

Count 24 game

Group G08

Meet the Group G08



นาย ภาสวิชญ์
พุทธิบานเจริญศรี

ID : 1911310116

Hardware Team
Software Team



นางสาว กรณก
สัมฤทธิ์

ID : 1911310124

Circuit Designer
Testing Team
Presentation Leader



นาย พสวิร์
มุตติศาสน์

ID : 1911310801

Game Designer
Software Team
Flowchart



นาย ปรัชญ์
สันติไชยกุล

ID : 1911310710

Team Leader
Testing Team



นาย รอนໂชต
อุดมศาสตร์พວ

ID : 1911310132

Game Designer
Software Team
Overall Workflow
General Supporter



Content

Count 24 game

1

Overview
Requirement

2

Specification

3

Architectural
design

4

Detailed design

5

Coding flowchart

6

Demo video

7

Problem
and solution

8

Conclusion

Overview Requirement

01

1

✓ Count 24

หากฝ่ายไหนบันทึกตัวเลขได้ถึง 24 ก่อนจะเป็นฝ่ายแพ้

✓ Show status numbers from LCD

- ไม่มีสี -> ตัวเลขที่เลือกเป็นเลข 1
- สีเขียว -> ตัวเลขที่เลือกเป็นเลข 2
- สีน้ำเงิน -> ตัวเลขที่เลือกเป็นเลข 3
- สีเขียวและน้ำเงิน -> ตัวเลขที่เลือกเป็นเลข random

✓ Music box

เกมจะเล่นเพลง เมื่อผู้เล่นเป็นฝ่ายชนะ

2 Specification

System information

- MCU : STM32L152RB
- LCD : 6-digit, 24 segment
- Pushbuttons : user button
- TIMER : PWM Mode
- Software : Keil / Language C



Function

- Show game count number (LCD)
- Selected number (Matrix Switch)
- Confirm number (push button)
- Indicate Number status (LEDs)
- Music box (PWM Mode)

Behavior

- Show game count number (LCD)
 - 6-digit EG: xx-p-y
 - xx : current total number
 - p: p-player, c-computer
 - y: selected number
- Selected number (Matrix Switch)
 - 00 -> 1, 01 -> 2 , 10 -> 3 , 11-> rand
- Confirm number (push button) : User button
- Indicate Number status (LEDs)
 - None -> 1 ,Green -> 2, Blue -> 3
,Green & Blue -> rand
- Music box (PWM Mode)
 - win: play soundtrack
 - lose: buzzer

Specification

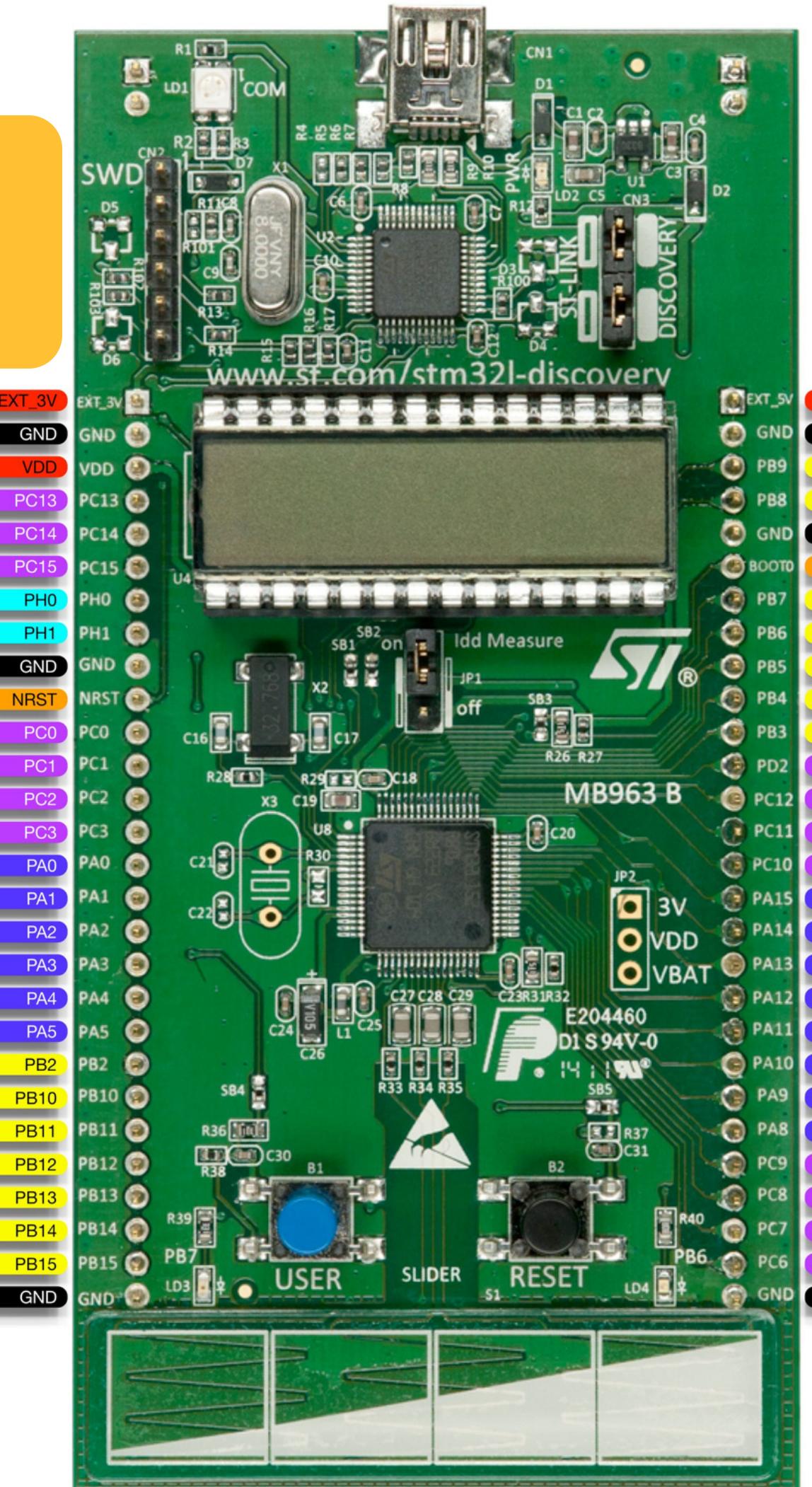
Project planning

Task	December		January			February		
	3	4	1	2	3	4	1	2
Requirement & planning	1							
Specification		1						
Architectural design		1						
Detailed design			1					
Coding			1	1				
Unit testing				1				
Integration testing					1			
System testing					1	1		
Acceptance testing						1	1	
Presentation					1	1		

