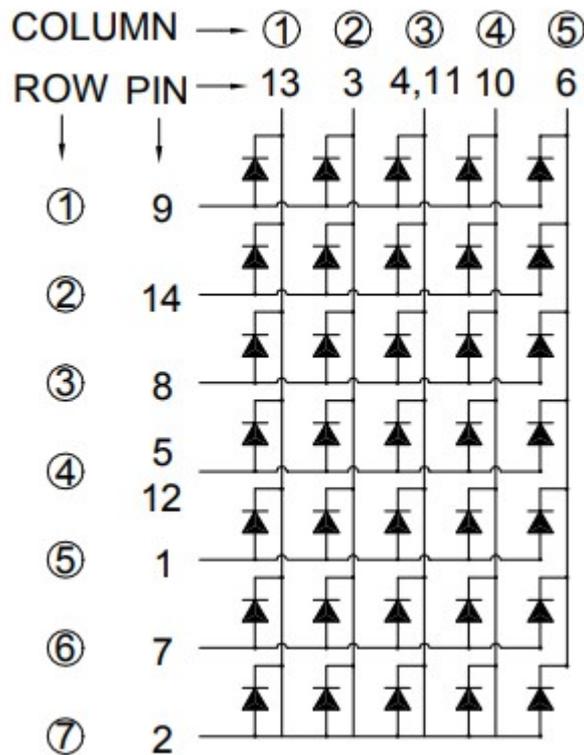


Figura 9: Esquemático del circuito interno del LMD5711AE

Esta disposición, habitual en componentes matriciales, no permite el control independiente de cada LED individual, sino que para su uso requiere la aplicación de una técnica de multiplexado. Esto consiste en conectar a masa cada columna independientemente, de a una por vez, en forma secuencial, de modo que en todo momento solo hay una única columna cuyos LEDs se encenderían si se alimentan sus ánodos con tensión positiva. En forma sincronizada con esto, se deben alimentar entonces sólo aquellas filas cuyos LEDs de la columna conectada a masa en ese momento deban encenderse. Un instante después se desconecta esa columna de masa, se conecta a masa la siguiente y se activan las filas correspondientes a esa columna. Una vez que se llega a la última, se inicia el proceso nuevamente con la primera y se continúa cíclicamente. Si se realiza este proceso con la velocidad suficiente y gracias al fenómeno de persistencia retiniana, la vista del observador percibirá como si todas las columnas estuvieran encendidas simultáneamente y verá el símbolo completo en el display, sin percibir parpadeo alguno.

Hay diferentes formas de controlar las secuencias de filas y columnas desde una FPGA o un microcontrolador. Lo más habitual es el uso de transistores, uno para cada fila y uno para cada columna. En el caso de la Visera se optó por una solución distinta para las filas que para las columnas, de modo que los usuarios puedan practicar diferentes técnicas de multiplexado, electrónica digital y conversiones serie-paralelo de bajo nivel.

Como se observa en la Figura 10, las columnas son manejadas individualmente por sendos transistores 2N3904, en este caso, pero puede usarse cualquier transistor NPN de

uso general. Cada uno de ellos es operado por una salida de la FPGA, tal como se especifica en la Tabla 2 y permite la conexión selectiva de cada columna a GND para efectuar la activación secuencial de las mismas.

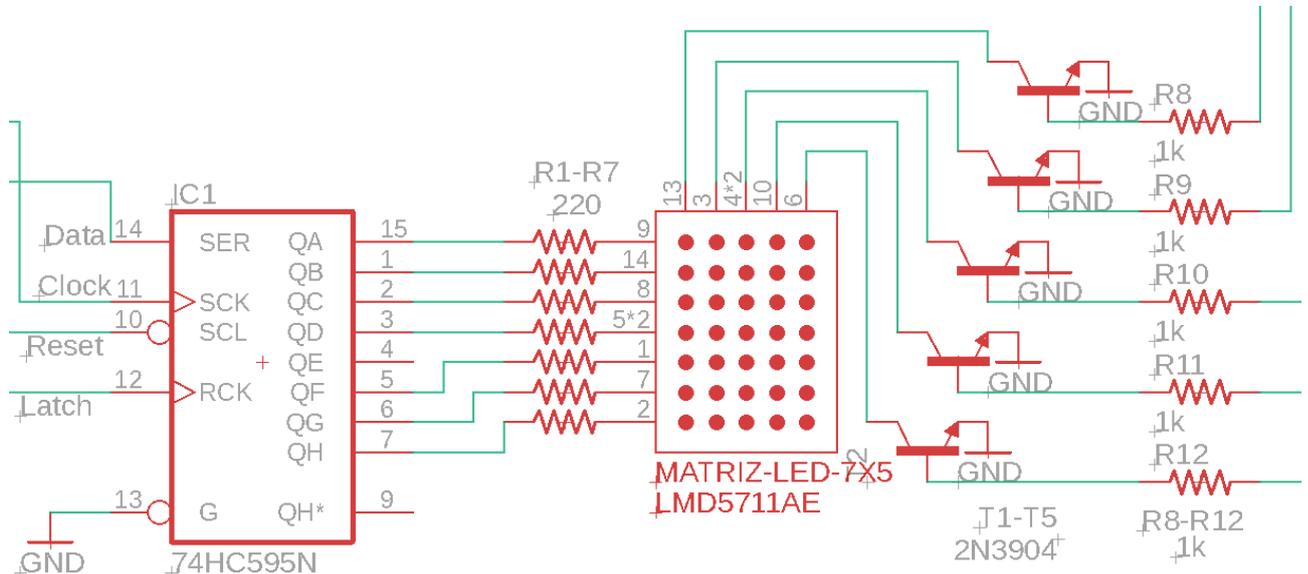


Figura 10: Detalle de los transistores y circuito integrado que controlan la matriz de LEDs

