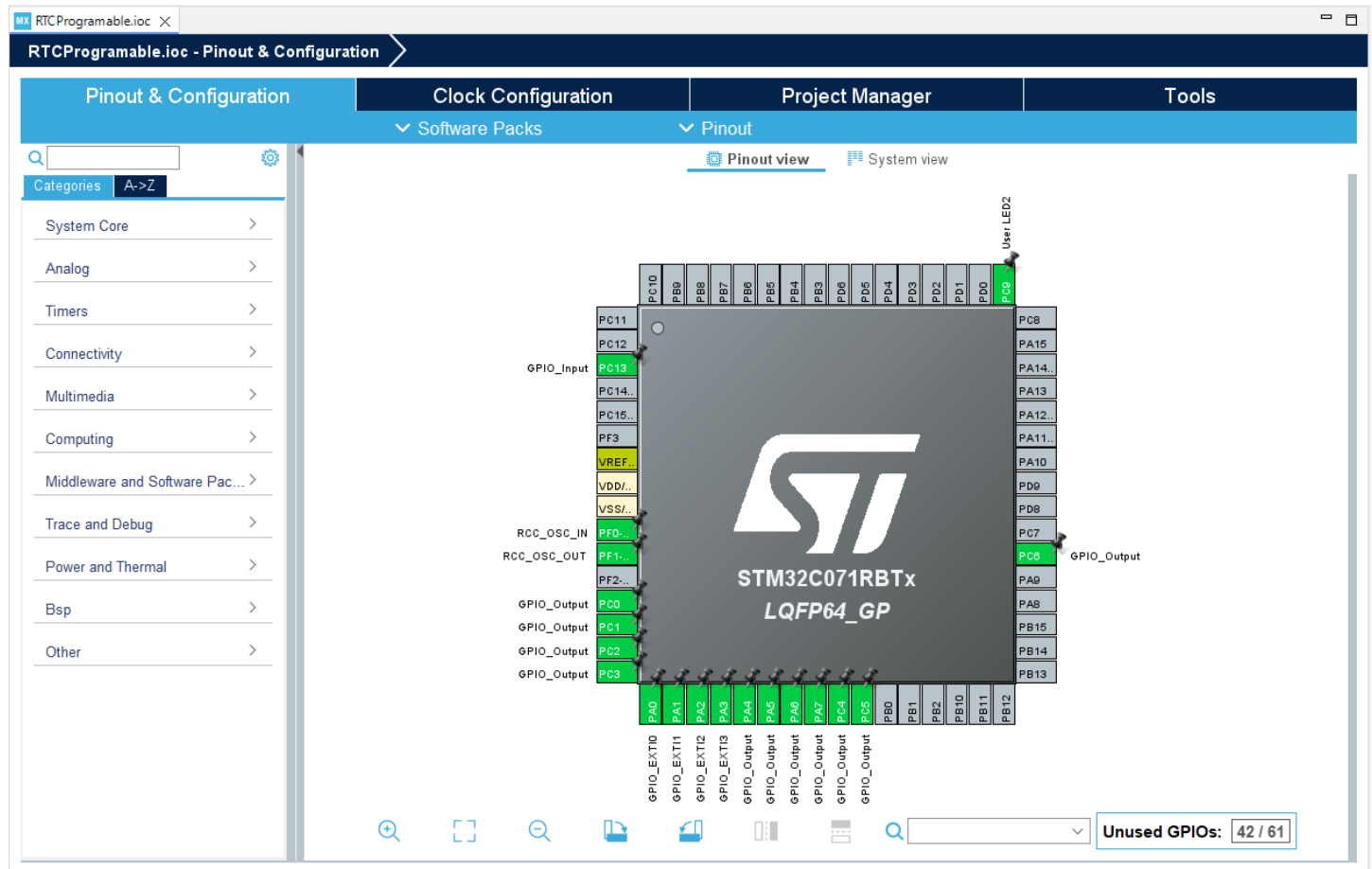


## CONFIGURACION INICIAL



## Pinout &amp; Configuration

## Clock Configuration

▼ Software Packs

▼ Pinout

Q [ ] ▼



Categories

A-&gt;Z

System Core &gt;

Analog &gt;

Timers &gt;

▲ RTC

TIM1

TIM2

TIM3

TIM14

TIM16

TIM17

Connectivity &gt;

Multimedia &gt;

Computing &gt;

Middleware and Softw... &gt;

Trace and Debug &gt;

Power and Thermal &gt;

Bsp &gt;

Other &gt;

## RTC Mode and Configuration

## Mode

☒ Activate Clock Source☒ Activate Calendar

Alarm A Disable ▼

☐ Timestamp

Calibration Disable ▼

☐ Reference clock detection

## Configuration

Reset Configuration

✔ Parameter Settings

✔ User Constants

Configure the below parameters :

Q Search (Ctrl+F)