MODEL DESIGN ARCHITECTURAL DESIGN

Count 24 game - Group G08

3



p

Input

05

Push Button

Matrix Switch

Process

EXTI

DAC

TIM

Output

LED

LCD

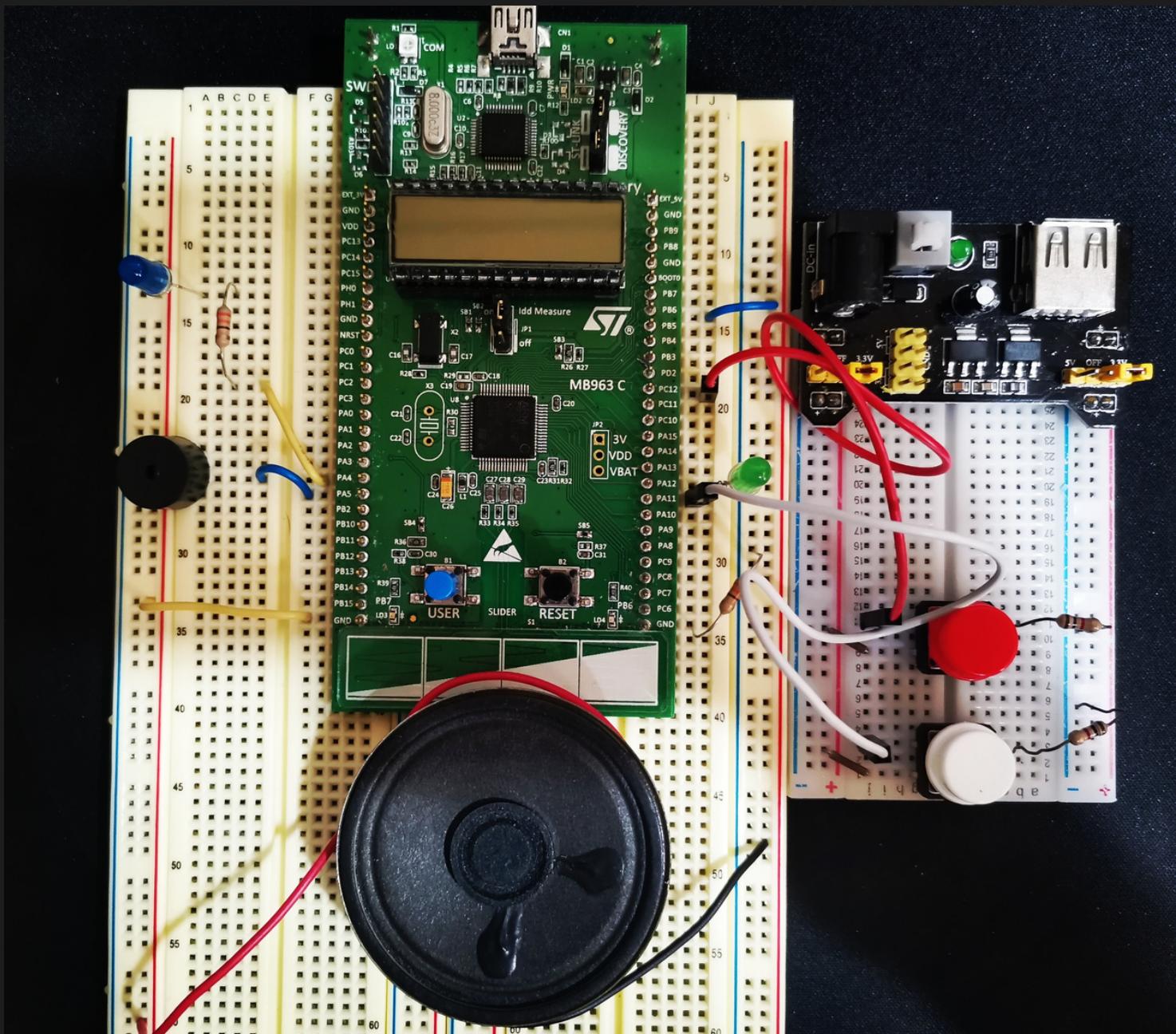
Buzzer

Speaker

Hardware and circuit

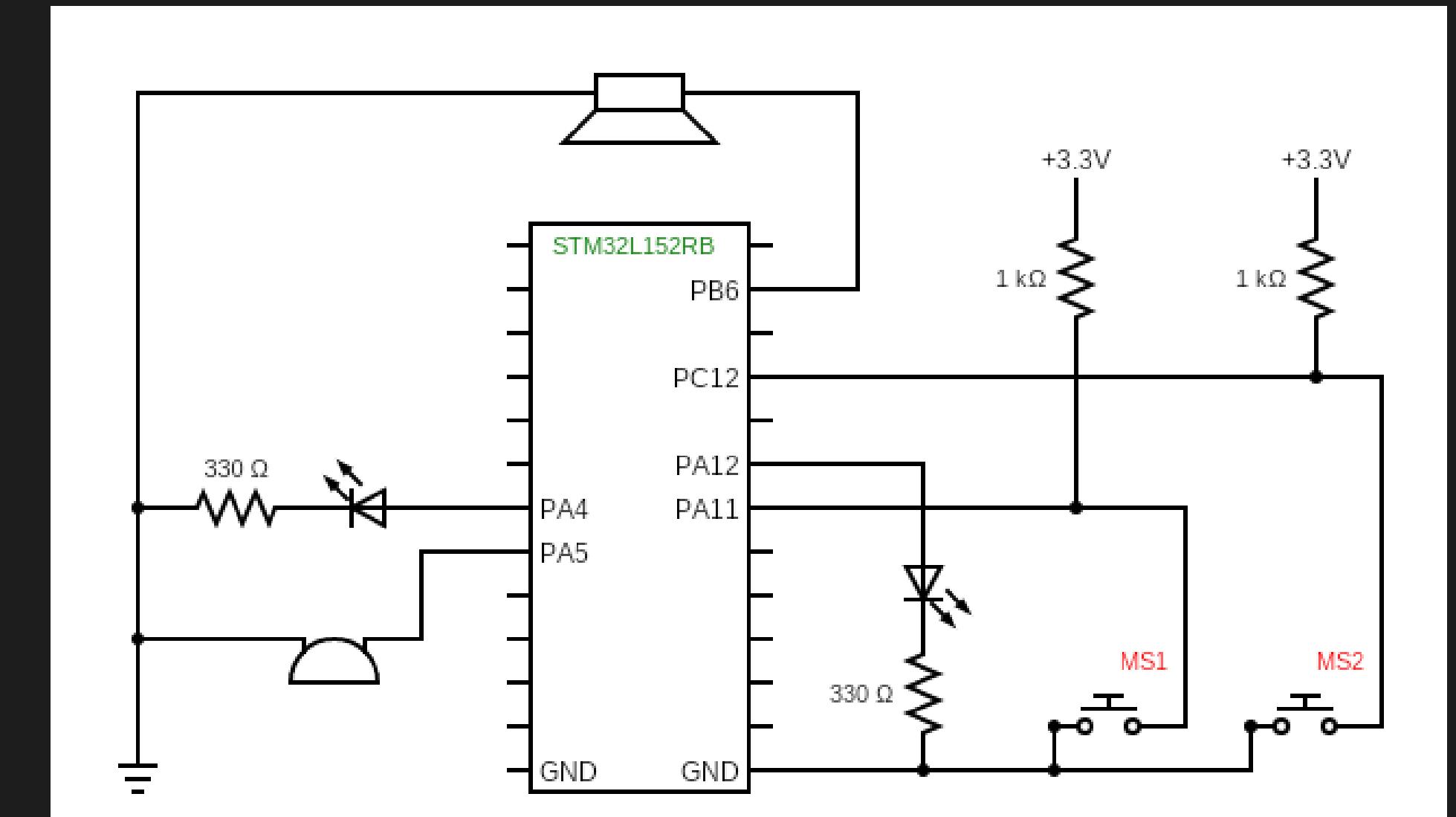
p

HARDWARE



06

CIRCUIT

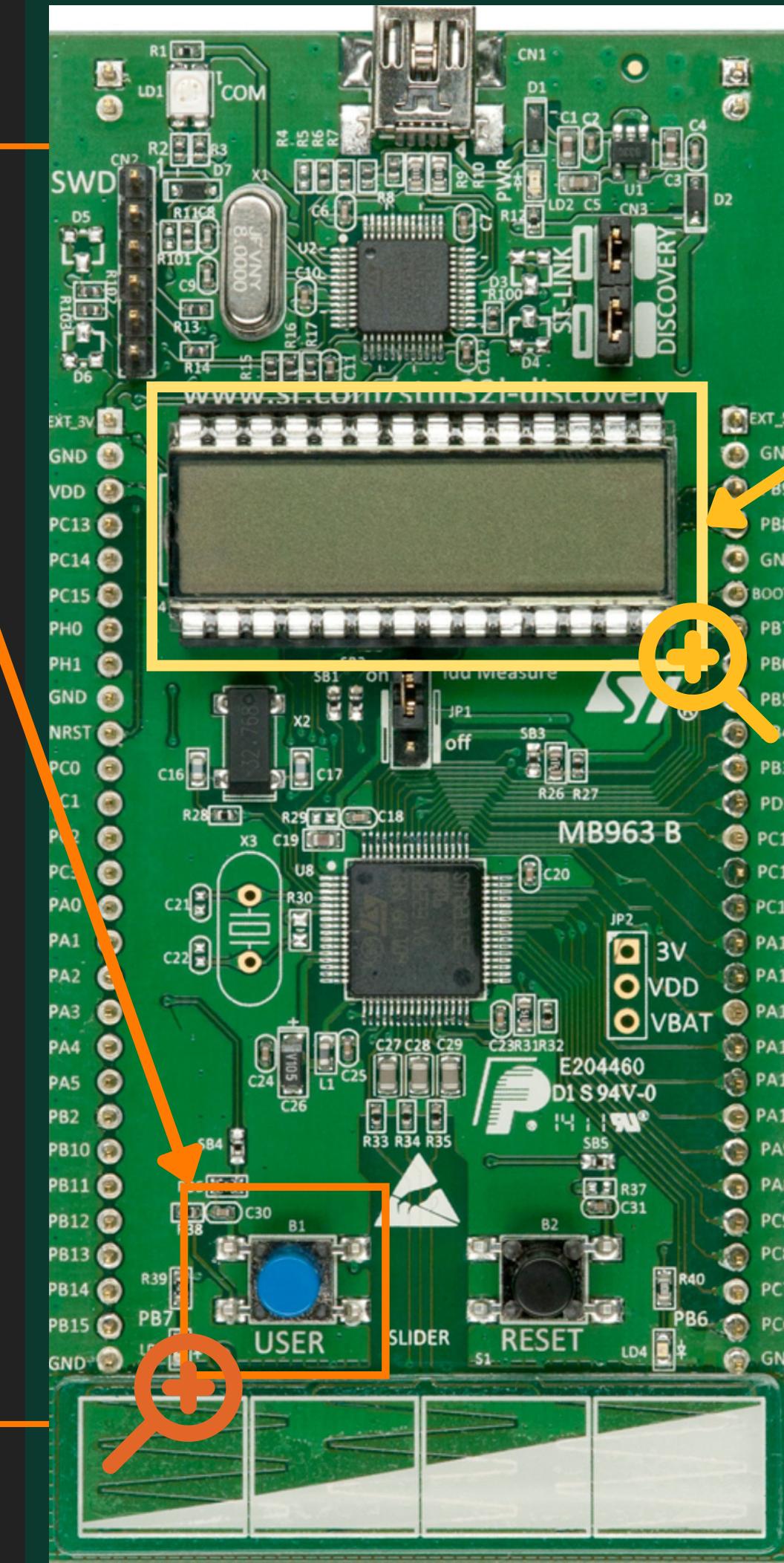
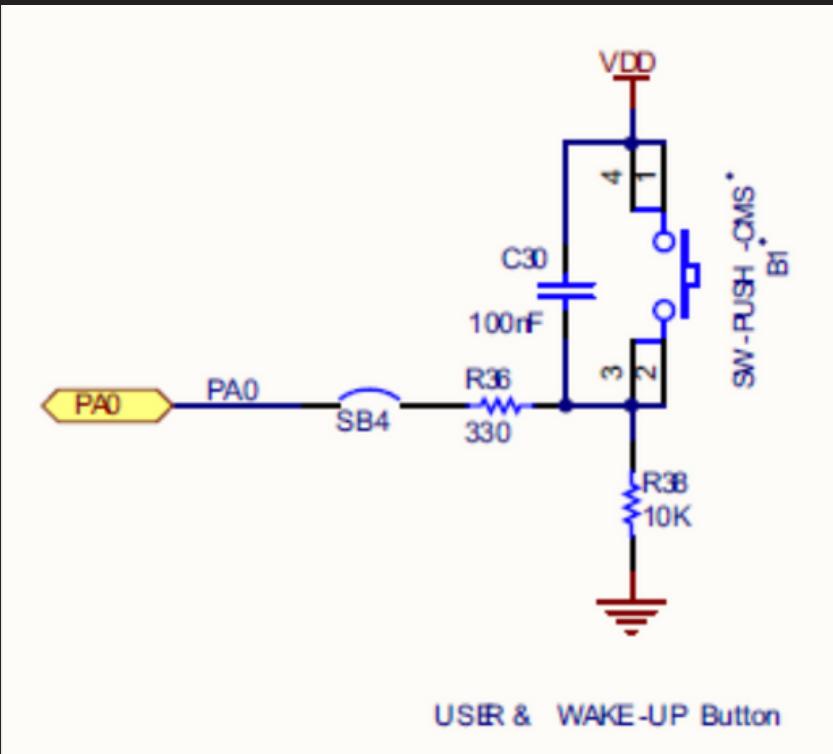


Architectural Design : User Button

3 Features

The STM32L1 discovery offers the following features:

- Two pushbuttons (user and reset)
- One linear touch sensor and four touchkeys



Architectural Design : LCD

4.9 LCD (24 segments, 4 commons)

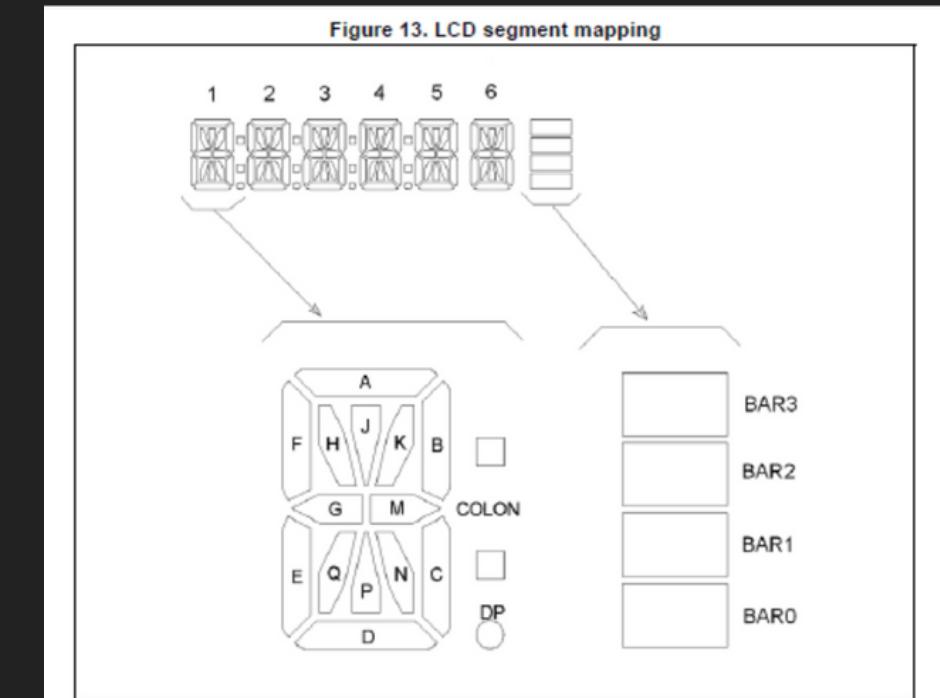
This LCD allows the STM32L152 to display any information on six 14-segment digits and 4 bars, using all COMs. (See the LCD segment mapping in [Figure 18](#) and pin connections in [Table 7](#).)