MATRIZ DE LEDS 7 X 5								
Columna	C1	C2	C3	C4	C5			
Pin	N16	M16	K15	K14	G14			
Pin 74HC595	QA	QB	QC	QD	QE	QF	QG	QH
Fila	1	2	3	4	-	5	6	7
Bits Salida	7	6	5	4	-	2	1	0
Data	M15							
Clock	R15							
Latch	P15							

Tabla 2: Conexionado de pines de la matriz de LEDs

Para las filas, en cambio, se optó por usar un integrado 74HC595 que permite recibir hasta 8 bits en forma de comunicación serial (uno a uno) y disponerlos en paralelo, en sus ocho salidas. Siete de estas salidas (QA a QD y QF a QH) se usan para controlar igual número de filas de la matriz LMD5711AE, quedando una de ellas (salida QE) sin uso. De esta manera, para operar las 7 filas de LEDs se requieren solo tres salidas de la FPGA (Data, Clock y Latch) que se individualizan también en la Tabla 2.

La operación de la matriz desde el circuito sintetizado es relativamente simple. Para cada caracter grande o par de caracteres, según el modo que se quiera utilizar, es necesario generar una matriz binaria que se corresponda con los LEDs a encender. La matriz tendrá un "1" en los LEDs que deban encenderse y un "0" en los que se desea mantener apagados. La Figura 11 muestra ejemplos, correspondientes a dos de los modos antes

mencionados. Nótese que como la salida QE del 74HC595 no se usa, el 5° elemento de la matriz de 1s y 0s es irrelevante y puede contener cualquier valor.

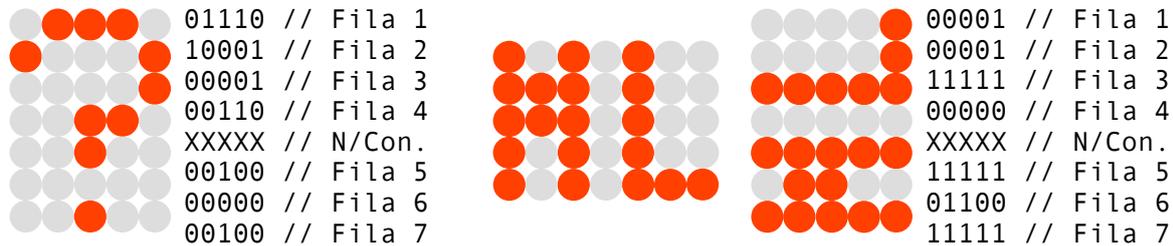


Figura 11: Matrices binarias para generar el símbolo "?" y las letras "M L", rotadas 90°

Una vez determinada esta matriz numérica, es necesario volcarla a la matriz de LEDs, en forma multiplexada, es decir, columna por columna, y activar sincronizadamente el transistor de la columna correspondiente, para que los LEDs de la misma se iluminen.

Las columnas, como ya se dijo, son manejadas por el circuito integrado 74HC595, un *shift register* de 8 bits, con entrada serial y salida paralela con latch, cuyo pinout y diagrama lógico se observan en la Figura 12.

