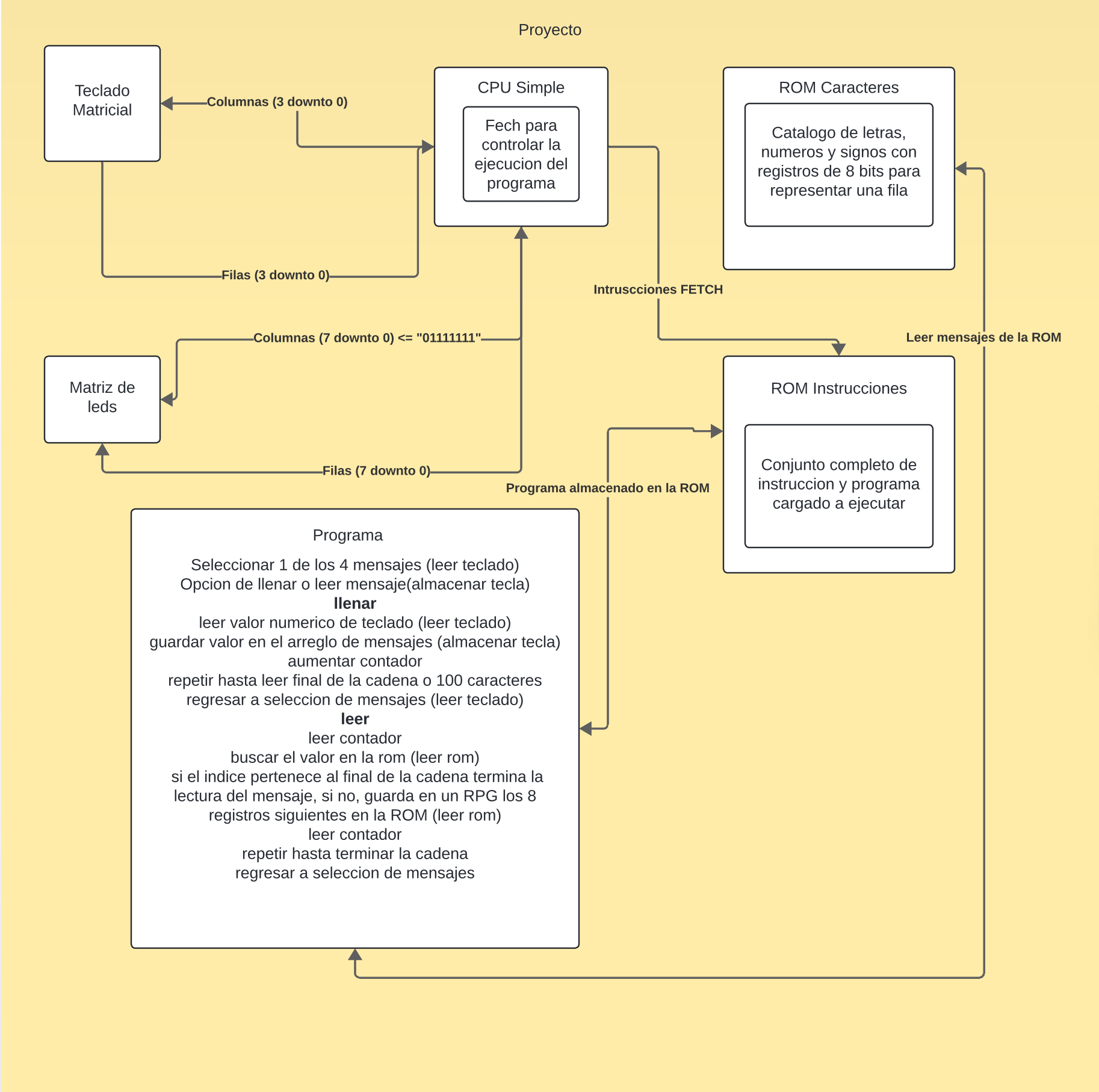
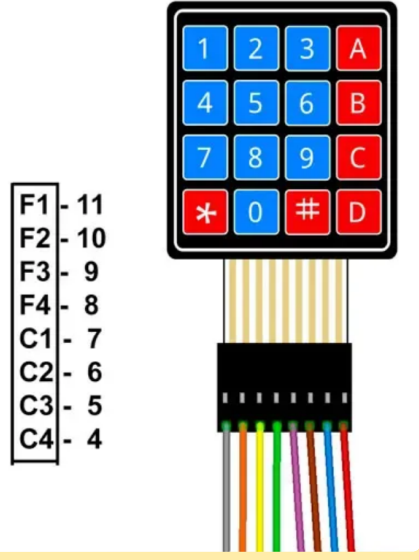