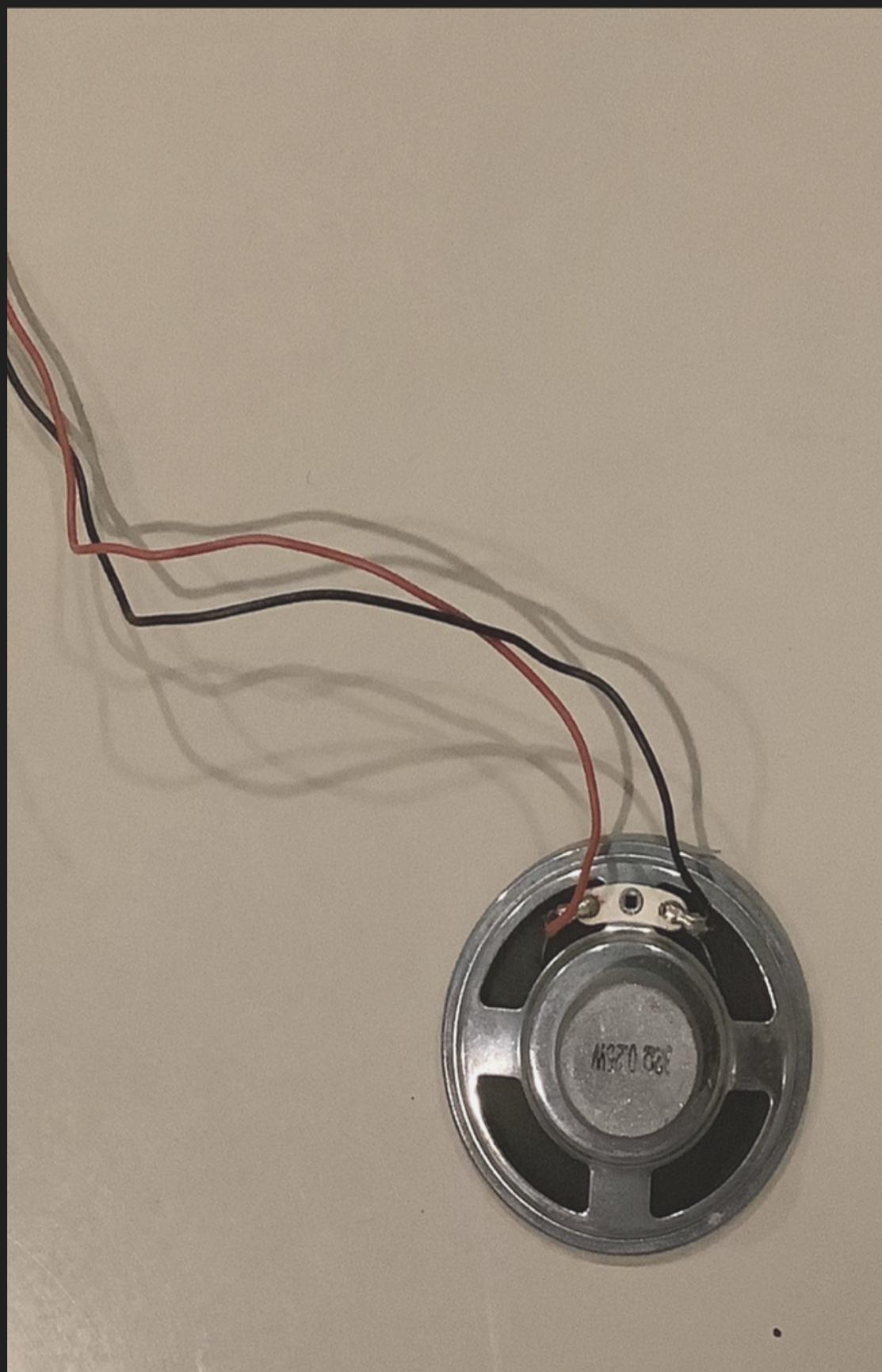
Note: This LCD also supports six 8-segment digits by only using COM0 and COM1. This configuration allows COM2 and COM3 to be used as I/O ports. In this case the 2 LCD pins must not be plugged into the LCD socket. To proceed with this configuration, remove the LCD carefully, slightly open the COM2 and COM3 pins (pin 13 and pin 14) of the LCD, then replug it in the socket.

Characteristics overview:

- 24 segments and 4 commons
- Drive method: multiplexed 1/4 duty, 1/3 bias
- Operating voltage: 3 V
- Operating temperature: 0 to 50°C
- Connector: 28-pin DIL 2.54 mm pitch

Note: When the LCD is plugged, all I/O ports listed in [Table 7](#) are unavailable. To use one of these as I/O, you must remove the LCD.





Architectural Design : Timer

3.15 Timers and watchdogs

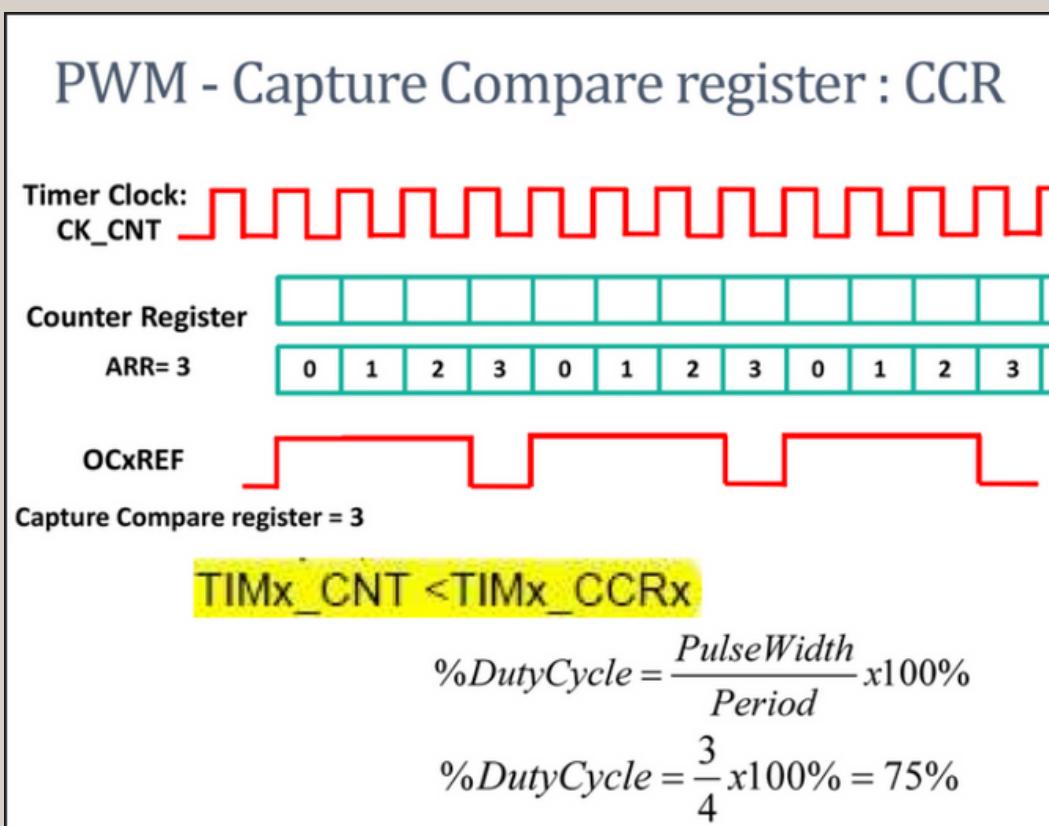
The ultra-low-power STM32L15xxx devices include six general-purpose timers, two basic timers and two watchdog timers.

Timer	Counter resolution	Counter type	Prescaler factor	DMA request generation	Capture/compare channels	Complementary outputs
TIM2, TIM3, TIM4	16-bit	Up, down, up/down	Any integer between 1 and 65536	Yes	4	No
TIM9	16-bit	Up	Any integer between 1 and 65536	No	2	No

Timer	Counter resolution	Counter type	Prescaler factor	DMA request generation	Capture/compare channels	Complementary outputs
TIM10, TIM11	16-bit	Up	Any integer between 1 and 65536	No	1	No
TIM6, TIM7	16-bit	Up	Any integer between 1 and 65536	Yes	0	No

PWM mode

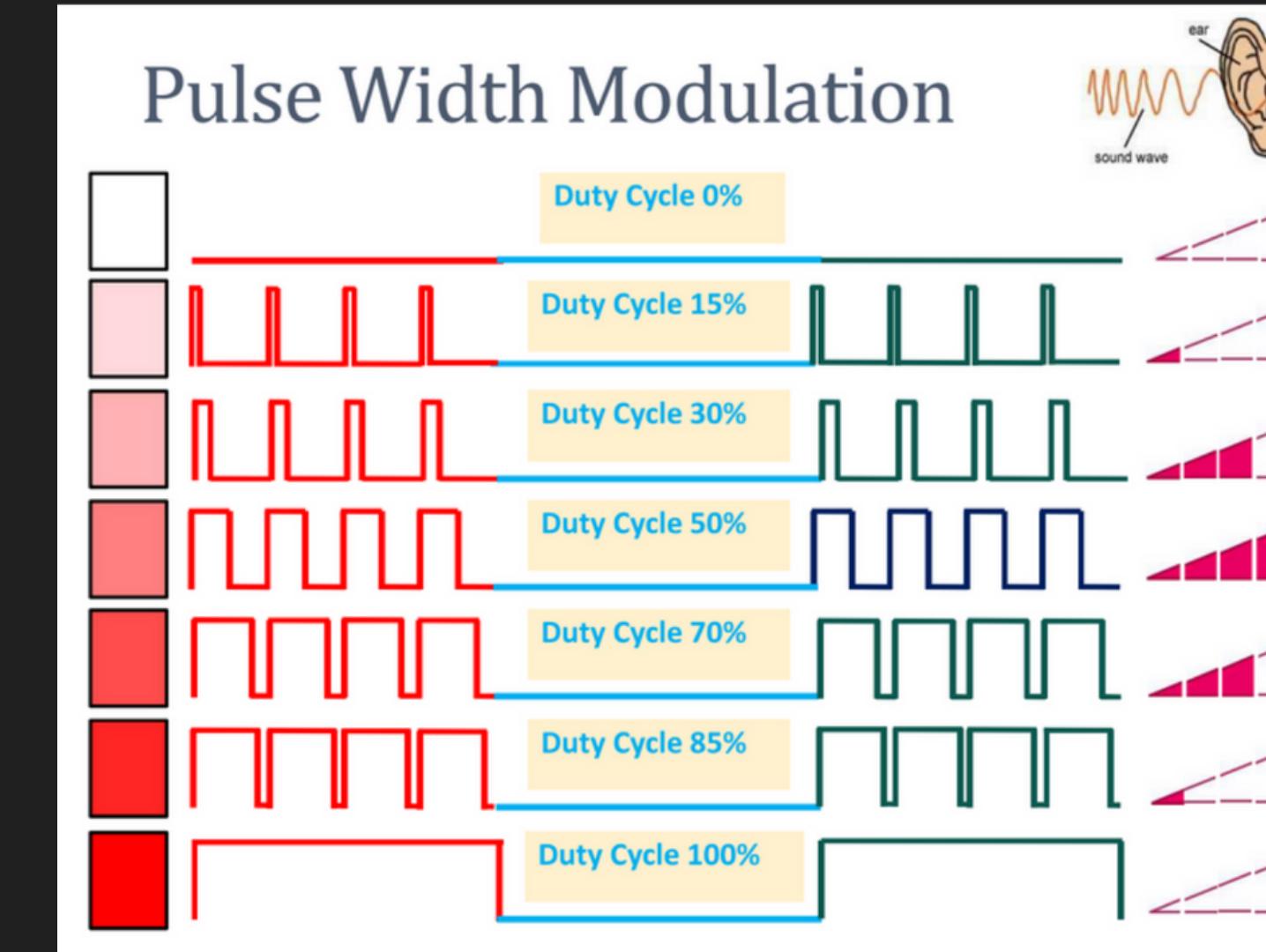
Pulse Width Modulation mode allows you to generate a signal with a frequency determined by the value of the **TIMx_ARR** register and a duty cycle determined by the value of the **TIMx_CCRx** register.