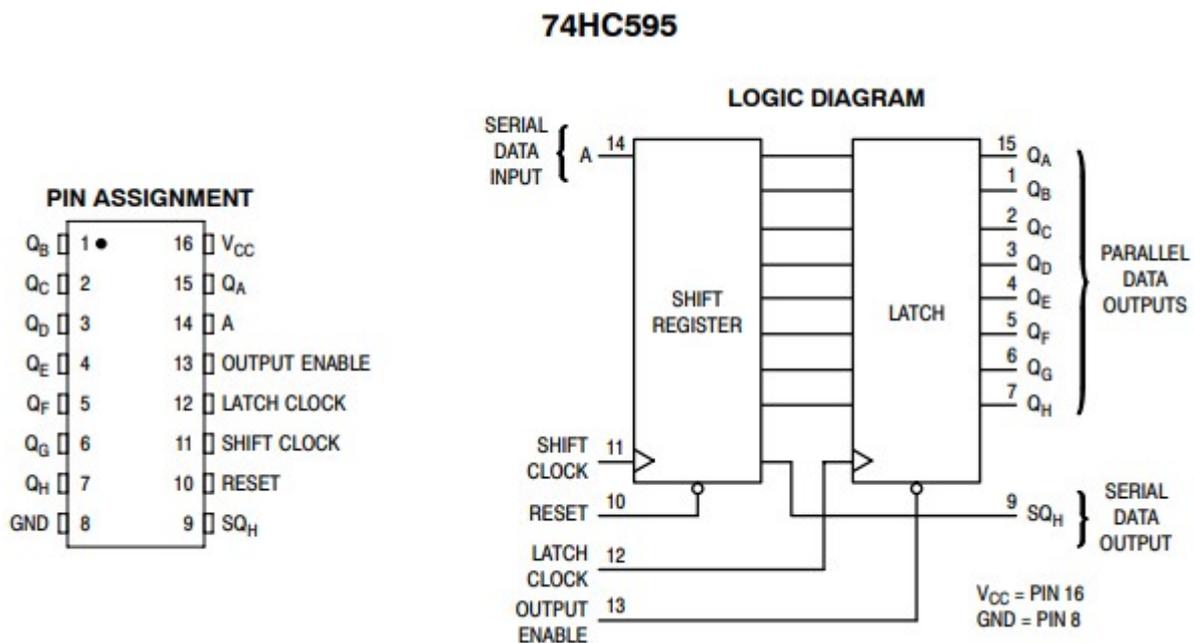


Figura 12: Pinout y diagrama lógico del 74HC595

Para simplificar el trazado del PCB, y al mismo tiempo ahorrar algunas salidas de la FPGA, se optó por una configuración mínima, que utiliza sólo tres entradas: la "A" o entrada de datos serie (en adelante "Data", pin 14 del IC, conectado a M15 de la FPGA), la de *Shift Clock* (pin 11 del IC, en adelante designada simplemente como "Clock", y conectada a R15 de la FPGA) y la de *Latch Clock* (en adelante "Latch", pin 12 del IC, conectado a la salida P15 de la FPGA). Las entradas *Reset* (pin 10 del IC) y *Output Enable* (pin 13 del IC) fueron conectadas directamente a VCC y GND respectivamente, dejando al circuito integrado permanentemente habilitado y la salida de su registro Latch permanentemente presente en las salidas Q1-Q8.

La Figura 13 muestra el diagrama de tiempos del IC 74HC595, que ayuda a comprender bien su funcionamiento.

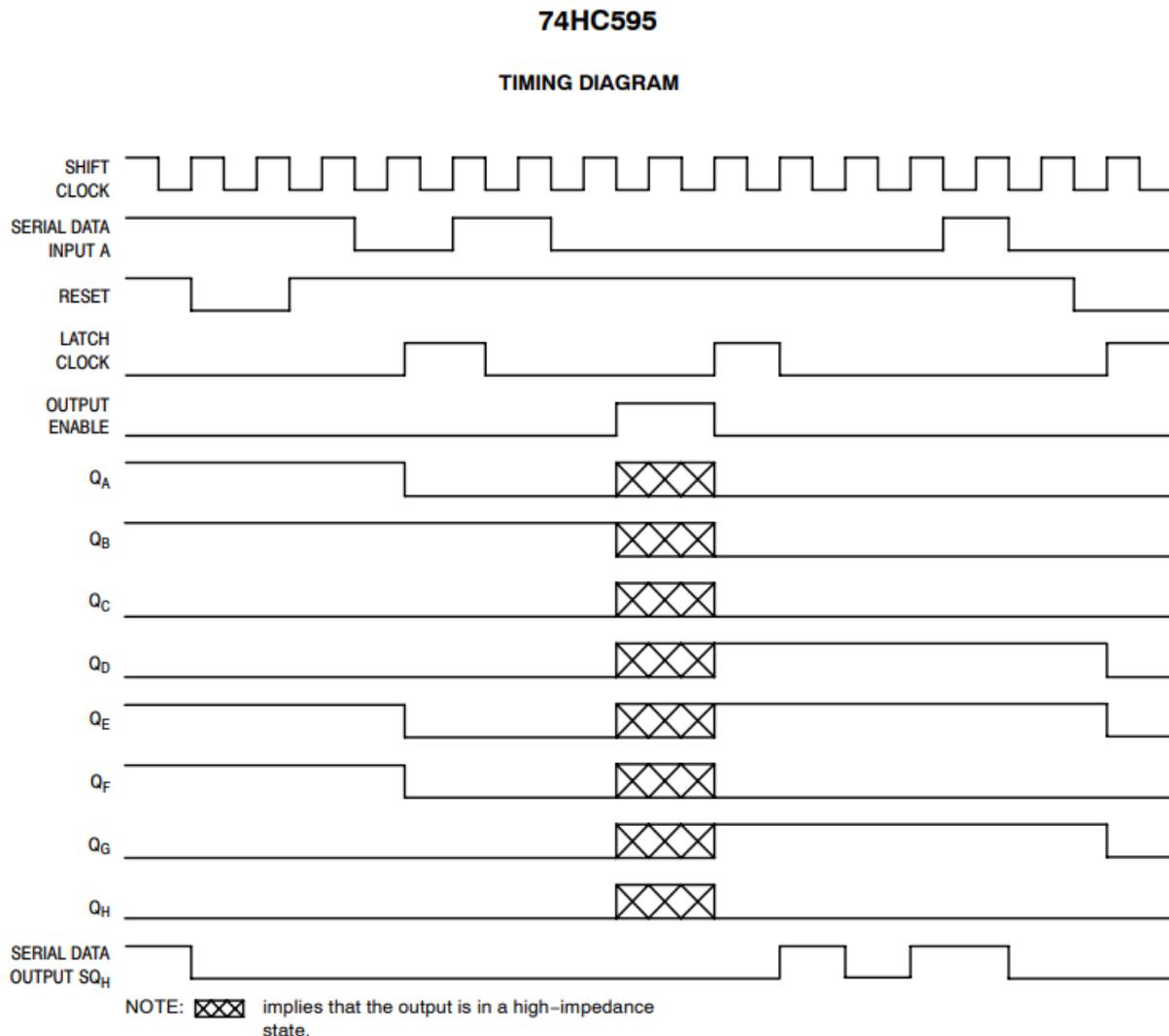


Figura 13: Diagrama de tiempos del 74HC595

Para transmitir una columna, se procede bit por bit, en forma serial, como se indica:

- Se coloca el 1er bit de la columna en la salida Data (M15)
- Se envía el pulso de reloj por la salida Clock (R15) lo que cargará el bit en la primera posición del registro de entrada del 74HC595. No pasará al registro de salida hasta que no se active la salida Latch (P15)
- Se coloca el siguiente bit en la salida Data
- Se envía un nuevo pulso de Clock, lo que producirá un desplazamiento (shift) en el registro de entrada del 74HC595, dando lugar al nuevo bit y corriendo hacia la derecha el o los bits anteriores.
- Se continúa así hasta cargar lo 8 bits de la matriz numérica, (Recordar que aunque se usan 7 elementos, es necesario enviar 8 y que el 5º no será tenido en cuenta).
- Una vez lleno el registro de entrada, se envía un pulso por la salida de Latch, lo que enviará todo el registro cargado a las salidas QA-QH del 74HC595. Al mismo tiempo, se activa la salida conectada al transistor de la columna que se acaba de transmitir, y se desactivan los otros cuatro. Esto producirá que se enciendan sólo

los LEDs de esa columna que tienen un “1” en su posición (fila). Los LEDs que tengan un “0” y todos los de las otras cuatro columnas permanecerán apagados.

Transcurrido un tiempo T, se repite el proceso con la columna siguiente, es decir, se carga serialmente una nueva columna en el 74HC595, se desactiva el transistor anterior y se activa el siguiente. Al llegar a la quinta columna, se comienza nuevamente con la primera. La duración del tiempo T establecerá el período, y por lo tanto la Frecuencia, de multiplexado. Si bien con algunos centenares de Hertz es suficiente para el ojo desnudo, conviene establecer una frecuencia alta, para evitar que fotografías o tomas de video del display generen un efecto visual indeseado. En los ejemplos se usa una frecuencia de refresco de columna de 60 kHz

Conectores Pmod

La interfaz Pmod (*Peripheral Module Interface*) es un standard definido por la compañía Digilent, en una especificación publicada en 2011⁴, para conectar módulos periféricos a placas de desarrollo FPGA y microcontroladores. Se trata de una interfaz simple, de 6 líneas (4 de señal y 2 de alimentación) y existen en el mercado gran variedad de módulos que la utilizan⁵. Desde un simple pulsador hasta interfaces de red, displays gráficos y diferentes clases de sensores digitales y analógicos, por citar sólo algunos. Las interfaces pueden utilizar las líneas como señales digitales independientes, tanto de entrada como de salida, o comunicarse mediante diferentes protocolos, tales como SPI, I²C o UART.

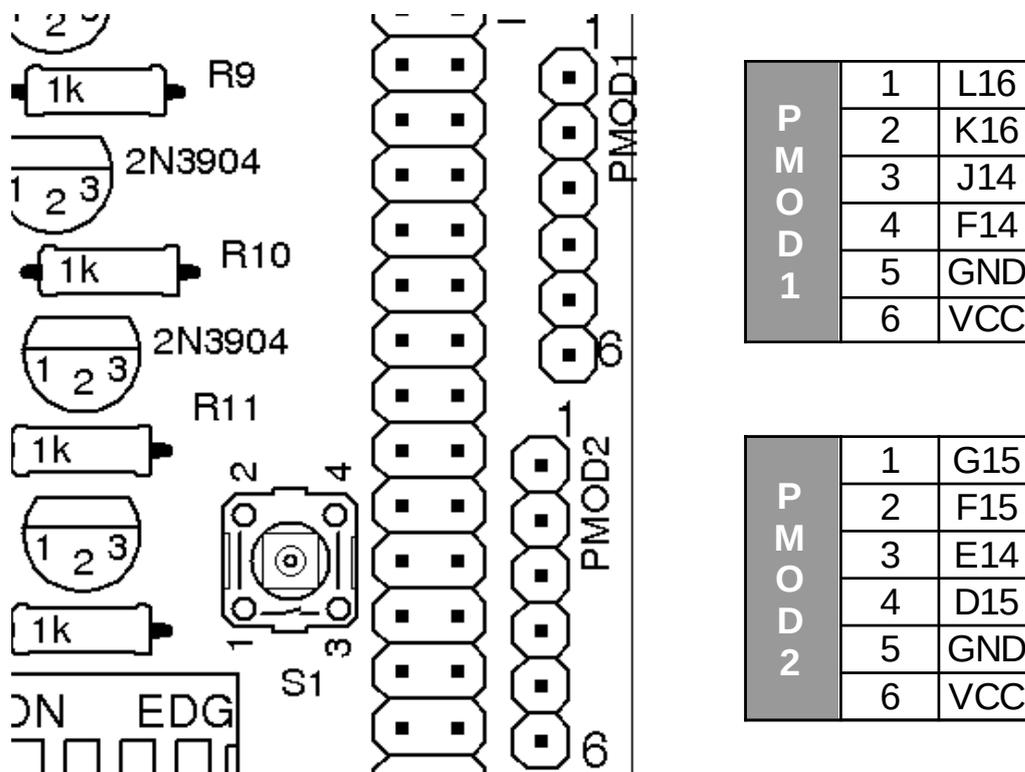


Figura 14: Conectores Pmod de la Visera, con identificación numérica de los pines 1 a 6 de cada conector y tablas de líneas I/O de la iCE40 asignadas a cada uno de esos pines.

