

Universidad de Costa Rica

Facultad de Ingeniería

Escuela de Ingeniería Eléctrica

IE-0623 Microprocesadores

III ciclo 2020

Tarea 4

Luis Guillermo Ramírez - B76222

Roberto Sánchez Cárdenas - B77059

Profesor: Ing. Rafael Esteban Badilla Alvarado

Grupo 01

26 de enero del 2021

Índice

Índice de figuras	III
1. Estructuras de datos	1
2. Diagramas	2
2.1. Programa principal	2
2.2. Tarea Teclado	3
2.3. Mux teclado	4
2.4. Formar array	5
2.5. Interrupciones	6
3. Memoria de cálculo	6
4. Pruebas del código	7

Índice de figuras

1.	Diagrama de flujo de programa principal	2
2.	Diagrama de subrutina flujo de tarea teclado	3
3.	Diagrama de subrutina para leer teclado	4
4.	Diagrama de subrutina para formar array	5
5.	Diagrama de subrutina para interrupciones	6
6.	Prueba de llenado	7
7.	Prueba de borrado	7
8.	Prueba de enter	7
9.	Prueba de interrupción Keywakeup	7
10.	Prueba de interrupción Keywakeup con enter	7

1. Estructuras de datos

- MAX_TCL: Byte para indicar máxima cantidad de datos
- Tecla: Byte para guardar temporalmente el dato leído
- Tecla_IN: Byte que se usa para mover al array de datos
- Cont_Reb: Se usa para evitar rebotes
- Cont_TCL: byte para llevar tamaño actual del array
- Patron: Byte para guardar un patron que recorre el teclado
- Banderas: Byte para guardar 3 banderas en 3 bits
- Num_Array: Se reservan 6 bytes para guardar datos
- Teclas: Array de valores posibles del teclado

