
INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

Departamento de Electrónica, Sistemas e Informática

Sistemas Embebidos Basados en Microcontroladores II

Correo electrónico:

ie714715@iteso.mxIe714415@iteso.mx**Nombre de la carrera****Ingeniería Electrónica****ITESO**
Universidad Jesuita
de Guadalajara**Práctica 3:**

Implementación de HDI driver en un solo Endpoint

Integrantes del equipo:

Yadira Vanessa Bonifacio Benavides

Sofía Michel Salazar Valdovinos

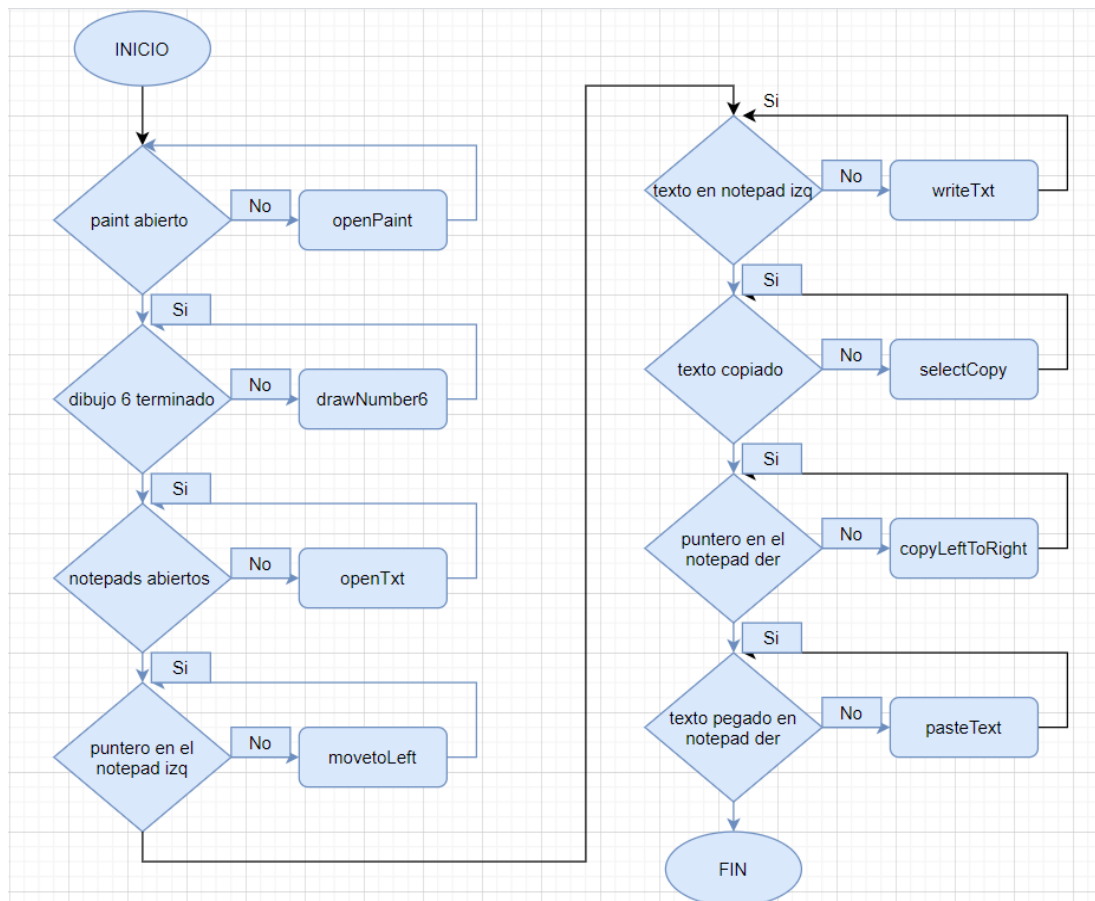
Profesor: Edgardo Serna Enciso

Tlaquepaque, Jalisco, 6, noviembre, 2020

Objetivo: En esta práctica construiras un sistema en la FRDM-K66F que al conectarse por el puerto de USB K66 de la tarjeta a la computadora realice la siguiente secuencia de pasos:

- Se abre alguna herramienta de dibujo: Paint en Windows o Pinta en Ubuntu (puedes ver las instrucciones para descargarlo al final del documento).
- El mouse se mueve a un lugar en el lienzo de dibujo, hace click y con el click presionado se mueve para dibujar el número de su equipo. Finalmente se libera el click.
- Se abre una aplicación de bloc de notas (notepad en windows o gedit en ubuntu).
- La ventana se coloca a la izquierda ocupando la mitad de la pantalla. Esto se puede hacer con WINDOWS+FLECHA IZQUIERDA (en Windows) o WINDOWS+CTRL+FLECHA IZQUIERDA (en Ubuntu).
- Se abre otra aplicación de bloc de notas y se coloca a la derecha.
- El curso se mueve para hacer click sobre la primera ventana (la de la izquierda).
- Se escribe un mensaje como “Hola mundo”.
- Se selecciona todo el texto en esta ventana y se copia al portapapeles.
- El cursor se mueve para llegar a la otra ventana de bloc de notas y se hace click para seleccionarla.
- El texto se pega a esta nueva ventana.