4 http://digilentinc.com/Pmods/Digilent-Pmod_%20Interface_Specification.pdf

5 <https://youtu.be/q8psdwG15Qs>

La Visera ofrece dos conectores (Figura 14) compatibles con la interfaz Pmod, aunque no la implementan rigurosamente. Se trata de dos conectores independientes, es decir que cada uno cuenta con sus propias líneas de alimentación, lo que no debe confundirse con un conector Pmod doble, que tiene 10 pines (8 de datos y 2 de alimentación). De todos modos, resulta sencillo confeccionar un cable que adapte los dos conectores individuales a uno doble.

Consideraciones de compatibilidad

Eléctricamente, debe tenerse en cuenta que, si la placa Visera se alimenta con 5V en lugar de 3.3V, la tensión disponible en el pin 5 de los conectores Pmod será esa (5V) y se deberán tomar las precauciones adecuadas para adaptar los niveles de tensión conforme a las convenciones lógicas LVCMOS 3.3V o LVTTL 3.3V requeridas por el módulo utilizado. Las mismas precauciones se deben tomar si el módulo, aunque sea compatible con tensiones mayores, devuelve esa tensión hacia las líneas de datos del conector Pmod de la Visera, que sólo soportan 3.3V.

Con respecto a las resistencias de pull-up requeridas para interfaces I²C y similares, debe tenerse en cuenta que la placa Visera no las integra. Usualmente los módulos externos Pmod las traen incorporadas y, de ser necesario, se pueden habilitar las resistencias de pull-up internas de la iCE40. En el caso de las entradas predeterminadas de la Visera, detalladas en la Tabla 1, es **imprescindible** habilitar dichas resistencias internas, o los controles no funcionarán. El ejemplo para probar las entradas muestra el simple procedimiento necesario para habilitarlas.

La ventaja de esto es que permite alimentar la Visera con un voltaje superior al que soportan las entradas de la iCE40, dado que nunca se enviará esa tensión a sus entradas. Cuando se activa un switch, llave o encoder de la visera, se cierra un contacto a masa. Cuando se desactiva, la Visera deja esa conexión en estado flotante (sin pull-up) y será la resistencia interna de la ICE 40 la que llevará esa tensión a la correspondiente al 1 lógico. La lógica resultante será entonces de tipo inverso.

Esta configuración, sin elementos externos conectados a +V, permite además que, dejando la Visera sin alimentar, las entradas queden liberadas y sin interferencias para su uso mediante el conector principal de la Visera, que replica las conexiones del conector J2 de la Breakout Board. No obstante, si se va a usar la Breakout Board sin alimentar la Visera y no se quiere desmontar la misma, es conveniente asegurarse que el circuito sintetizado contenido en la flash no cargue registros en el 74HC595 y active simultáneamente los transistores que comandan las columnas, cosa que sucede si el último circuito sintetizado generaba un patrón en el display LED. Esto provocaría que el display se encienda (débilmente, pero lo hace) drenando la corriente para los LEDs de la placa principal, lo que podría representar una carga excesiva para la misma, si la situación se mantiene por un cierto tiempo. Si se sospecha que el último circuito sintetizado usaba el display, conviene alimentar la Visera, sintetizar un nuevo circuito que no active los LEDs de la Visera y luego retirar la alimentación y continuar trabajando sin riesgo de sobrecarga. En las pruebas realizadas, el display fue encendido varias veces con la Visera sin alimentar y esto no provocó daño alguno, pero es conveniente tomar las precauciones mencionadas.

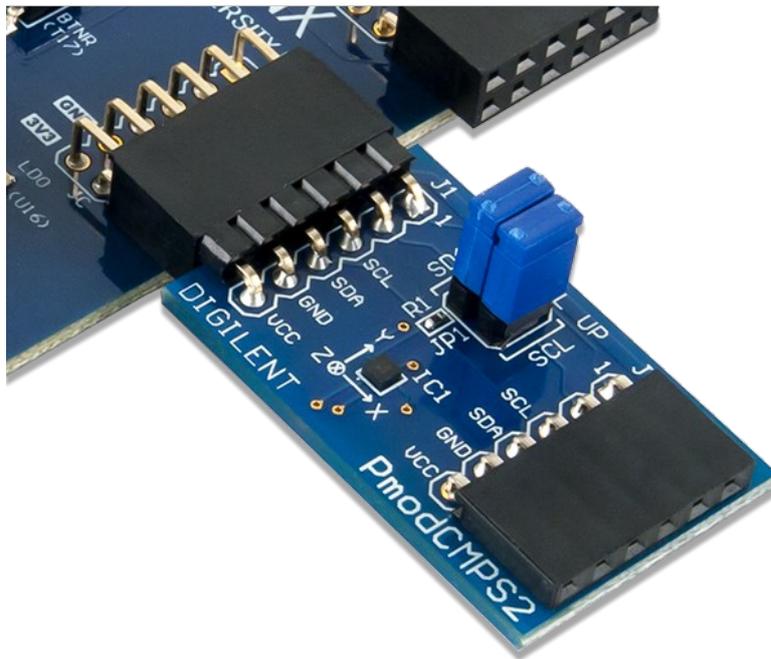


Figura 15: Ejemplo de un módulo Pmod insertado en un conector sin necesidad de cable