2. Diagramas

2.1. Programa principal

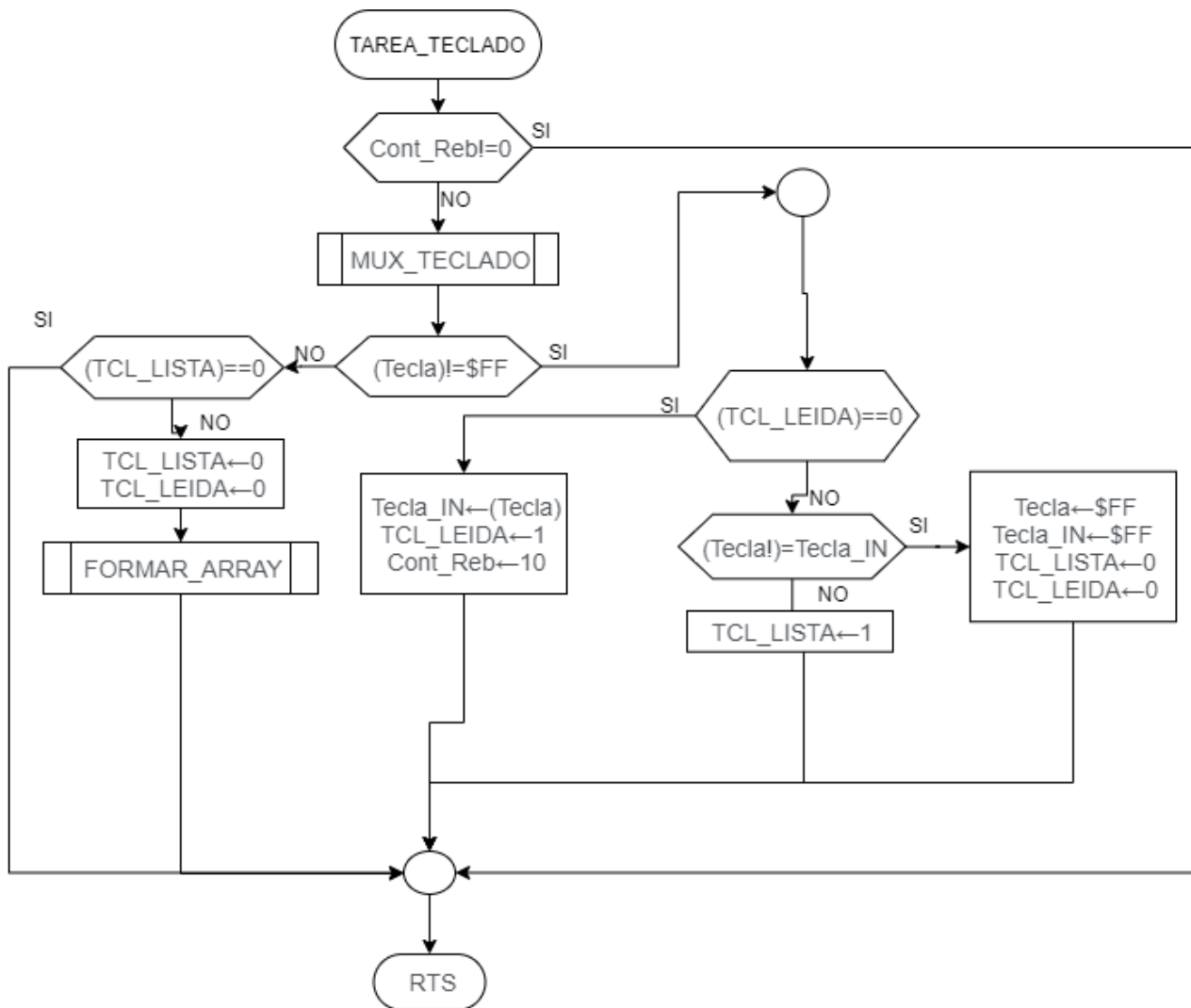


Figura 1: Diagrama de flujo de programa principal

2.2. Tarea Teclado

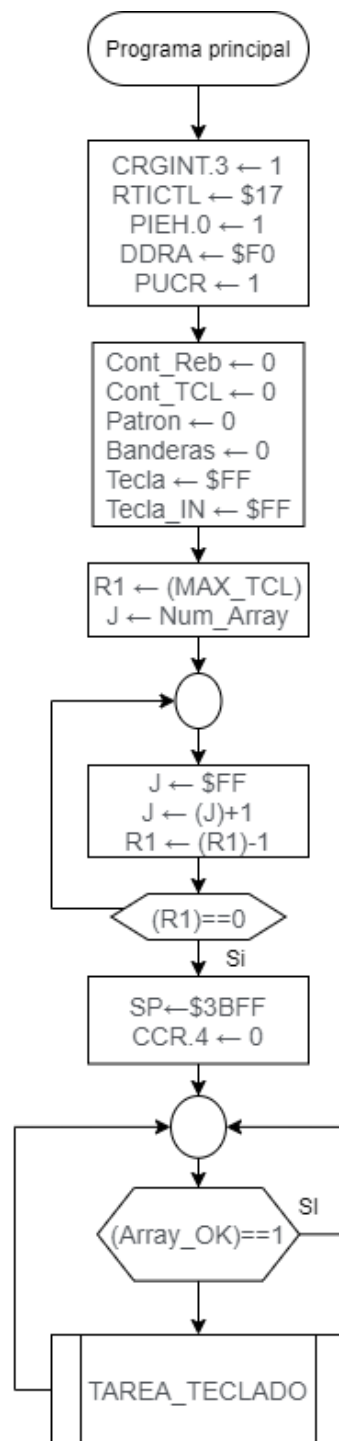


Figura 2: Diagrama de subrutina flujo de tarea teclado

2.3. Mux teclado

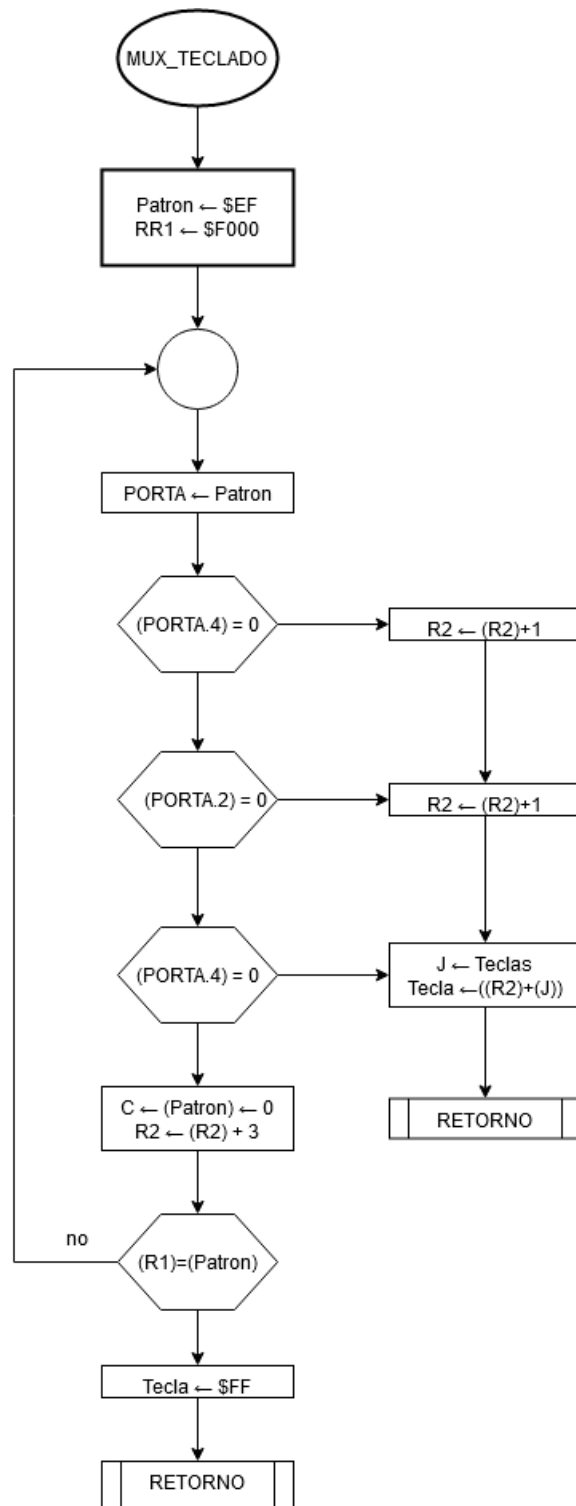


Figura 3: Diagrama de subrutina para leer teclado

2.4. Formar array

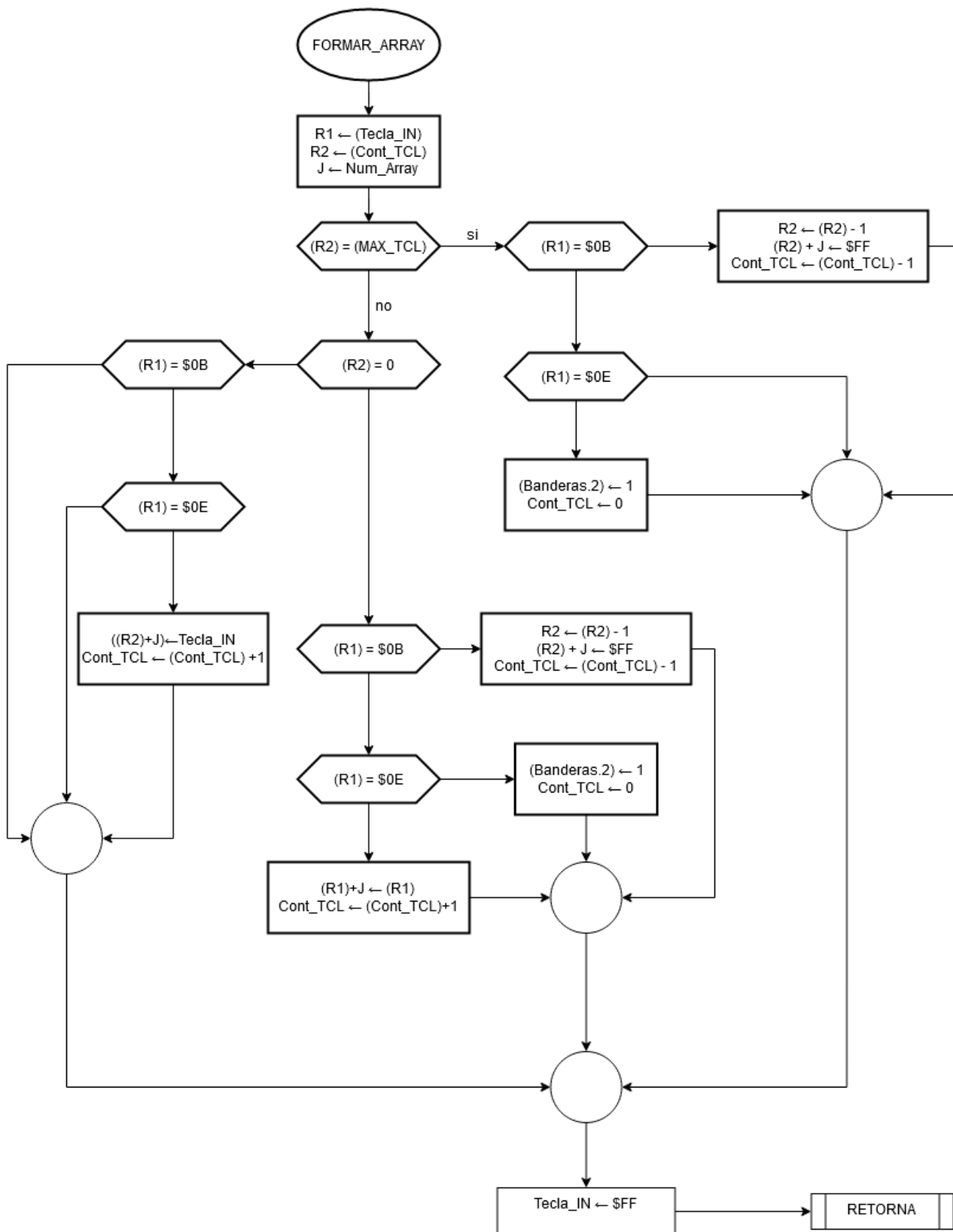


Figura 4: Diagrama de subrutina para formar array

2.5. Interrupciones

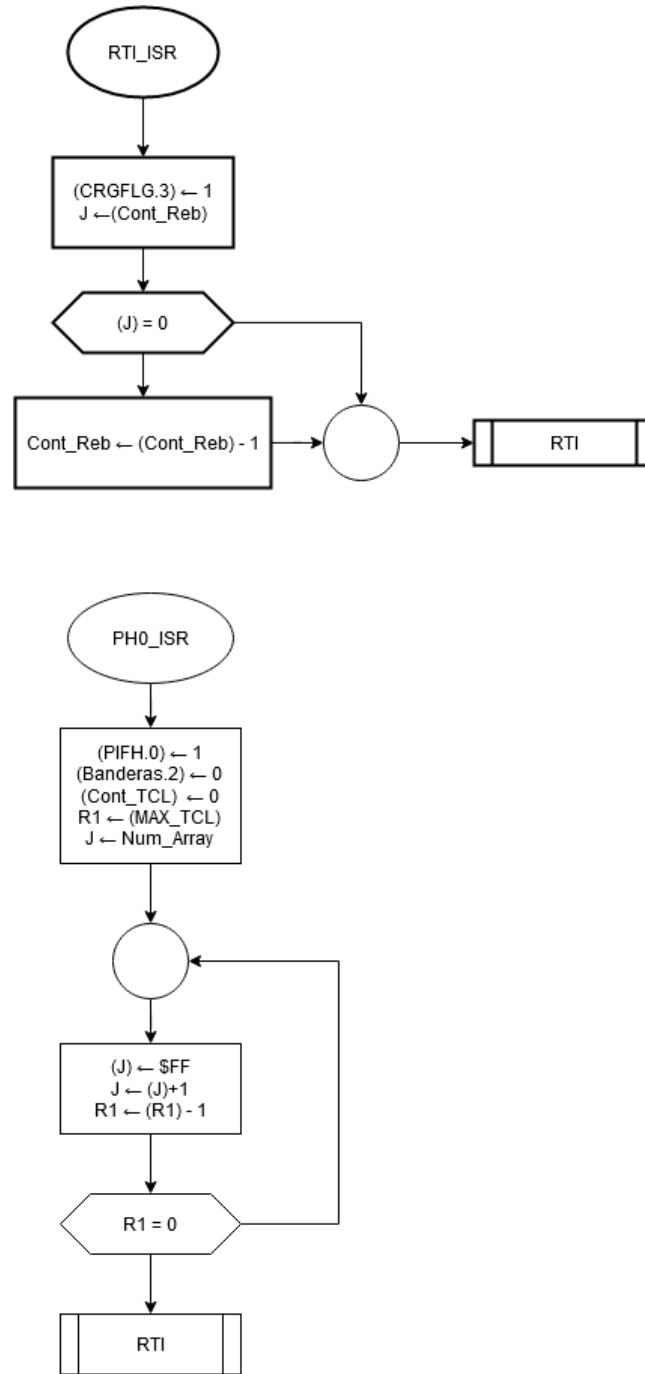


Figura 5: Diagrama de subrutina para interrupciones

3. Memoria de cálculo