## ▼ General

Hour Format Hourformat 24

Asynchronous Pr... 127

Synchronous Pre... 255

## ▼ Calendar Time

Data Format BCD data format

Hours 5

Minutes 0

Seconds 0

SubSeconds 0

Day Light Saving:... Daylightsaving None

Store Operation Storeoperation Reset

## ▼ Calendar Date



## Pinout &amp; Configuration

## Clock Configuration

v Software Packs

v Pinout

Q [ ] v



Categories A-&gt;Z

System Core v

CORTEX\_M0+

DMA

FLASH

✓ GPIO

IWDG

✓ NVIC

⚠ RCC

✓ SYS

WWDG

Analog &gt;

Timers &gt;

Connectivity &gt;

Multimedia &gt;

Computing &gt;

Middleware and Softw... &gt;

Trace and Debug &gt;

Power and Thermal &gt;

Bsp &gt;

GPIO Mode and Configuration

Mode

Configuration

Group By Peripherals v

✓ GPIO

✓ RCC

✓ NVIC

Search Signals

Search (Ctrl+F)

☐ Show only Modified Pins

P...	Sign...	GPI...	GPI...	GPI...	Maxi...	Fast ...	User...	Modi...
PA0	n/a	n/a	Exte...	Pull-...	n/a	n/a		✓
PA1	n/a	n/a	Exte...	Pull-...	n/a	n/a		✓
PA2	n/a	n/a	Exte...	Pull-...	n/a	n/a		✓
PA3	n/a	n/a	Exte...	Pull-...	n/a	n/a		✓
PA4	n/a	Low	Outp...	No p...	Low	n/a		<input type="checkbox"/>
PA5	n/a	Low	Outp...	No p...	Low	n/a		<input type="checkbox"/>
PA6	n/a	Low	Outp...	No p...	Low	n/a		<input type="checkbox"/>

 Select Pins from table to configure them. **Multiple selection is Al**

## Pinout &amp; Configuration

## Clock Configuration

v Software Packs

v Pinout

Categories

A-&gt;Z

System Core v



CORTEX\_M0+

DMA

FLASH

☒ GPIO

IWDG

☒ NVIC☒ RCC☒ SYS

WWDG

Analog &gt;

Timers &gt;

Connectivity &gt;

Multimedia &gt;

Computing &gt;

Middleware and Softw... &gt;

Trace and Debug &gt;

Power and Thermal &gt;

Bsp &gt;