- Actividades a Realizar
- 1.- Cambiar de leer un catacter a lectura de combinaciones con el teclado
  - 2.- Cambiar la visualizacion de una matriz a 3 matrices
  - 3.- Generar la rom con la representacion de los registros de cada letra, numero y caracter
  - 4.- Crear el espacio de almacenamiento de los 4 mensajes y el mensaje que se estara mostrando en ese momento
  - 5.- Crear las instrucciones para el programa con sus ciclos y sus instrucciones jump
  - 6.- Juntar el cpu simple que ya se tiene con la nueva interfaz y el nuevo set de instrucciones



Teclado Matricial



Para leer el teclado matricial es necesario contar con:

- Un reloj con una frecuencia suficientemente rapido como para encender las 4 columnas casi al mismo tiempo
- Dos vectores de bits, uno de entrada de las filas y uno de la salida de las columnas
- El vector de las columnas se enciende con 1 = "1000" el cual con cada ciclo de reloj hara un corrimiento a la izquierda = "0001", "0010", "0100"

Para lograr obtener una combinacion de teclas presionadas podemos tomar el momento cuando se seleccione una tecla y despues activar un contador! lo sufientemente duradero como para que el usuario retire su dedo y presione otra tecla

Matriz de leds

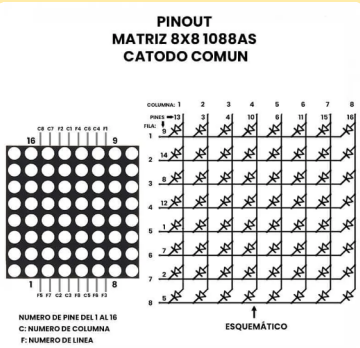
El funcionamiento de la matriz de leds que vamos a utilizar se basa en lo siguiente:

- Para encender una posicion necesitamos mandar un 0 a la columna que vamos a encender y un arreglo de 8 bits con las filas que vamos a encender

Para encender una matriz necesitamos:

- Un vector de 8 bits que representa a las columnas inicializado <= "01111111" el cual va a contar con un corrimiento a la izquierda para encender una por una
- Un vector de 8 bits el cual representa las filas, estas se encienden mandando un '1' logico

Si queremos agregar dos matrices solo necesitamos agregar 8 bits mas de las columnas, es decir <= "0111111111111111", con esto y un corrimiento vamos a encender cada columna con el mismo vector de 7 bits como fila



ROM Caracteres

Para almacenar y leer las letras que se van a mostrar necesitamos contar con una ROM con todos las filas de las letras

Cada letra que vamos a mostrar en la matriz necesita contar con 8 vectores uno por cada fila de 8 bits, por lo que las letras son indexadas de 8 en 8 en la ROM

- A va de 0 a 7
- B va de 8 a 15
- y asi sucesivamente

Para poder leer las letras necesitamos contar con:

- Un reloj que vaya a la misma frecuencia que se encienden las columnas de la o las matrices
- Una señal llamada "row\_counter" el cual va a ir de 0 a 7, al cual con cada ciclo del reloj mencionado va a sumar 1
- Para buscar la letra solo necesitamos el valor del comienzo es decir, para leer la A solo necesitamos leer un 0, porque con el row\_counter podemos mandar de direccion a la ROM el indice + row\_counter

El componente de la ROM tendra como entrada un entero con la direccion de la fila que vayamos a leer y un vector de 8 bits 7 downto 0 de los bits que se van a encender en la fila de la matriz

ROM Instrucciones

Para el control del CPU necesitamos una ROM con las instrucciones para ejecutar el programa las cuales serian:

- leer tecla (usamos el componente del teclado matricial para obtener la combinacion de teclas presionadas, este valor debe ser un entero)
- almacenar tecla en un registro de proposito general (obtener el valor entero del teclado y almacenarlo en algun registro de proposito general)
- Aumentar contador(cada que entre una nueva tecla aumentamos en 1 el contador almacenado en un registro de proposito general, asi podemos saber el tamaño final del mensaje almacenado)
- leer rom (usando el del indice guardado al presionar una combinacion de teclas en el registro de proposito general vamos a mandarlo como direccion usando row\_counter y asi obtener la fila de la letra a leer)
- mostrarRom (cada que leemos una columna vamos a mandarla a la salida que se muestra en la matriz de leds)

Registros de Proposito General

Solo necesitamos dos registros de proposito general:

- Uno para el mensaje almacenado el cual contiene un arreglo de 100 numeros enteros
- Uno con el contador del tamaño del mensaje almacenado