Architectural Design :

Timer

TIM -> PWM Mode



Architectural Design :

TIM Music box



10



Note	Frequency
E	1318.5

STM32L152RB Time base

Time base generator

The update event period is calculated as follows:

$$\text{Timer clock} = CK_CNT = \frac{\text{TIM_CLK}}{(PSC+1) \times (ARR+1)}$$

Where: TIM_CLK = timer clock input = 16 MHz

PSC = 16-bit Prescaler register

ARR = 16/32-bit Auto Reload register

** 16-bit value = 0 - 65535 **

$$1318.5 = \frac{16\text{MHz}}{(PSC+1) \times (ARR+1)}$$

$$= \frac{16\text{MHz}}{(0+1) \times (ARR+1)}$$

$$= \frac{16\text{MHz}}{(ARR+1)}$$

$$ARR+1 = 16,000,000 \times \frac{1}{1318.5}$$

$$ARR = 12135.0018 - 1 = 12135 - 1 = 12134$$

	C	C#	D	Eb	E	F	F#	G	G#	A	Bb	B
0	16.35	17.32	18.35	19.45	20.60	21.83	23.12	24.50	25.96	27.50	29.14	30.87
1	32.70	34.65	36.71	38.89	41.20	43.65	46.25	49.00	51.91	55.00	58.27	61.74
2	65.41	69.30	73.42	77.78	82.41	87.31	92.50	98.00	103.8	110.0	116.5	123.5
3	130.8	138.6	146.8	155.6	164.8	174.6	185.0	196.0	207.7	220.0	233.1	246.9
4	261.6	277.2	293.7	311.1	329.6	349.2	370.0	392.0	415.3	440.0	466.2	493.9
5	523.3	554.4	587.3	622.3	659.3	698.5	740.0	784.0	830.6	880.0	932.3	987.8
6	1047	1109	1175	1245	1319	1397	1480	1568	1661	1760	1865	1976
7	2093	2217	2349	2489	2637	2794	2960	3136	3322	3520	3729	3951
8	4186	4435	4699	4978	5274	5588	5920	6272	6645	7040	7459	7902

Board pin allocation

p

G08 Pin allocation

\Pin Port	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	Cyan	Green	Green	Green	Orange	Blue	Grey	Grey	Green	Green	Green	Red	Orange	Grey	Grey	Green
B	Grey	Grey	Grey	Green	Green	Green	White	Green	Green	Green	Green	Green	Green	Green	Green	Green
C	Green	Green	Green	Green	Grey	Grey	Green	Green	Green	Green	Green	Red	Grey	Grey	Grey	Grey
D	Grey	Grey	Grey	White	Grey	Grey	Grey	Grey	Grey	Grey	Grey	Grey	Grey	Grey	Grey	Grey
H	Grey	Grey	Grey	Grey	Grey	Grey	Grey	Grey	Grey	Grey	Grey	Grey	Grey	Grey	Grey	Grey

User Button

Matrix Switch

LED

Buzzer - DAC

LCD

Speaker -
TIM4_CH1



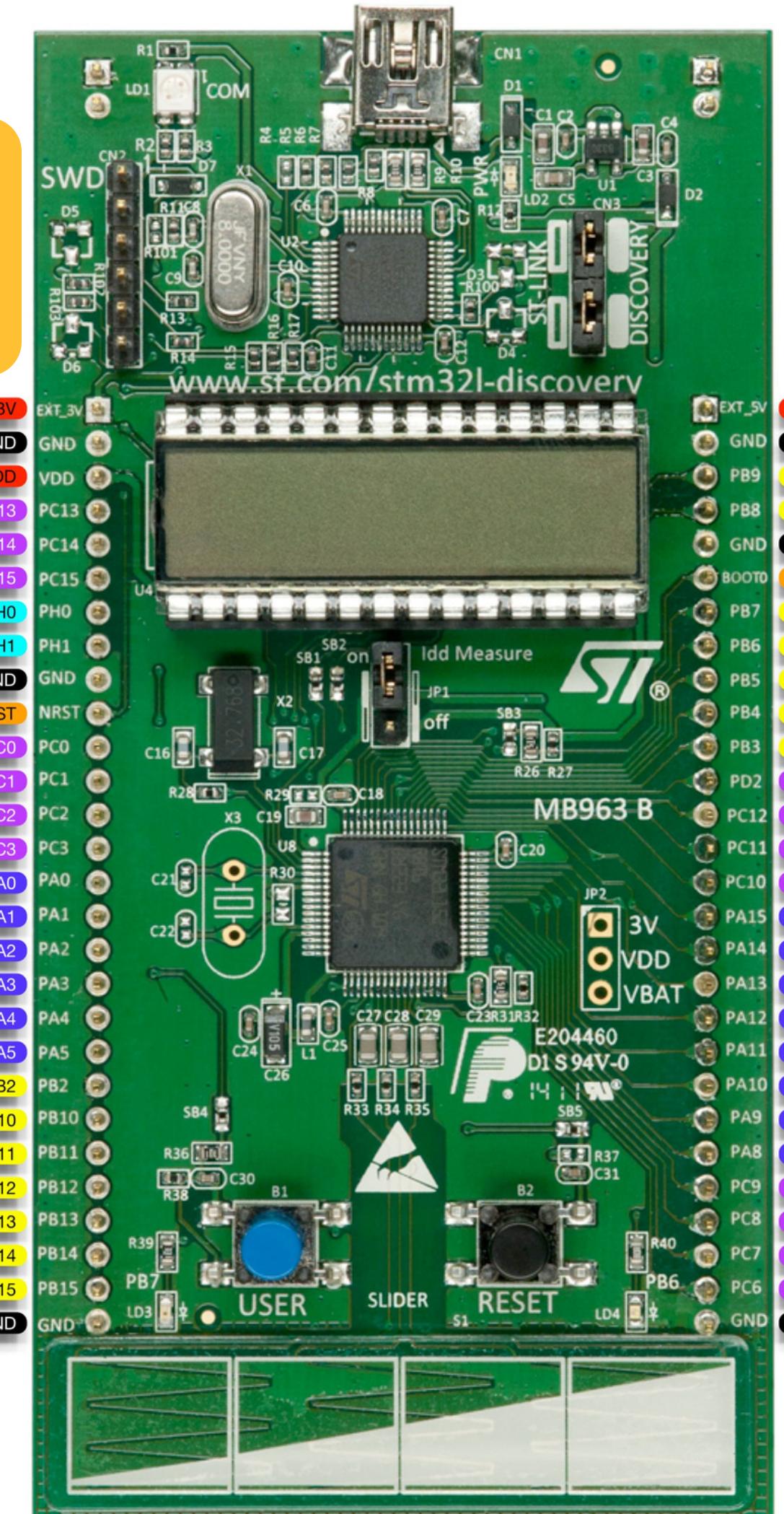
12

4

MODEL DESIGN

DETAILED DESIGN

Count 24 game -Group G08



Top-down design

COUNT 24 GAME

p

Count 24 game

13

Function :
Show game
count number

Function :
Selected
number

Function :
Confirm
number

Function :
Indicate
Number status

Function :
Music
box

LCD

Matrix
switch

User
button

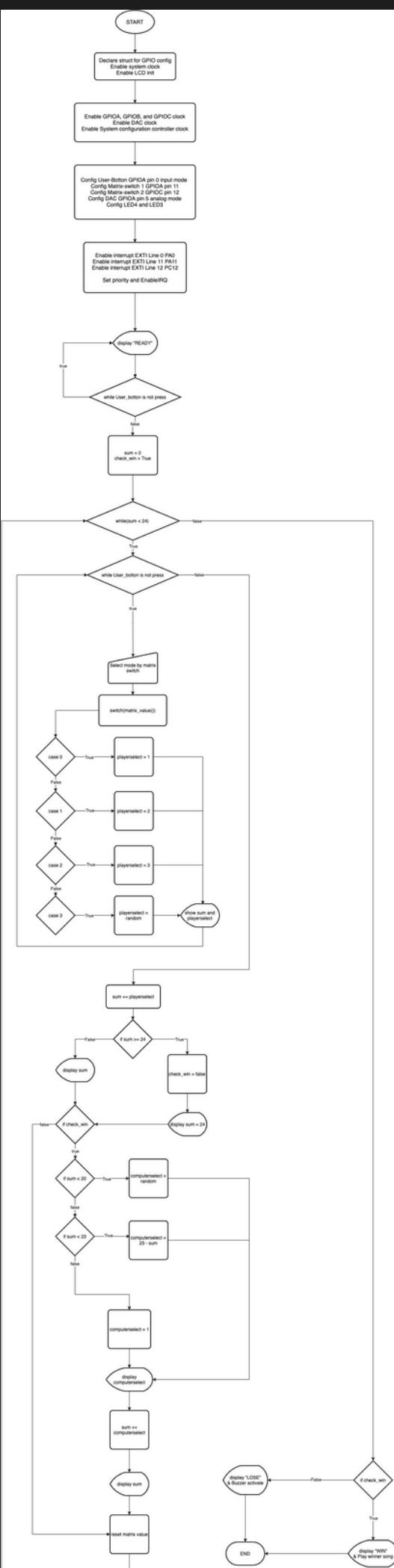
LED

Buzzer
Speaker



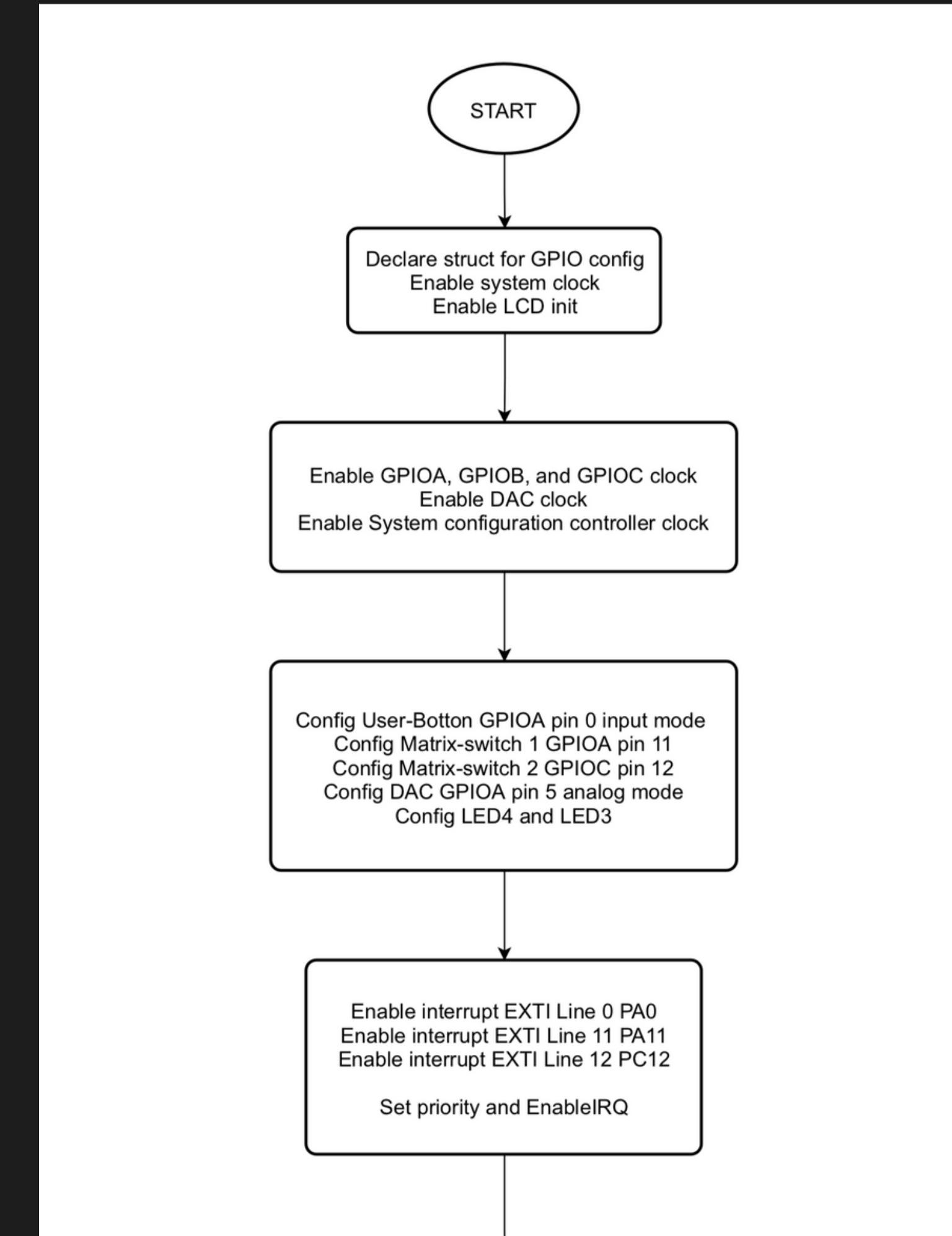
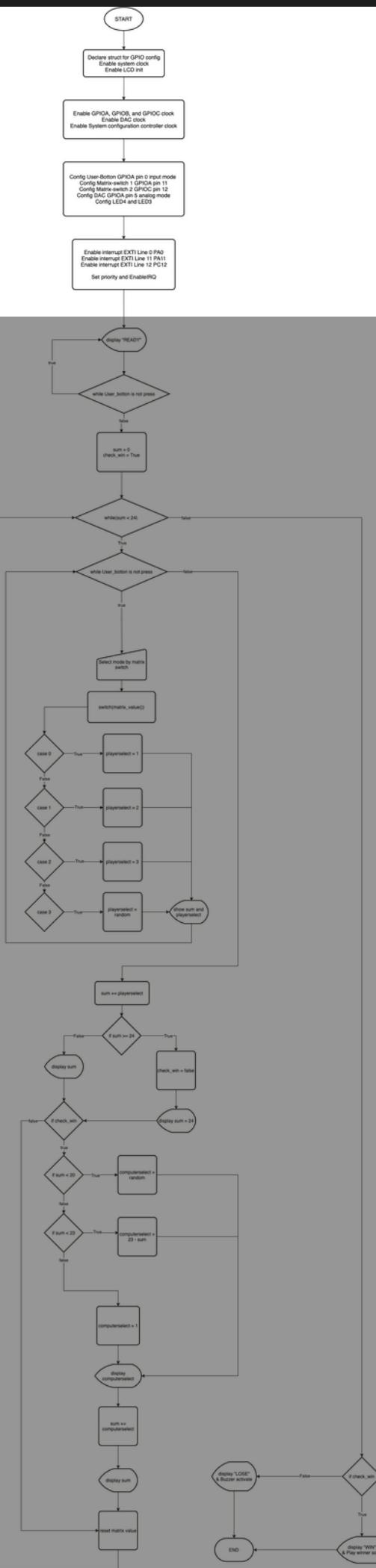