GPIO Mode and Configuration

Mode

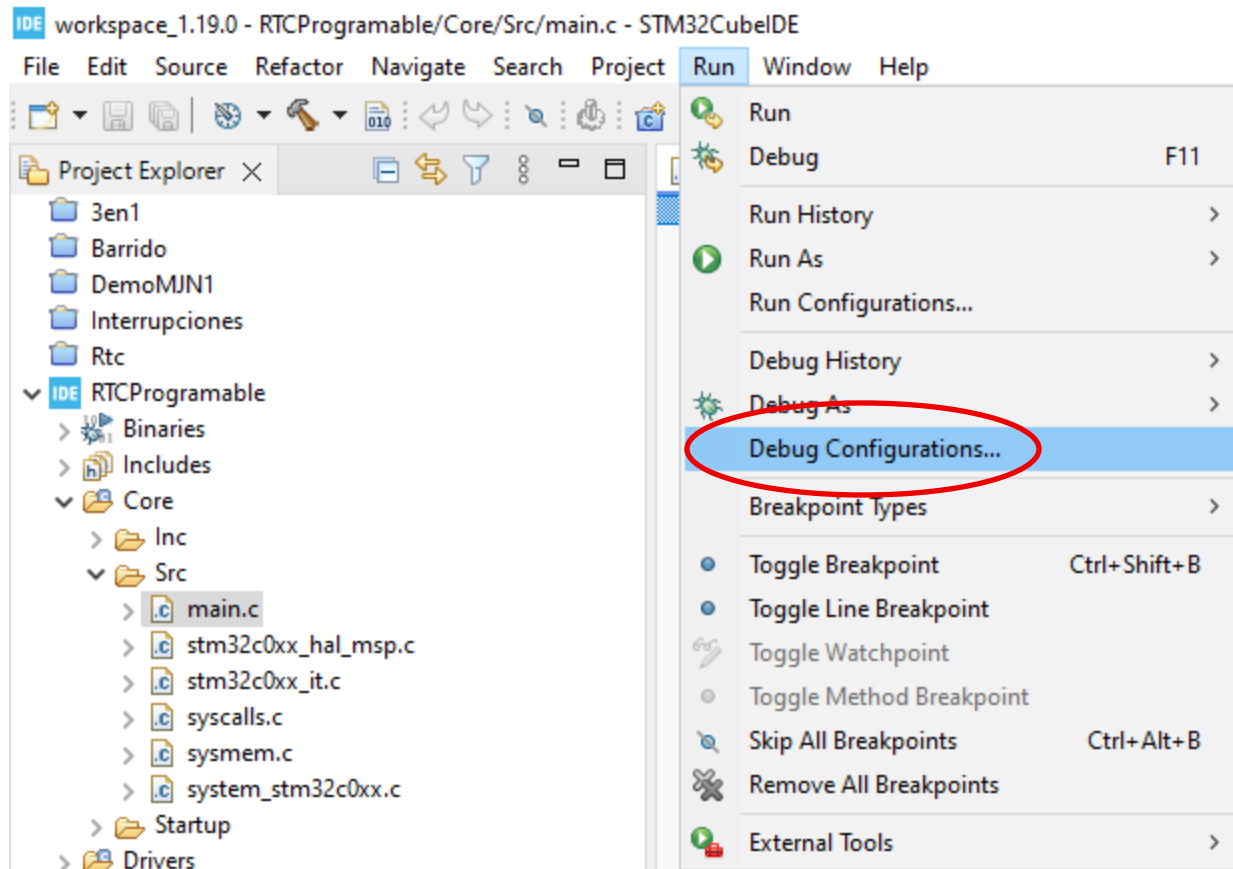
Configuration

Group By Peripherals v

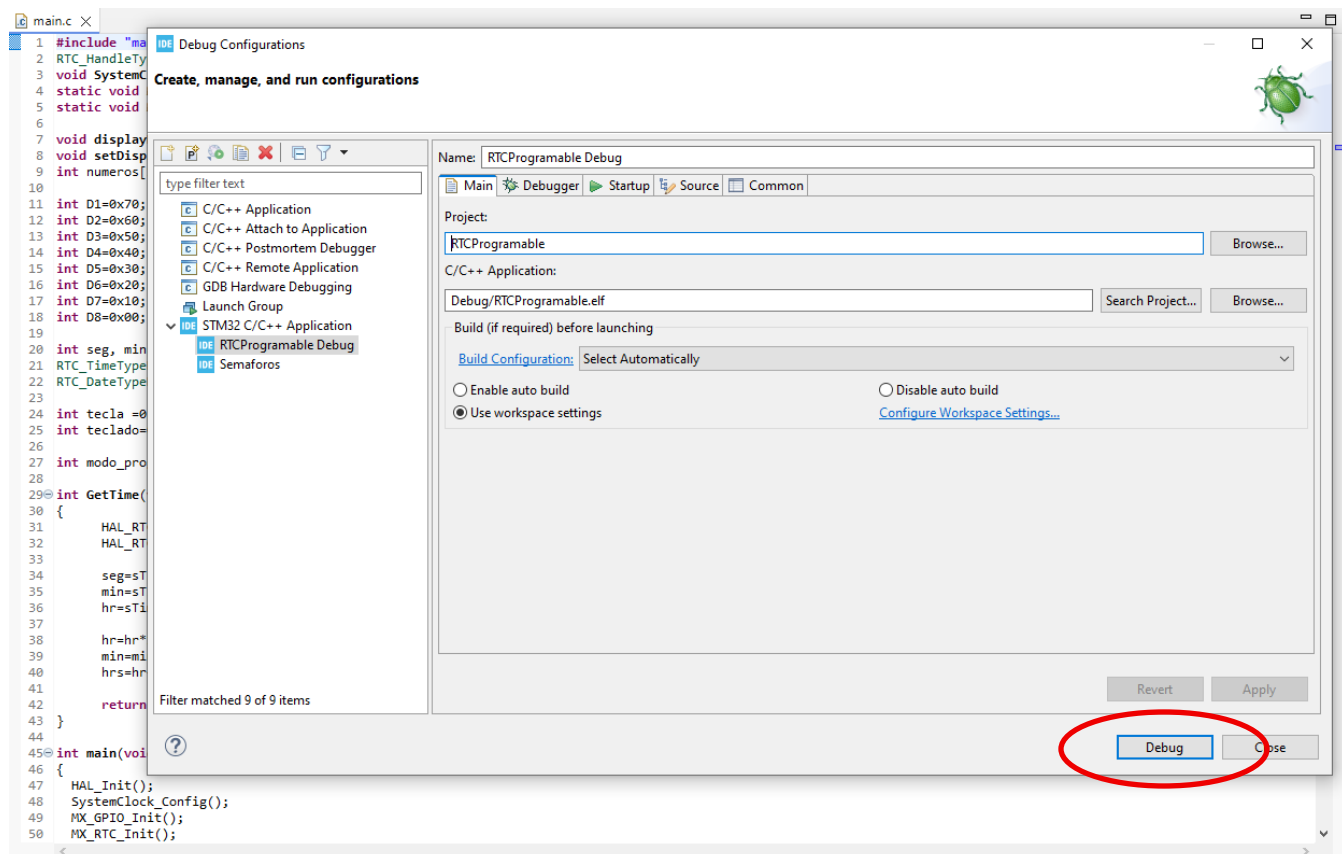
☒ GPIO☒ RCC☒ NVIC

NVIC Interrupt Table	Enabled	Preemption Priority
EXTI line 0 and line 1 interrupts	<input checked="" type="checkbox"/>	0
EXTI line 2 and line 3 interrupts	<input checked="" type="checkbox"/>	0