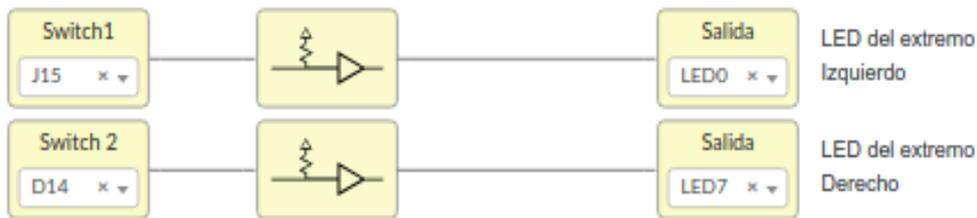
Otra consideración menor respecto a la compatibilidad se refiere al tamaño de los módulos Pmod que se pueden montar directamente en los conectores PMOD1 y 2 de la placa Visera sin el uso de cables. El standard sugiere el uso de conectores hembra, de paso 0.1", en ángulo recto, alineados al borde de la placa y espaciados 0.9" de centro a centro, para permitir la inserción directa de módulos Pmod de 0.8" de ancho, como el de la Figura 15, sin necesidad de cables. La placa de circuito impreso diseñada provee dos de esos conectores, que en el prototipo se montaron con pines rectos, para usar con cables, pero que perfectamente pueden poblarse con los *headers* hembra de 90° recomendados por la norma. Sólo debe tenerse en cuenta que la separación actual, entre el centro de los dos conectores del PCB es de 17,58 mm y no de 22.86 mm como especifica el standard. Es decir que, si se desea conectar simultáneamente dos módulos anchos (0.8" o 20.3 mm) podría haber alguna dificultad para insertarlos o directamente no caber los dos. En ese caso, lo conveniente es montar los conectores hembra con algunos mm de diferencia en su altura o conectar uno de los módulos mediante un cable de 6 pines.

Pruebas y ejemplos

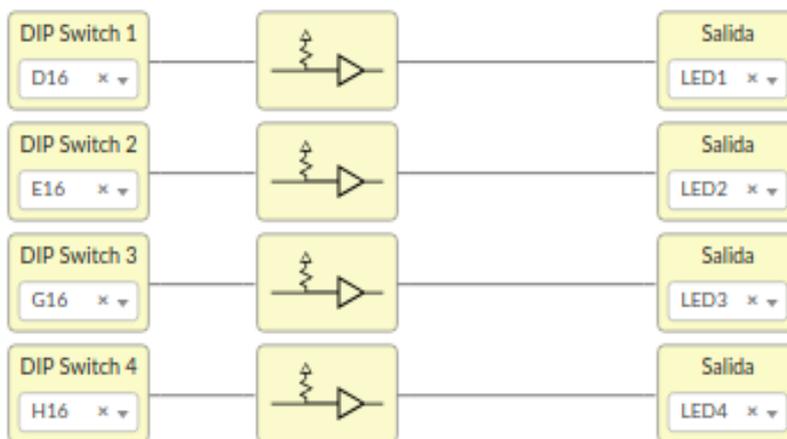
Antes de utilizar la placa para realizar cualquier ejercicio avanzado, es conveniente comprobar el correcto funcionamiento del hardware de la Visera que se haya montado. La forma más simple para esto es sintetizar los ejemplos que se suministran en el apartado de Enlaces, al final de este documento.

A medida que se vayan desarrollando y probando nuevos ejercicios se irán subiendo al website especificado en ese mismo apartado.

Conecta los dos tact-switch de la visera a las entradas correspondientes (J15 y D14),habilita las respectivas resistencias de pull-up para detectar cuando las teclas se pulsán (conectan a masa) y apaga los leds 0 y 7 cuando eso sucede.



Conecta los cuatro DIP switch de la visera a las entradas correspondientes (D16, E16,G16 y H16), habilita las respectivas resistencias de pull-up para detectar cuando las teclas se pulsán (conectan a masa) y apaga los leds 1 2 3 y 4 cuando eso sucede.



Conecta los dos pines del encoder rotativo a las entradas correspondientes (C16 y B16), habilita las respectivas resistencias de pull-up para detectar cuando hay pulsos (conectan a masa) y apaga los leds 5 y 6 en coincidencia con esos pulsos.

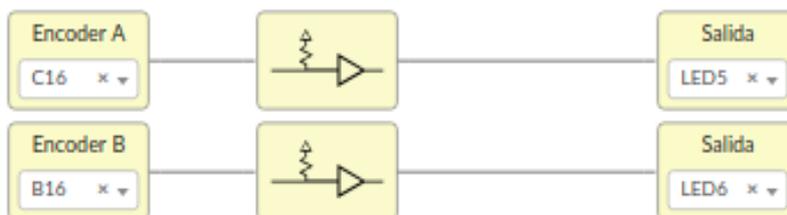


Figura 16: Diagrama de bloques "ice" del ejemplo para comprobar entradas de la Visera

Los ejemplos fueron desarrollados en Icestudio y los archivos para descarga incluyen sus versiones exportadas a Verilog, para que puedan ser sintetizados por quienes usen otro *toolchain* y no deseen instalar Icestudio, herramienta que está disponible para Linux, Windows y Mac y que recomendamos enfáticamente, sobre todo para un inicio rápido en el campo de las FPGA.

El ejemplo `DosTeclas_4Llaves_1Encoder[.ice .v]` sirve para probar todos los inputs, actuando sobre los LEDs originales de la Breakout Board, mientras que el ejemplo `Matriz_LED[.ice .v]` genera las letras ML (por Matriz Led) en el display y presenta una forma simple de cargar otras matrices de "1s" y "0s" para generar cualquier patrón

diseado, con la metodología presentada en los apartados anteriores y en la Figura 11. Los diagramas de bloques correspondientes a ambos ejemplos se presentan en las Figuras 16 y 17 respectivamente.