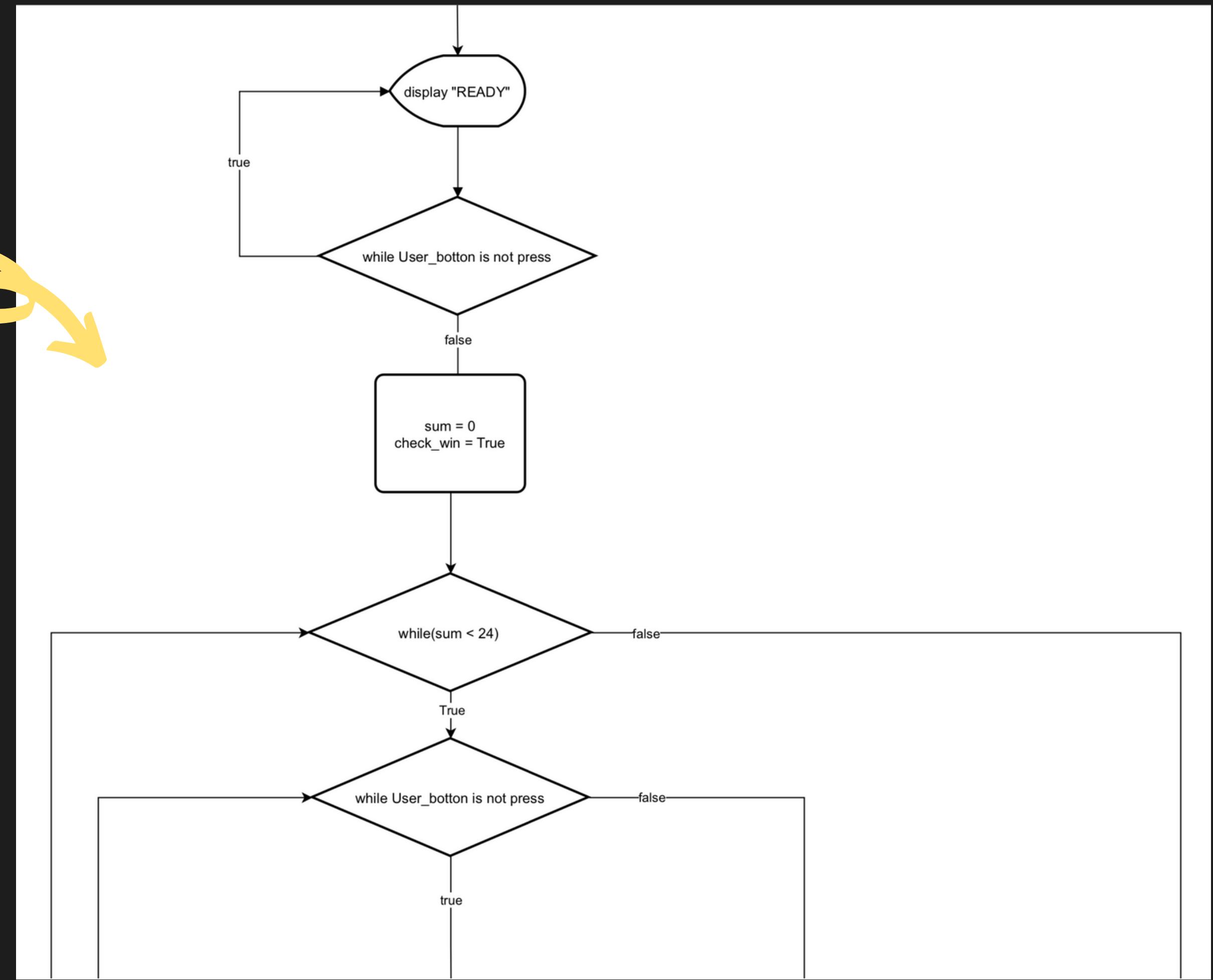
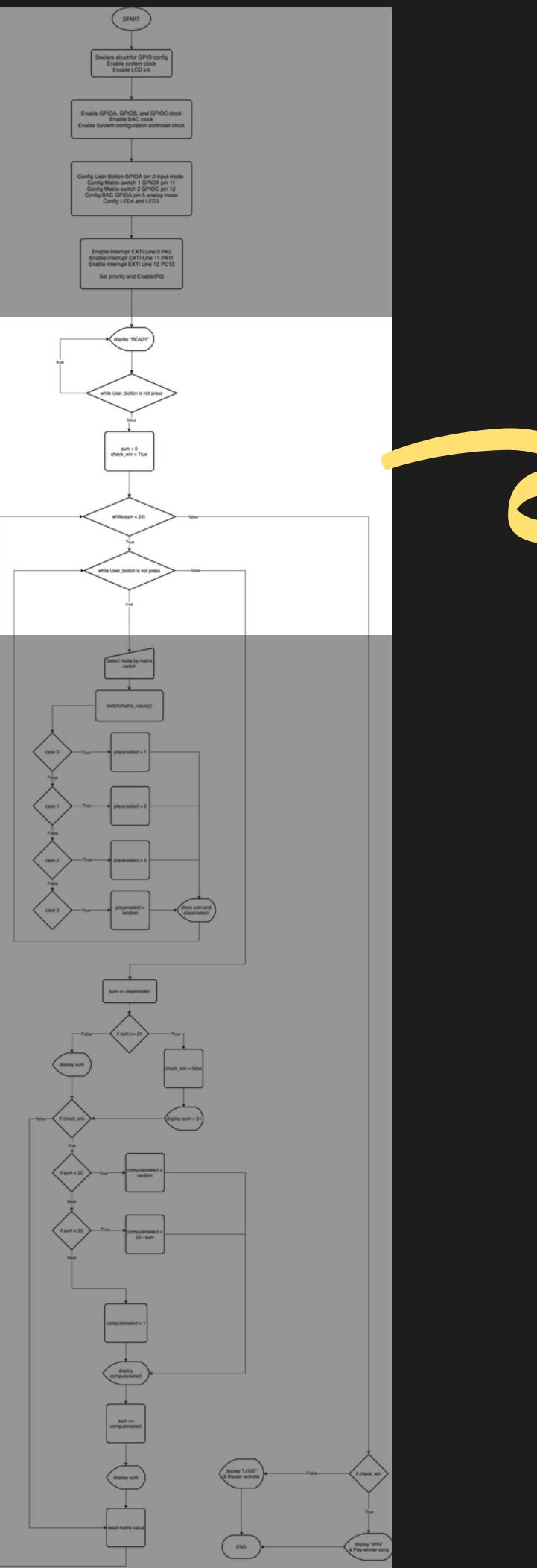
14