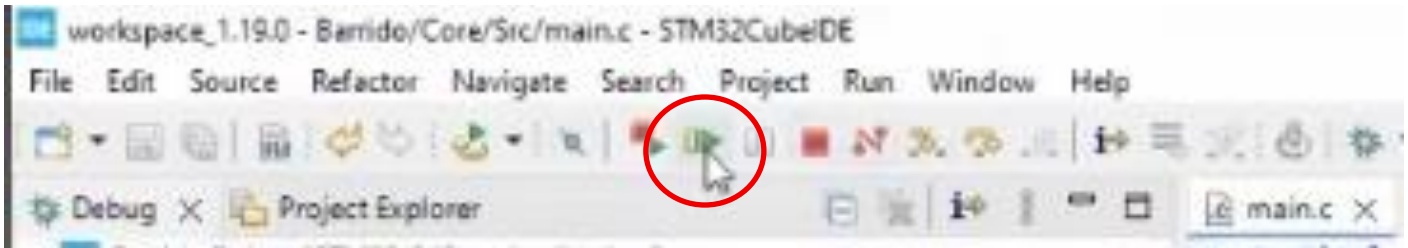
Para que el programa corra deberá de estar en modo debugger:



Seleccionar el proyecto:



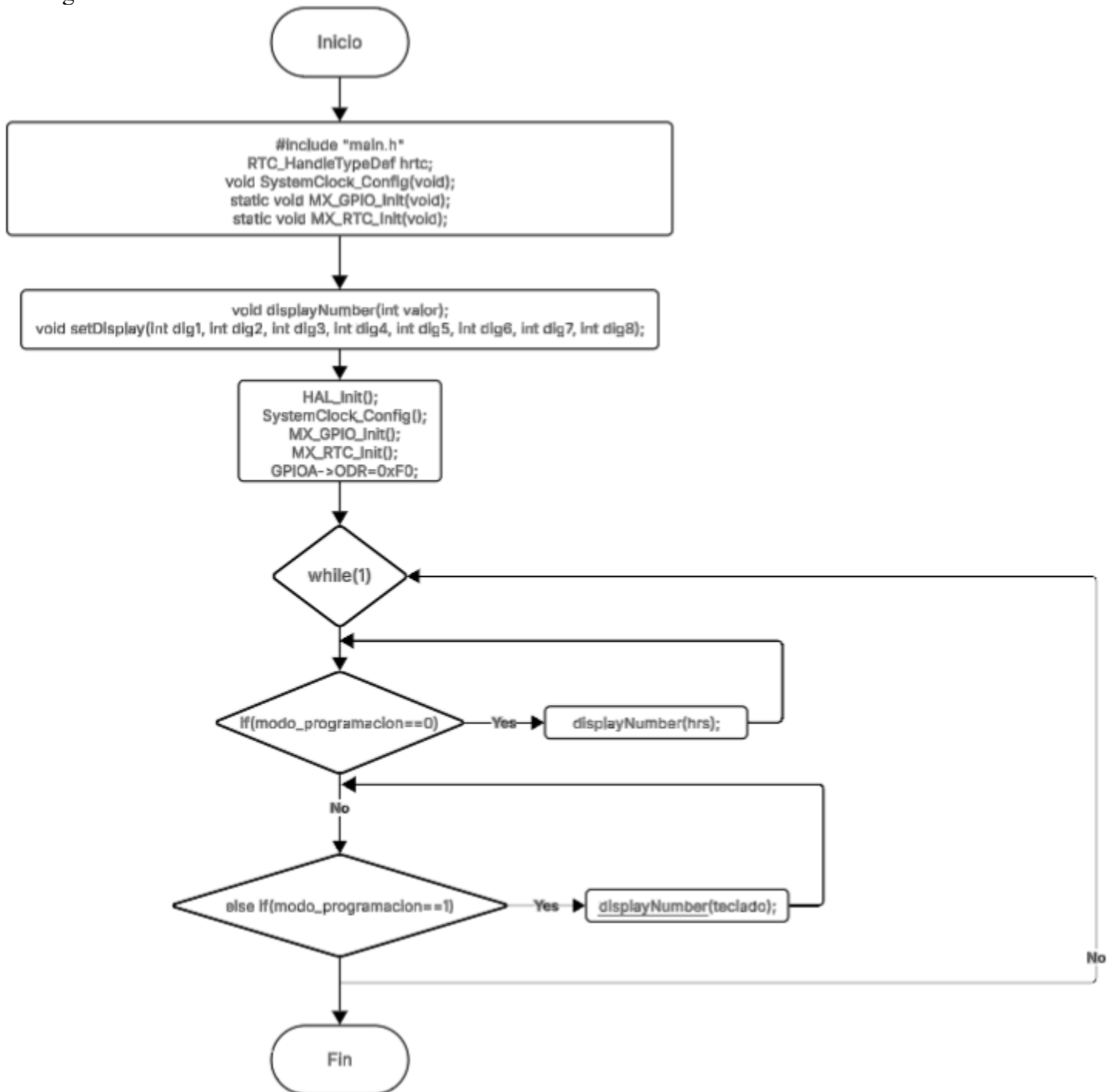
Estando en modo debugger, presionar el siguiente icono:



Ejecutar el código estando en ese apartado, fuera del modo debug no correrá, es decir, si solo se conecta al pc no funcionara como los primeros proyectos que si funcionan sin la necesidad del debugger.