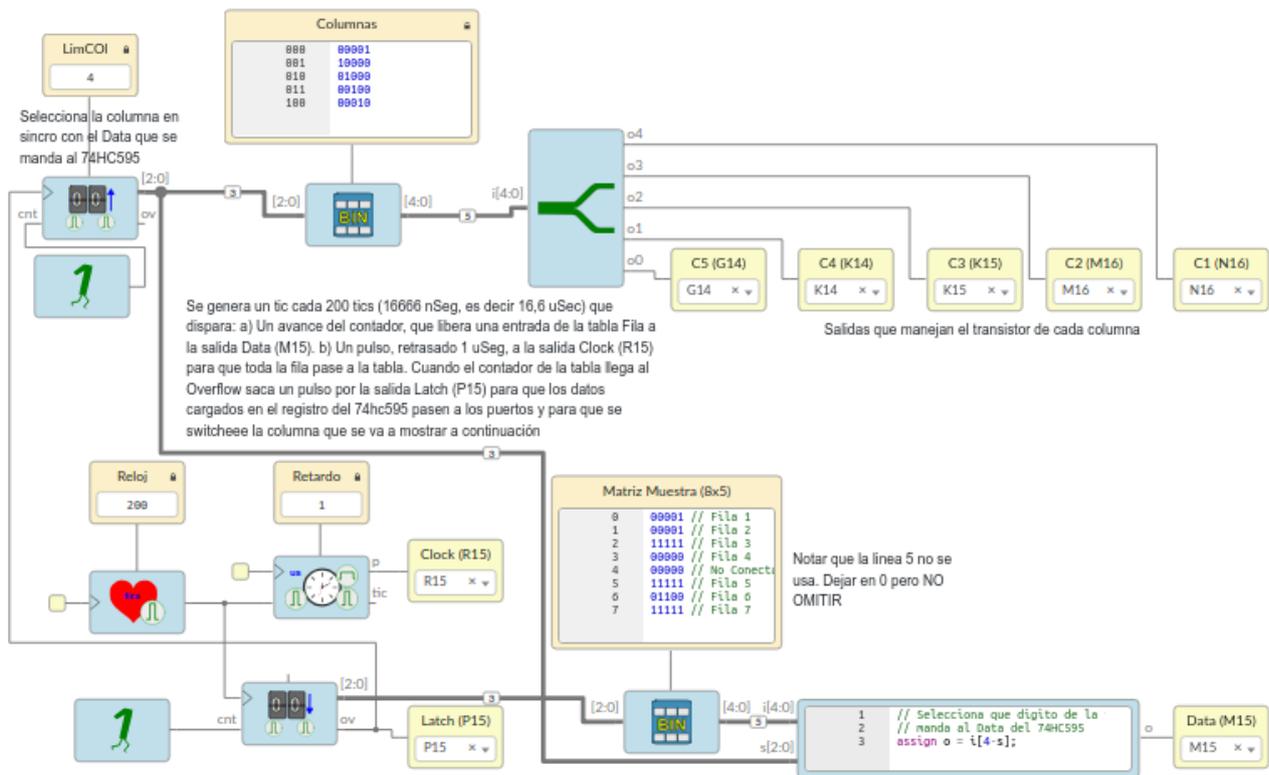


Figura 17: Diagrama de bloques "ice" del ejemplo para cargar matrices de "1s" y "0s" al display de 7x5 LEDs

Resultado final y planes futuros

Luego de algunos días de uso, sintetizando circuitos y operando con ellos, se ha podido establecer que el prototipo de la Visera 1.0 *Quarantine* funciona exactamente como estaba previsto, resultando una herramienta de suma utilidad para fines educativos. La Figura 18 muestra la placa instalada en la Breakout Board y todo el conjunto montado en el gabinete diseñado.

A lo largo de estos ejercicios, se han identificado algunas pequeñas mejoras que se enumeran a continuación para que sean tenidas en cuenta por quienes deseen construir su propio prototipo, o para una futura edición, desarrollada con mayor disponibilidad de recursos (*Visera 1.1 – New normality?*)

Las posibles mejoras son:

- Re-posicionamiento de los pulsadores, dip-switchs y encoder rotativo, de modo que se facilite la operación de esos controles. Se trasladaría el Switch 1 al borde inferior, a la izquierda del Switch 2, y se intercambiaría la posición entre el encoder rotativo y el array de DIP switchs, de modo que todos resulten más accesibles para la mano.

- Ligero desplazamiento de los conectores Pmod (un total de 5.28 mm) para aumentar su distancia entre centros hasta los 22.86 mm que recomienda la norma y poder dar cabida a dos módulos Pmod externos de 0.8" de ancho simultáneamente, sin necesidad de cables.
- Reemplazo de los dip-switchs comunes por otros, del tipo "piano", más fáciles de accionar sin herramientas, sobre todo si se traslada al borde inferior como está previsto.
- Vincular a las salidas P16 y H14 de la FPGA las entradas *Reset* (pin 10 del IC) y *Output Enable* (pin 13 del IC) del 74HC595, actualmente conectadas directamente a VCC y GND, respectivamente, para lograr mayor flexibilidad en el uso del *shift register*, y la posibilidad de poner las salidas Q1-Q8 en estado de alta impedancia.