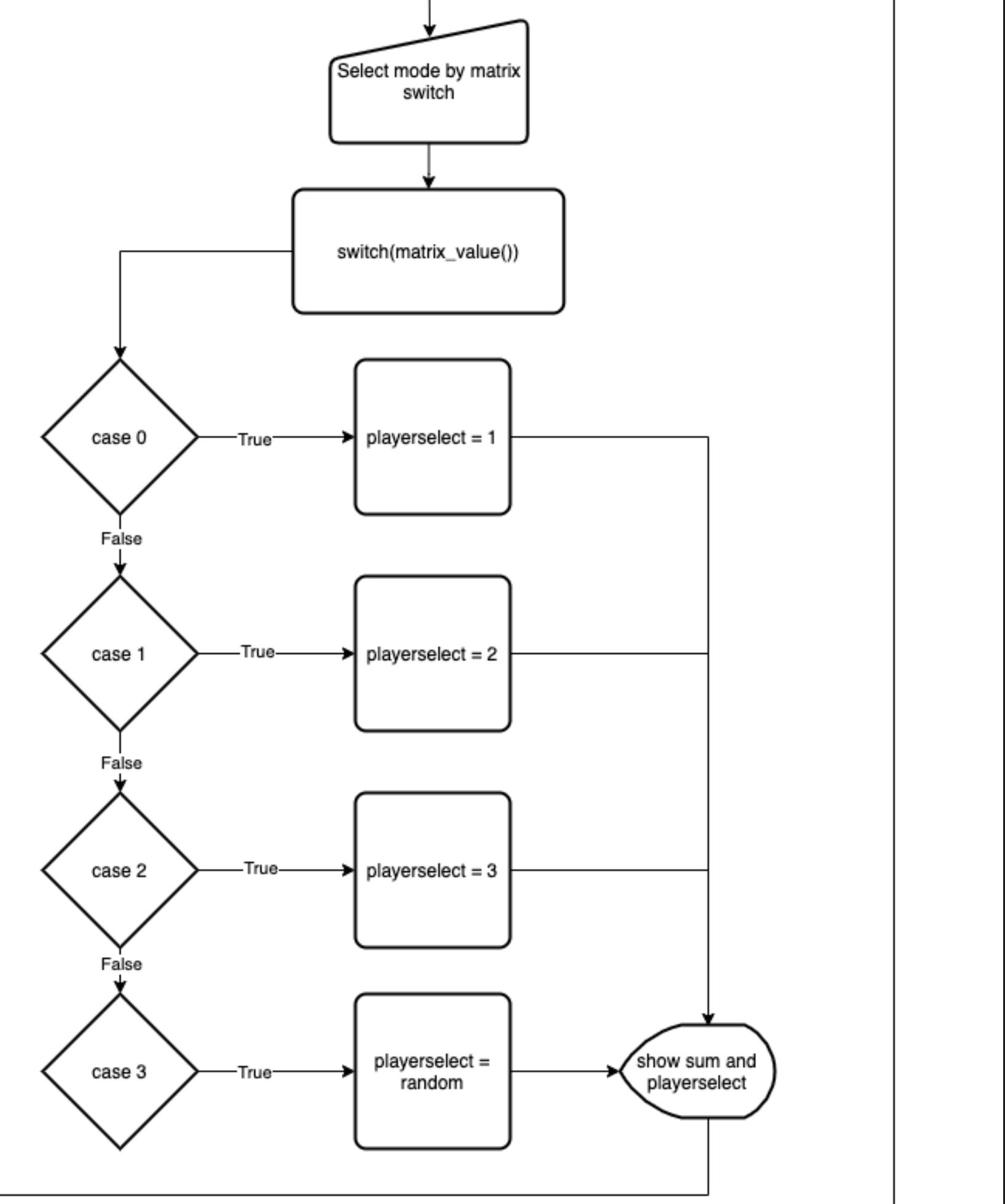
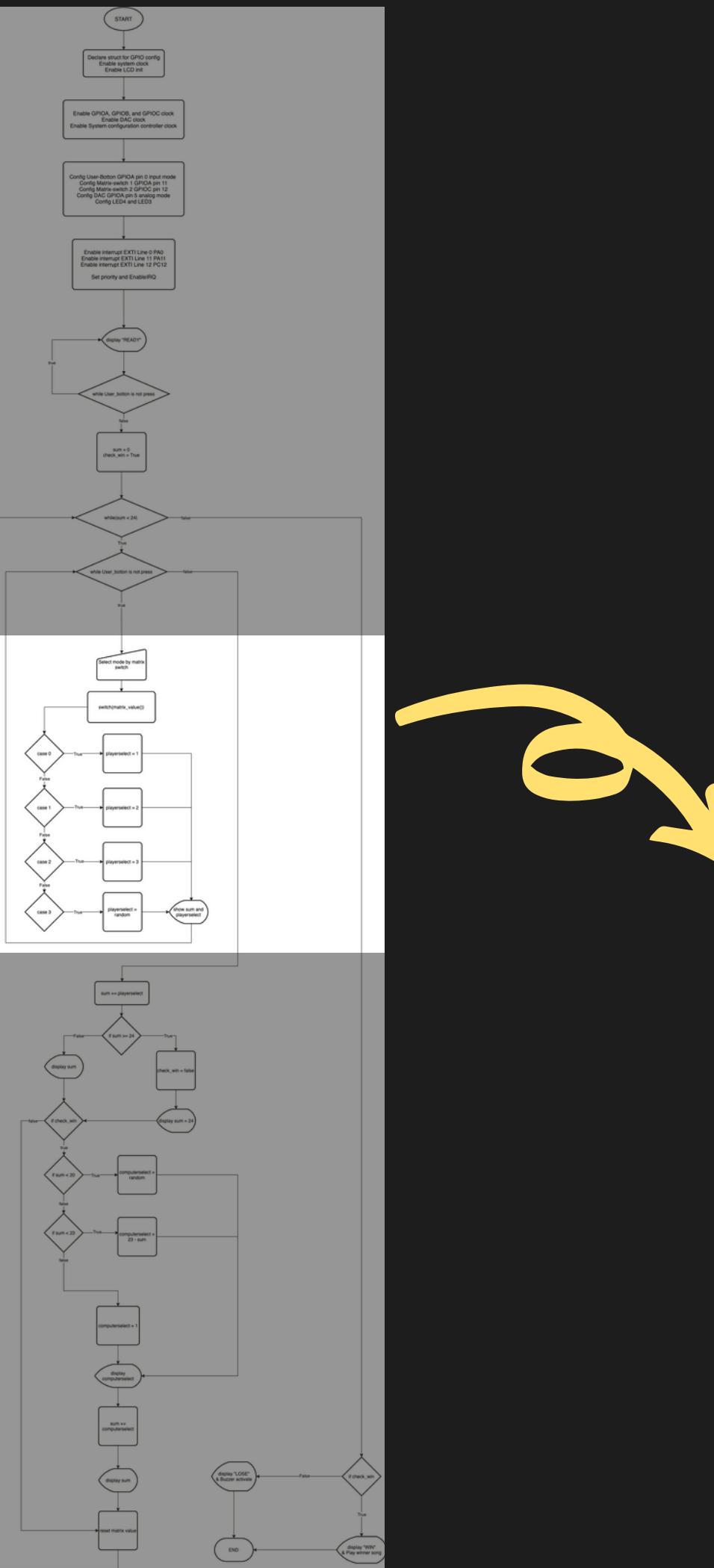


Coding flowchart

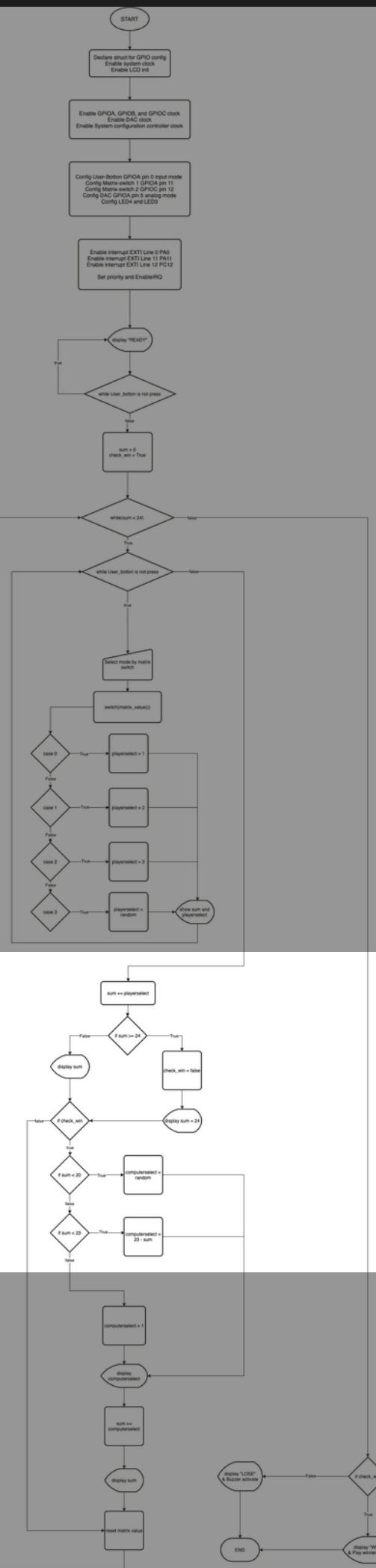
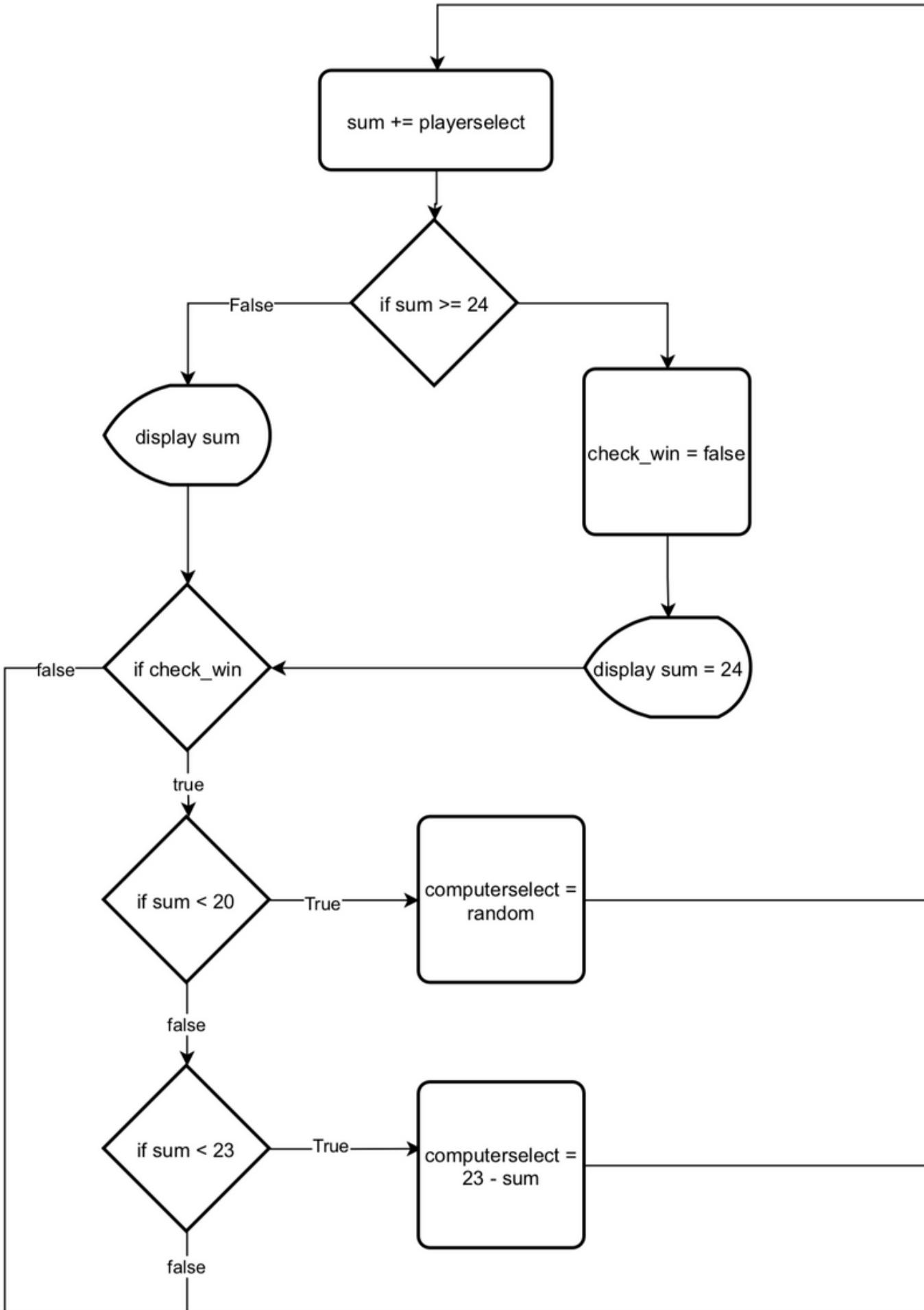
COUNT 24 GAME