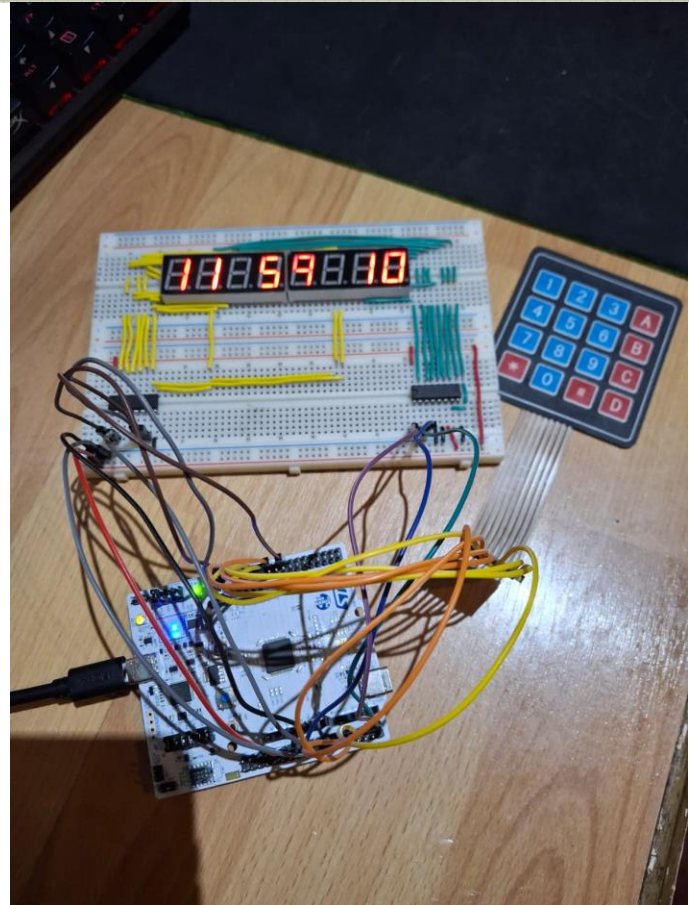
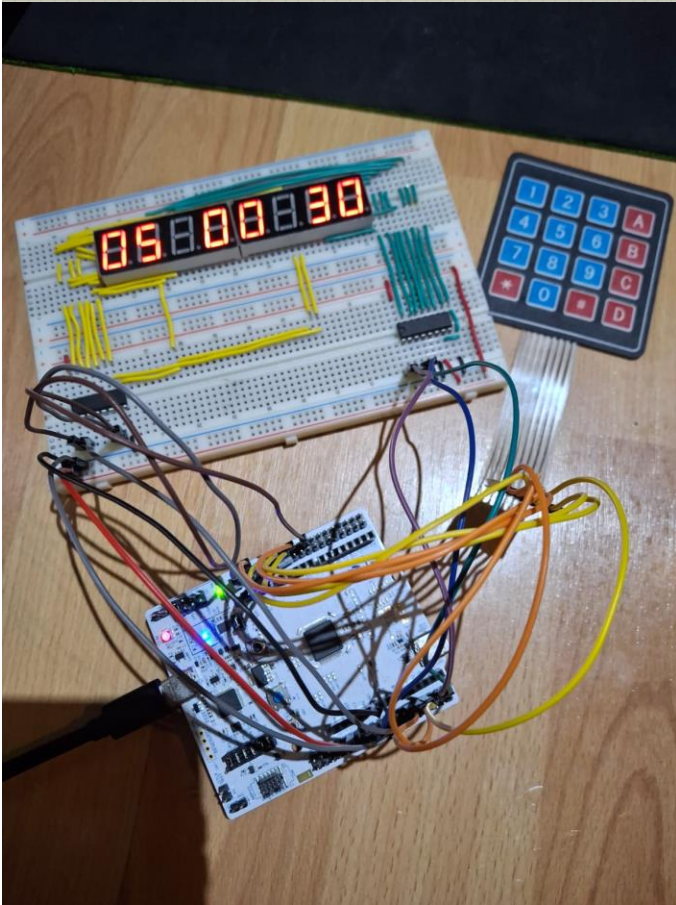
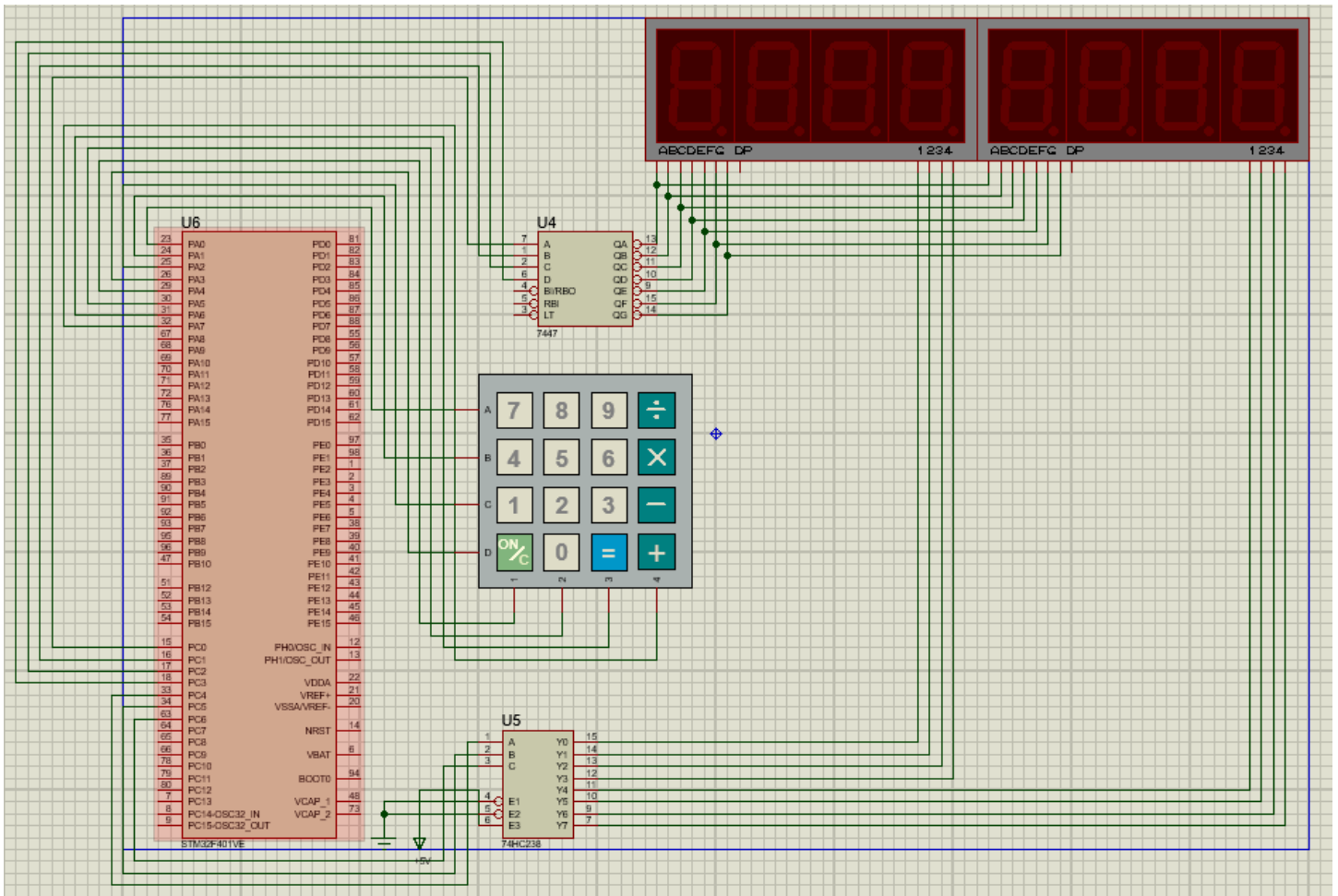
Otra opción es darle en `run -> run` pero puede fallar.  
¿Y porque solo funciona así? Es la misma pregunta que me hice.