Descripción de la implementación: El algoritmo básico en el código con el cual se flasheo la k66 es el que se muestra en la siguiente imagen. Lo que se realiza para cumplir las instrucciones objetivo es realizado mediante una máquina de estados en la cual cada paso es una instrucción al completarse cada una de estas hace un incremento en la variable que nos permitirá cambiar de instrucción.



Para el cumplimiento de esta práctica las partes fundamentales fueron la unión en un descriptor tener dos reportes, el del teclado y el mouse para esto se unieron los descriptores individuales y se colocaron los report ID de cada uno para su correcto funcionamiento.

```

/* Mouse */
0x05U, 0x01U, /* Usage Page (Generic Desktop)*/ /* KEYBOARD */
0x09U, 0x02U, /* Usage (Mouse) */              0x05U, 0x01U, /* Usage Page (Generic Desktop)*/
0xA1U, 0x01U, /* Collection (Application) */    0x09U, 0x06U, /* Usage (Keyboard) */
        0x85U, 0x01U, /* Report ID (mouse) */    0xA1U, 0x01U, /* Collection (Application) */
0x09U, 0x01U, /* Usage (Pointer) */            0x85U, 0x02U, /* Report ID (Keyboard) */

```

De la misma manera para el funcionamiento se realizaron funciones para cada una de las acciones a realizar descritas en el objetivo, en estas funciones se mandaron al buffer de cada uno de los dispositivos las instrucciones a realizar; en el caso del mouse se mandaron los desplazamientos que tenía que realizar el puntero y en el caso del teclado se mandaron las teclas a escribir.

```

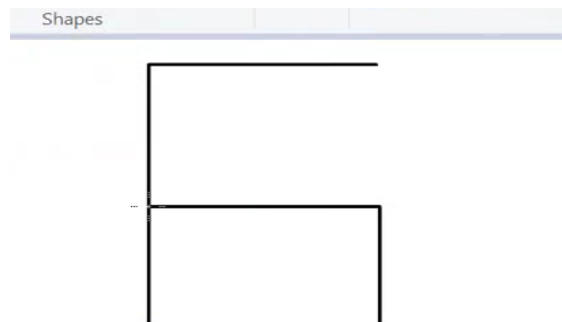
/* Move down. Increase Y value. */      /* Windows + left */
hid_buffer[1] = 1U;                      delay(2000000);
hid_buffer[2] = 0U;                      bufferKey[1] = MODIFIERKEYS_LEFT_GUI;
hid_buffer[3] = 2U;                      bufferKey[3] = KEY_LEFTARROW;
                                          delay(2000000);

```

Y por último fue indispensable la realización de una máquina de estados que como se observa en el diagrama verifica si ya se realizaron todos los pasos para la acción deseada de ser así pasa a la siguiente instrucción y esto se realiza de igual manera con cada una de las instrucciones desde abrir Paint para dibujar el número de nuestro equipo hasta abrir los dos Notepad para escribir, copiar y pegar el texto. Para el correcto funcionamiento de las funciones también fue indispensable un buen manejo de los delay ya que estos otorgaban tiempo para que se enviaran de forma correcta cada una de las instrucciones del teclado y mouse, de no hacerse esto se presentaban dos casos, el primero era la interrupción de la ejecución de las tareas y la segunda es el fallo del código que llevaba a un reinicio del sistema.

La implementación del sistema consta principalmente de los siguientes módulos:

- usb_device_descriptor: Aquí se realiza la unión de los reportes de los dispositivos necesarios para poder llevar a cabo las acciones funcionamiento
- mouse: aquí se realiza la máquina de estados para la ejecución de las tareas en el orden deseado.
- movements: aquí se realiza la implementación de las funciones que controlan el mouse y el teclado, en este se envían las coordenadas en el caso del mouse y el valor de las teclas en el caso del teclado.



Link:

https://github.com/sofmich/P3_Bonifacion_Salazar

Referencia:

Gilson. (2014). mouse + keyboard combo. Consultado el 4 de noviembre del 2020, de DevZone Sitio web:
<https://devzone.nordicsemi.com/f/nordic-q-a/4408/mouse-keyboard-combo>