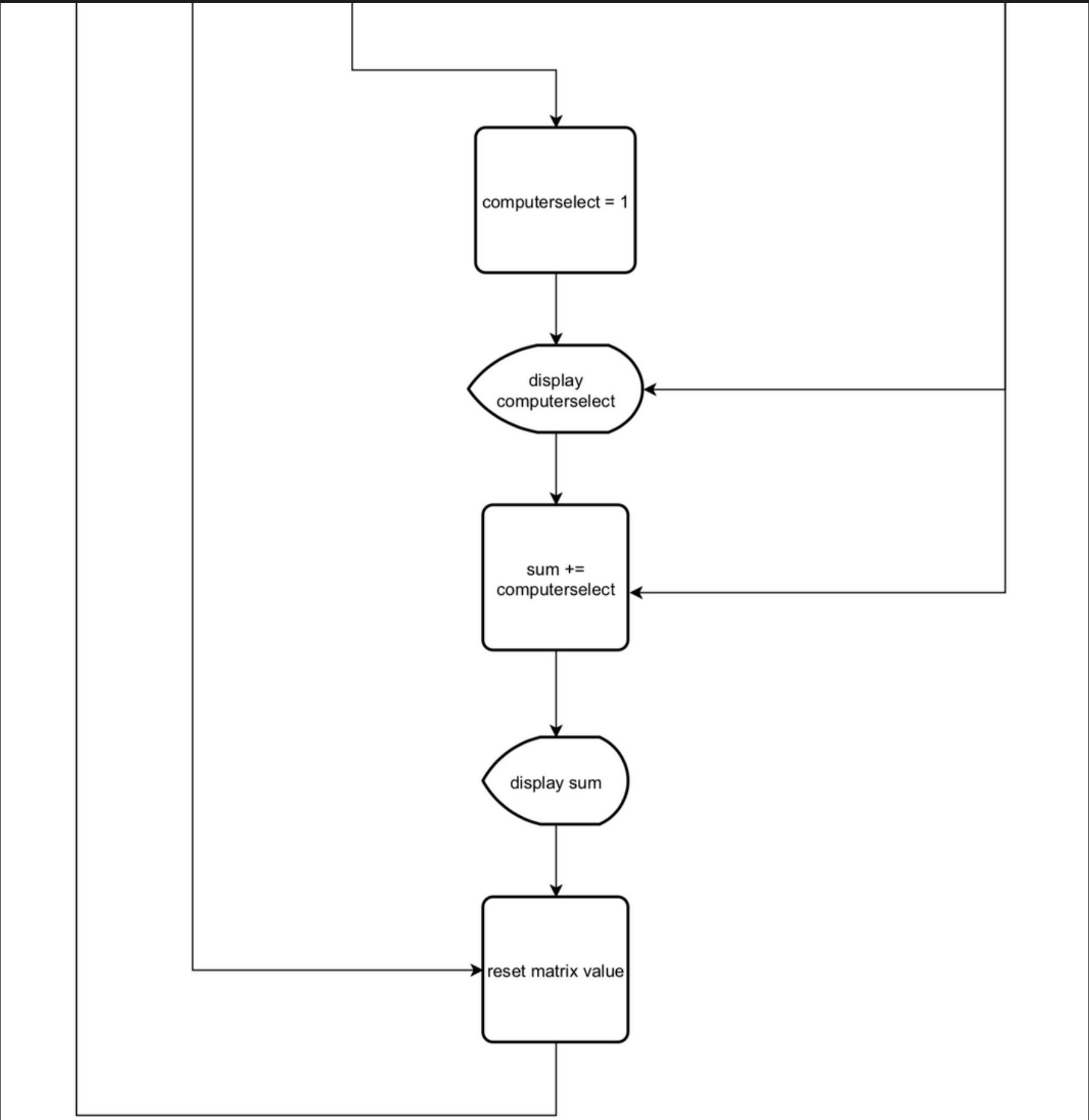
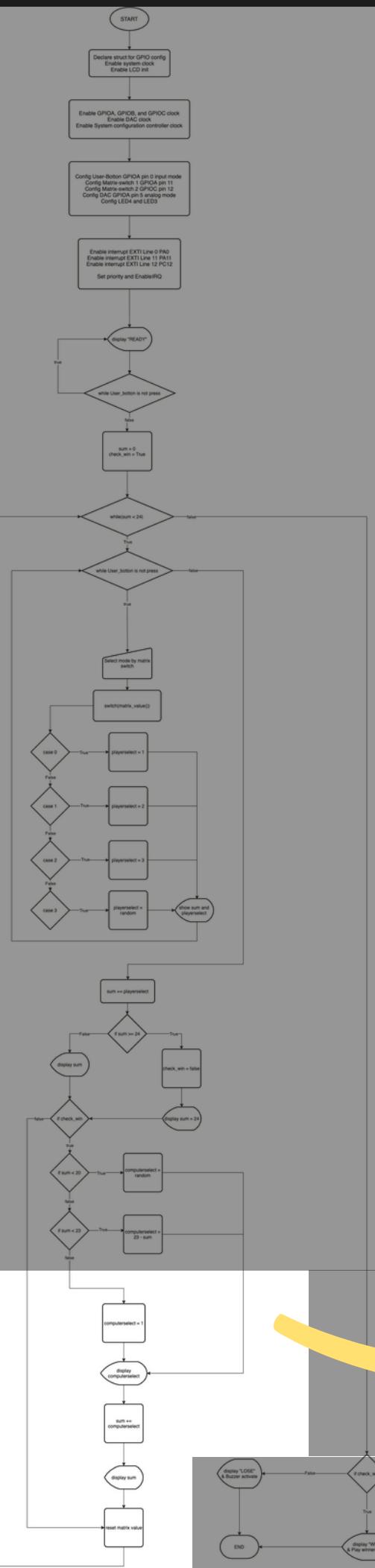






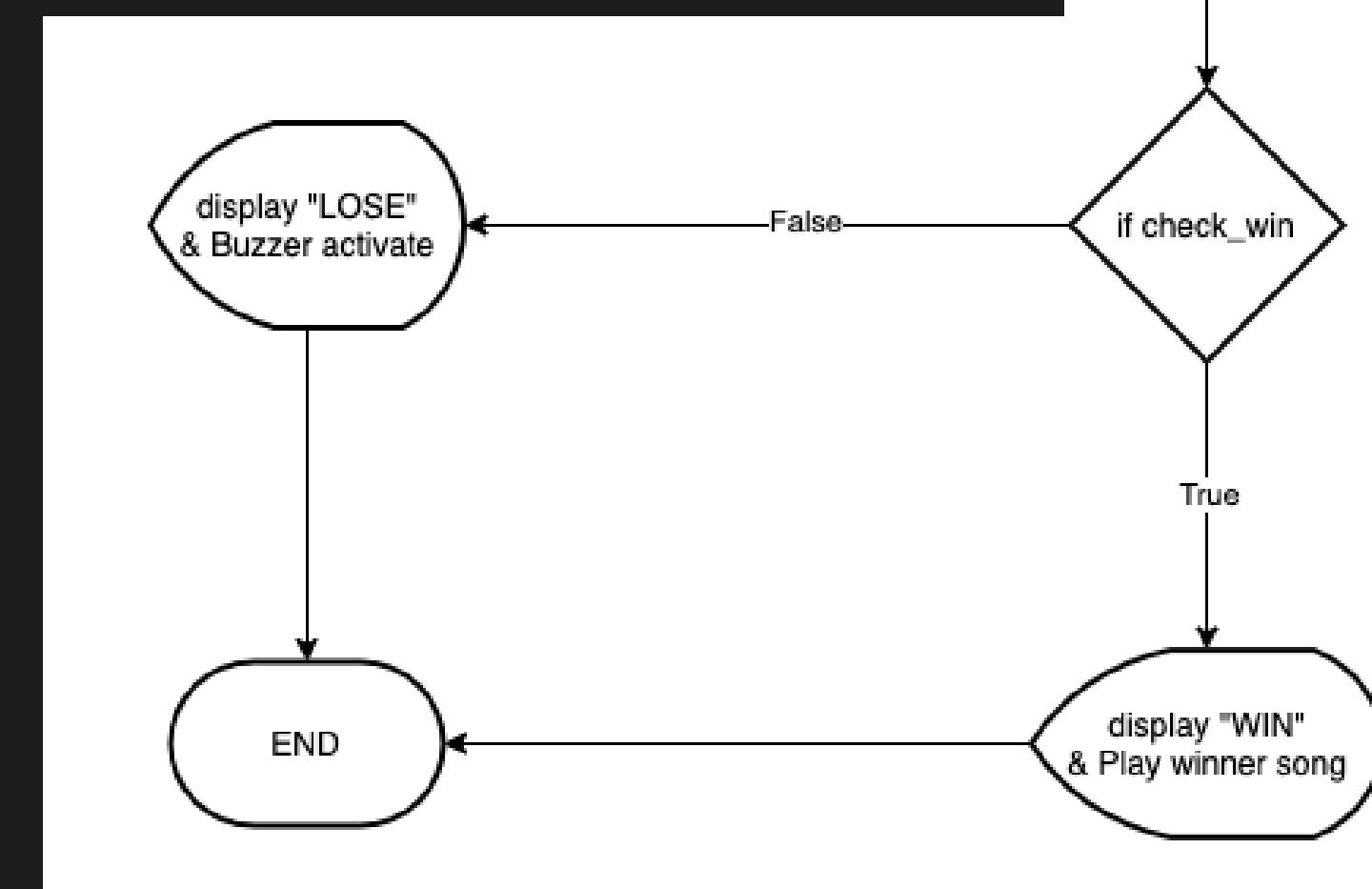
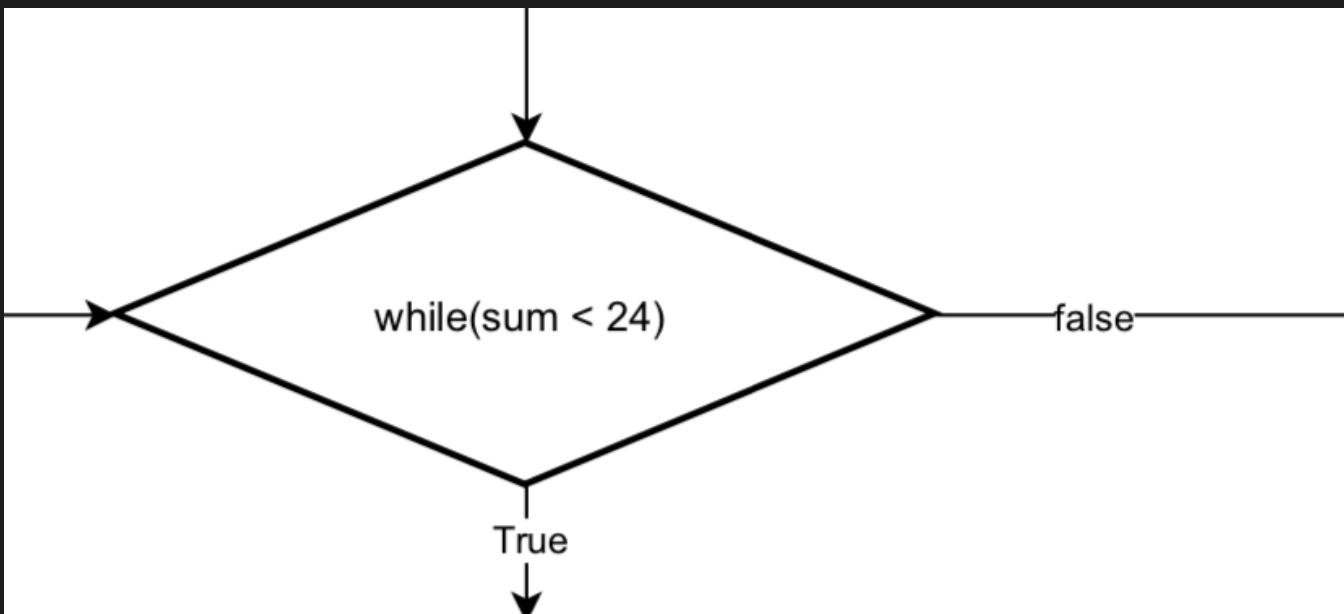
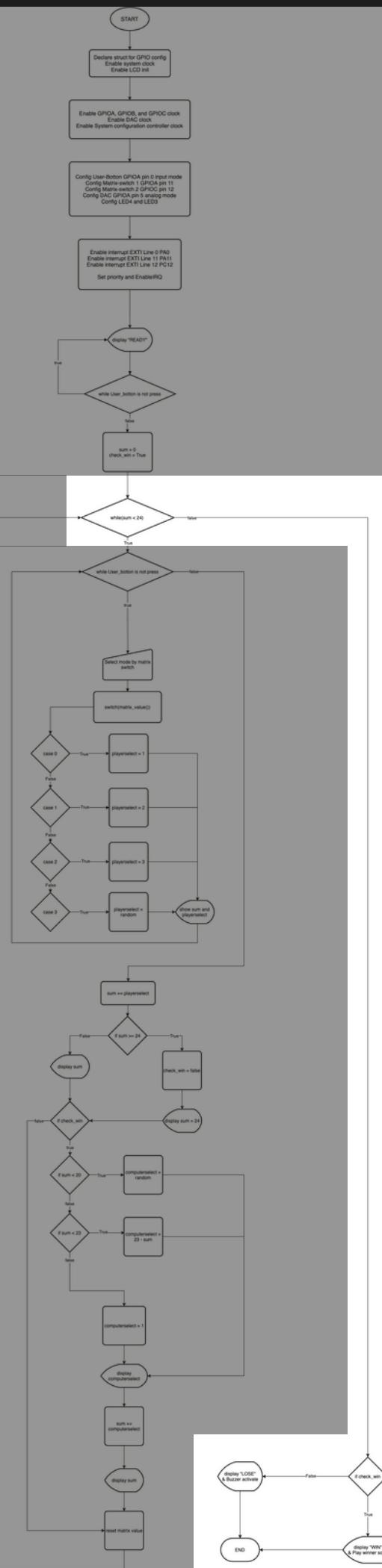
Coding flowchart