## 2.- Algoritmo.



## 3.- Diagrama Eléctrico





4.- Escriba y explique el código utilizado para los siguientes apartados



## Funcion Callback

Línea 1: Función callback para interrupción por flanco ascendente

Línea 3: Declaración de variable time\_actual

Línea 4: Declaración de variable estática time\_antes

Línea 5: Asignación del tiempo actual mediante HAL\_GetTick()

Línea 6: Condición antirrebotes (diferencia de tiempo menor a 50ms)

Línea 8: Retorno early si no ha pasado tiempo suficiente

Línea 10: Actualización de time\_antes con el tiempo actual

Línea 11: Limpieza de todos los pines de salida de GPIOA

Línea 13: Estructura switch basada en el pin que generó la interrupción

Caso GPIO\_PIN\_0 (Líneas 16-47):

Línea 18: Activación de PIN\_4 como columna

Línea 19: Lectura de PIN\_0 para tecla 1

Línea 20: Asignación de tecla = 1

Línea 21: Actualización de variable teclado

Línea 23: Desactivación de PIN\_4

Línea 24: Activación de PIN\_5 como columna

Línea 25: Lectura de PIN\_0 para tecla 2

Línea 26: Asignación de tecla = 2

Línea 27: Actualización de variable teclado

Línea 29: Desactivación de PIN\_5

Línea 30: Activación de PIN\_6 como columna

Línea 31: Lectura de PIN\_0 para tecla 3

Línea 32: Asignación de tecla = 3

Línea 33: Actualización de variable teclado

Línea 35: Desactivación de PIN\_6

Línea 36: Activación de PIN\_7 como columna

Línea 37: Lectura de PIN\_0 para función horas

Línea 38: Reset de variable hr

Línea 39: Validación de rango horario (0-23)

Línea 41: Asignación de hr = teclado

Línea 42: Asignación a estructura de tiempo

Línea 43: Configuración del RTC

Línea 44: Reset de teclado

Caso GPIO\_PIN\_1 (Líneas 49-80):

Línea 51: Activación de PIN\_4 como columna

Línea 52: Lectura de PIN\_1 para tecla 4

Línea 53: Asignación de tecla = 4

Línea 54: Actualización de variable teclado

Línea 56: Desactivación de PIN\_4

Línea 57: Activación de PIN\_5 como columna

Línea 58: Lectura de PIN\_1 para tecla 5

Línea 59: Asignación de tecla = 5

Línea 60: Actualización de variable teclado

Línea 62: Desactivación de PIN\_5

Línea 63: Activación de PIN\_6 como columna

Línea 64: Lectura de PIN\_1 para tecla 6

Línea 65: Asignación de tecla = 6

Línea 66: Actualización de variable teclado

Línea 68: Desactivación de PIN\_6  
Línea 69: Activación de PIN\_7 como columna  
Línea 70: Lectura de PIN\_1 para función minutos  
Línea 71: Reset de variable min  
Línea 72: Validación de rango de minutos ( $\leq 59$ )  
Línea 74: Asignación de min = teclado  
Línea 75: Asignación a estructura de tiempo  
Línea 76: Configuración del RTC  
Línea 77: Reset de teclado

Caso GPIO\_PIN\_2 (Líneas 82-113):

Línea 84: Activación de PIN\_4 como columna  
Línea 85: Lectura de PIN\_2 para tecla 7  
Línea 86: Asignación de tecla = 7  
Línea 87: Actualización de variable teclado  
Línea 89: Desactivación de PIN\_4  
Línea 90: Activación de PIN\_5 como columna  
Línea 91: Lectura de PIN\_2 para tecla 8  
Línea 92: Asignación de tecla = 8  
Línea 93: Actualización de variable teclado  
Línea 95: Desactivación de PIN\_5  
Línea 96: Activación de PIN\_6 como columna  
Línea 97: Lectura de PIN\_2 para tecla 9  
Línea 98: Asignación de tecla = 9  
Línea 99: Actualización de variable teclado  
Línea 101: Desactivación de PIN\_6  
Línea 102: Activación de PIN\_7 como columna  
Línea 103: Lectura de PIN\_2 para función segundos  
Línea 104: Reset de variable seg  
Línea 105: Validación de rango de segundos ( $\leq 59$ )  
Línea 107: Asignación de seg = teclado  
Línea 108: Asignación a estructura de tiempo  
Línea 109: Configuración del RTC  
Línea 110: Reset de teclado

Caso GPIO\_PIN\_3 (Líneas 115-139):

Línea 117: Activación de PIN\_4 como columna  
Línea 118: Lectura de PIN\_3 para activar modo programación  
Línea 119: Asignación modo\_programacion = 1  
Línea 120: Sincronización del RTC  
Línea 122: Desactivación de PIN\_4  
Línea 123: Activación de PIN\_5 como columna  
Línea 124: Lectura de PIN\_3 para tecla 0  
Línea 125: Asignación de tecla = 0  
Línea 126: Actualización de variable teclado  
Línea 128: Desactivación de PIN\_5  
Línea 129: Activación de PIN\_6 como columna  
Línea 130: Lectura de PIN\_3 para desactivar modo programación  
Línea 131: Asignación modo\_programacion = 0  
Línea 132: Reset de teclado  
Línea 134: Desactivación de PIN\_6  
Línea 135: Activación de PIN\_7 como columna

Línea 136: Lectura de PIN\_3 para reset de teclado

Línea 137: Reset de teclado

Línea 141: Caso default del switch

Línea 144: Reactivación de pines 4-7 de GPIOA como salidas.