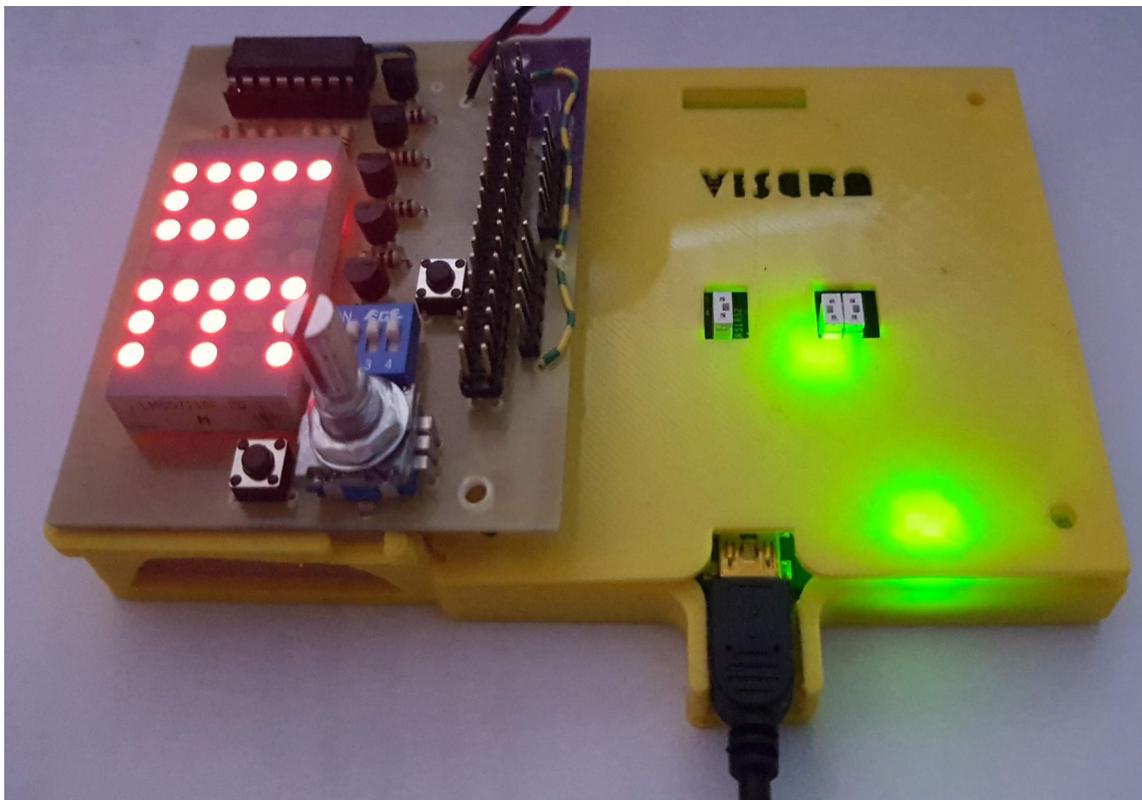


Figura 18: Vista del prototipo montado y conjunto alojado en el gabinete diseñado

Website y archivos para descargar

Website: miguelgrassi.com.ar/visera

Repositorio Git con documentación actualizada, ejemplos para sintetizar en Icestudio y Verilog, archivos de esquemático y PCB, (Eagle V9.03), Modelos stl para imprimir 3D gabinete y tapa, colección de bloques para Icestudio, etc.: <https://github.com/migrassi/Visera>

Herramienta para generación de caracteres y símbolos personalizados para usar en el display Matriz de Leds: <https://editor.p5js.org/migrassi/present/nZ4poi31X>

Enlaces externos:

Icestudio: icestudio.io

Editor visual *opensource* para placas FPGA abiertas basado en el proyecto Icestorm y Apio. Soporta las más comunes placas de desarrollo Lattice iCE40, incluyendo iCE40-HX8K Breakout Board, iCEstick y Kéfir I, entre muchas otras.

FPGAwards: fpgawars.github.io

Comunidad de desarrollo relacionada con icestudio. Contiene abundante información y un [foro](#) muy activo en español. Desarrollaron la plataforma de hardware libre [Alhambra](#)

FPGA para todos: gitlab.com/RamadrianG/wiki---fpga-para-todos/-wikis/FPGA-para-Todos

Proyecto desarrollado por un grupo de la Universidad Tecnológica Nacional (UTN) Regional Haedo con el apoyo de grupos de otras universidades argentinas. Mantienen una Wiki con mucho material, tutoriales de VHDL y Verilog y desarrollaron la plataforma de hardware libre [EDU-FPGA](#)

FPGA Libre: <http://fpgalibre.sourceforge.net/>

Grupo desarrollado en la Unidad Técnica Instrumentación y Control perteneciente al centro Electrónica e Informática del [INTI \(Instituto Nacional de Tecnología Industrial\)](#). Tiene gran cantidad de contenido valioso, muchas herramientas y cores libres. El grupo desarrolló la placa [Kéfir I](#), basada en iCE40HX4K y soportada por Icestudio .



El presente documento fue escrito e ilustrado totalmente utilizando software libre: GIMP 2.8, Inkscape 0.92 y Libre Office 6.3, en una computadora con Linux Mint 18.8 Cinnamon. El desarrollo del trabajo también fue realizado en su gran mayoría con herramientas libres, a excepción de Eagle 9.0 y Autodesk Inventor 2019, de los cuales se utilizaron versiones gratuitas, bajo licencias educativas.

Miguel Grassi – 2020

[\(migrassi@gmail.com\)](mailto:migrassi@gmail.com)



Material publicado bajo licencia Creative Commons BY NC. Permitida la distribución, copia y modificación libre y gratuita, con fines no comerciales y atribución al autor.