Buscamos lograr un T_{RTI} de 1 ms. Nos basamos en la siguiente ecuación para realizar los cálculos.

$$T_{RTI} = \frac{(N + 1)2^{M+9}}{Osc_Clk} \quad (1)$$

Si $M=1$, entonces obtenemos el valor de N

$$N = \frac{T_{RTI} \cdot Osc_Clk}{2^{10}} \approx 7 \quad (2)$$

Al sustituir estos valores en 1 obtenemos 1024 ms.

4. Pruebas del código

Utilizando un MAX_TCL de 5 se realizaron las siguientes pruebas:

Primero se llena del 1 al 5, utilizando el teclado, donde se obtienen los resultados esperados en la figura 6

```
>md 1000

1000 05 FF FF 00 - 05 F0 00 01 - 02 03 04 05 - 2F 01 02 03
.
```

Figura 6: Prueba de llenado

Realizamos una prueba de borrado simple donde escribimos 1,2,3 y borramos. Seguidamente colocamos 4, 5, 6. El resultado se observa como lo esperado en la figura 7.

```
>md 1000

1000 05 FF FF 00 - 05 DE 00 01 - 02 04 05 06 - 2F 01 02 03
.
```

Figura 7: Prueba de borrado

Se realiza la prueba para el funcionamiento del enter. Se llena del 1 al 3 y se presiona enter, luego se presionan las teclas 4 y 5. El resultado es el patrón de 1,2,3 debido a que no debe tomar nada después de haber presionado el enter. 8

```
>md 1000

1000 05 FF FF 00 - 03 DE 00 01 - 02 03 FF FF - 2F 01 02 03
.
```

Figura 8: Prueba de enter

Se llena del 7 al 9, y se presiona el PH0. Seguidamente se llena con 4 y 5. Se espera que se borre todo lo anterior y que solo tome en cuenta lo nuevo como se observa en la figura 9.

```
>md 1000

1000 05 FF FF 00 - 02 DE 00 04 - 05 FF FF FF - 2F 01 02 03
.
```

Figura 9: Prueba de interrupción Keywakeup

Esta prueba es lo mismo que la anterior, donde se presionan las teclas 4,5,6 seguido del enter. Y luego se borra con el PH0 y se escribe 9 y 7. Vemos el resultado correcto en 10

```
>md 1000

1000 05 FF FF 00 - 02 EF 00 09 - 07 FF FF FF - 2F 01 02 03
.
```

Figura 10: Prueba de interrupción Keywakeup con enter