1	<code>void HAL_GPIO_EXTI_Rising_Callback(uint16_t GPIO_Pin)</code>
2	<code>{</code>
3	<code>uint32_t time_actual;</code>
4	<code>static uint32_t time_antes = 0;</code>
5	<code>time_actual = HAL_GetTick();</code>
6	<code>if (time_actual - time_antes &lt; 50)</code>
7	<code>{</code>
8	<code>return;</code>
9	<code>}</code>
10	<code>time_antes = time_actual;</code>
11	<code>GPIOA-&gt;ODR=0x00;</code>
12	
13	<code>switch (GPIO_Pin)</code>
14	
15	<code>{</code>
16	<code>case GPIO_PIN_0:</code>
17	
18	<code>HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, SET);</code>
19	<code>if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0)==1){</code>
20	<code>tecla = 1;</code>
21	<code>teclado=(teclado%100000000)*10 + tecla;</code>
22	<code>}</code>
23	<code>HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, RESET);</code>
24	<code>HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, SET);</code>
25	<code>if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0)==1){</code>
26	<code>tecla = 2;</code>
27	<code>teclado=(teclado%100000000)*10 + tecla;</code>
28	<code>}</code>
29	<code>HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, RESET);</code>
30	<code>HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, SET);</code>
31	<code>if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0)==1){</code>
32	<code>tecla = 3;</code>
33	<code>teclado=(teclado%100000000)*10 + tecla;</code>
34	<code>}</code>
35	<code>HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, RESET);</code>
36	<code>HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, SET);</code>
37	<code>if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0)==1){</code>
38	<code>hr=0;</code>
39	<code>if (teclado&gt;=0 &amp;&amp; teclado&lt;=23)</code>
40	<code>{</code>
41	<code>hr = teclado;</code>
42	<code>sTime1.Hours = hr;</code>
43	<code>HAL_RTC_SetTime(&amp;hrtc, &amp;sTime1, RTC_FORMAT_BIN);</code>
44	<code>teclado = 0;</code>

45	}
46	}
47	break;
48	
49	case GPIO_PIN_1:
50	
51	HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, SET);
52	if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_1)==1){
53	tecla = 4;
54	teclado=(teclado%100000000)*10 + tecla;
55	}
56	HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, RESET);
57	HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, SET);
58	if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_1)==1){
59	tecla = 5;
60	teclado=(teclado%100000000)*10 + tecla;
61	}
62	HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, RESET);
63	HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, SET);
64	if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_1)==1){
65	tecla = 6;
66	teclado=(teclado%100000000)*10 + tecla;
67	}
68	HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, RESET);
69	HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, SET);
70	if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_1)==1){
71	min=0;
72	if (teclado<=59)
73	{
74	min = teclado;
75	sTime1.Minutes = min;
76	HAL_RTC_SetTime(&hrtc, &sTime1, RTC_FORMAT_BIN);
77	teclado = 0;
78	}
79	}
80	break;
81	
82	case GPIO_PIN_2:
83	
84	HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, SET);
85	if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_2)==1){
86	tecla = 7;
87	teclado=(teclado%100000000)*10 + tecla;
88	}
89	HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, RESET);
90	HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, SET);
91	if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_2)==1){
92	tecla = 8;
93	teclado=(teclado%100000000)*10 + tecla;
94	}

95	HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, RESET);
96	HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, SET);
97	if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_2)==1){
98	tecla = 9;
99	teclado=(teclado%100000000)*10 + tecla;
100	}
101	HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, RESET);
102	HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, SET);
103	if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_2)==1){
104	seg=0;
105	If (teclado<=59)
106	{
107	seg = teclado;
108	sTime1.Seconds = seg;
109	HAL_RTC_SetTime(&hrtc, &sTime1, RTC_FORMAT_BIN);
110	teclado = 0;
111	}
112	}
113	break;
114	
115	case GPIO_PIN_3:
116	
117	HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, SET);
118	if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_3)==1){
119	modo_programacion = 1;
120	HAL_RTC_SetTime(&hrtc, &sTime1, RTC_FORMAT_BIN);
121	}
122	HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, RESET);
123	HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, SET);
124	if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_3)==1){
125	tecla = 0;
126	teclado=(teclado%100000000)*10 + tecla;
127	}
128	HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, RESET);
129	HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, SET);
130	if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_3)==1){
131	modo_programacion = 0;
132	teclado=0;
133	}
134	HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, RESET);
135	HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, SET);
136	if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_3)==1){
137	teclado=0;
138	}
139	break;
140	
141	default:
142	break;
143	}
144	GPIOA->ODR=0xF0;

145	}
-----	---

Loop de control principal

Línea 1: Bucle infinito principal del programa  
 Línea 3: Condición para modo normal (modo\_programacion == 0)  
 Línea 5: Llama a la función GetTime() para obtener la hora actual  
 Línea 6: Muestra las horas en el display mediante displayNumber(hrs)  
 Línea 7: Condición else if para modo programación (modo\_programacion == 1)  
 Línea 9: Muestra el valor del teclado en el display mediante displayNumber(teclado)

1	while (1)
2	{
3	if(modo_programacion==0)
4	{
5	GetTime();
6	displayNumber(hrs);
7	} else if(modo_programacion==1)
8	{
9	displayNumber(teclado);
10	}
11	}

Nota: Adjunte el código completo del programa del microcontrolador y de la página web en un archivo zip cada uno.