Coding flowchart

Coding flowchart



Gantt chart and operating table



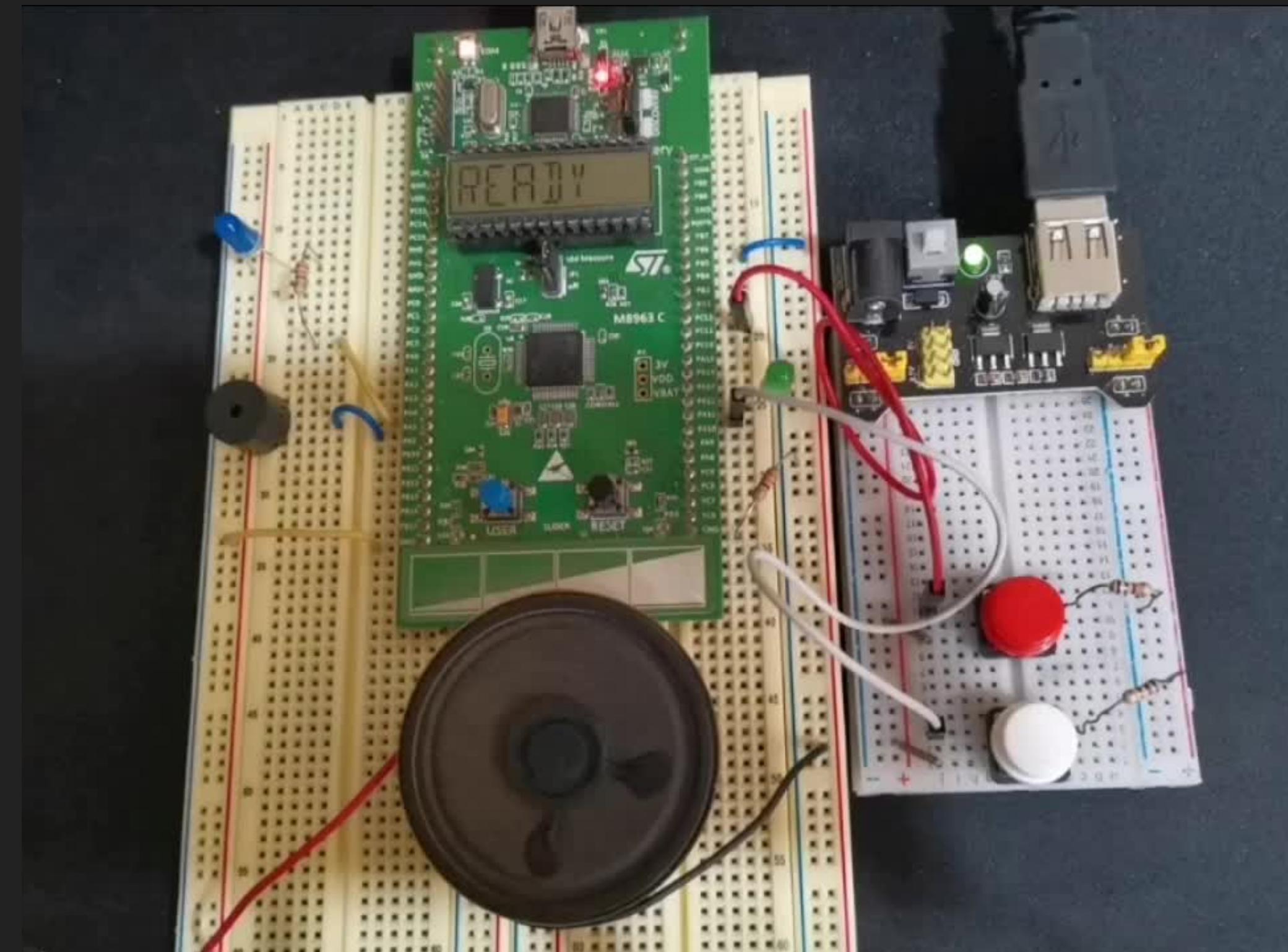
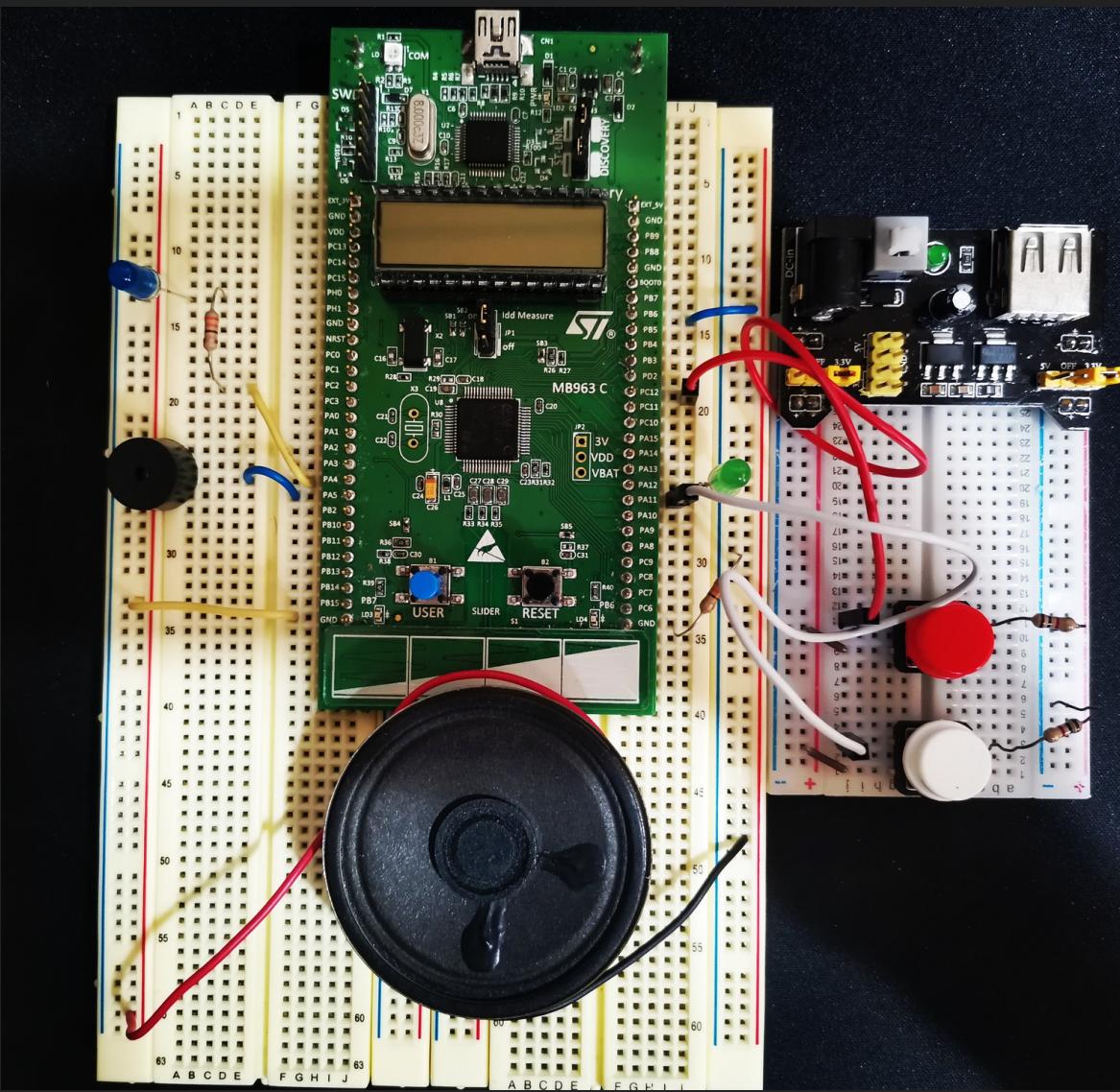
21

Project Plan										
Task	Responsibility	December		January				February		
		3	4	1	2	3	4	1	2	
Requirement & planning	Team member	■								
		■								
Specification	Leader, HW, SW		■							
	Testing team	■	■							
Architectural design	Hardware team		■							
		■	■							
Detailed design	Software team			■						
				■						
Coding	Software team			■	■	■	■			
				■	■	■	■	■		
Unit testing	Testing team				■	■	■	■		
					■	■	■	■	■	
Integration testing	Testing team							■	■	
								■	■	
System testing	Testing team							■	■	
								■	■	
Acceptance testing	Testing team								■	
									■	
Presentation	Marketing							■	■	
								■	■	

6

DEMO VIDEO

22



Count 24 game

...

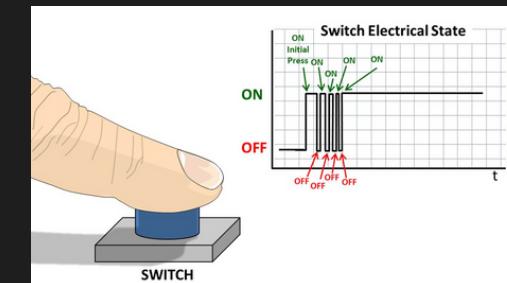
Problem and solution

23

random Number

- random Number generator (RNG) : rand()
- SRAND(time(0)) ➔ SRAND(++i)

Switch Bouncing



soundtrack problem



Problem and solution

p

SOUNDTRACK PROBLEM

24



Soundtrack can play only once per game loop #5

[Open](#) pichayakorn opened this issue 2 days ago · 0 comments

pichayakorn commented 2 days ago · edited

30f3772

s08_speaker_conf.c

```
...
LL_TIM_ClearFlag_UPDATE(TIM2);
LL_TIM_EnableIT_UPDATE(TIM2);

/* Interrupt Configure */
NVIC_SetPriority(TIM2 IRQn, 3);
NVIC_EnableIRQ(TIM2 IRQn);

NVIC_SetPriority(TIM4 IRQn, 4);
NVIC_EnableIRQ(TIM4 IRQn);
LL_TIM_EnableIT_CC1(TIM4);

/* Start Output Compare in PWM Mode */
//LL_TIM_CC_EnableChannel(TIM4, LL_TIM_CHANNEL_CH1);
//LL_TIM_EnableCounter(TIM4);

//LL_TIM_EnableCounter(TIM2);
}

void TIM2_IRQHandler(void) {
    if (LL_TIM_IsActiveFlag_CC1(TIM2) == SET) {
        LL_TIM_ClearFlag_CC1(TIM2);
        if (cur != sizeof(sheetNote)/sizeof(sheetNote[0]) - 1) {
            UPDATE_Keynote(sheetNote[++cur]);
        } else {
            LL_TIM_OC_SetCompareCH1(TIM4, 0); // 0% duty
            RESET_SOUNDTRACK();
    }
}
```

Owner Tip ...

Assignees pichayakorn

Labels bug

Projects Coding In progress

Milestone No milestone

Linked pull requests Successfully merging a pull request may close this issue. None yet

Notifications Unsubscribe

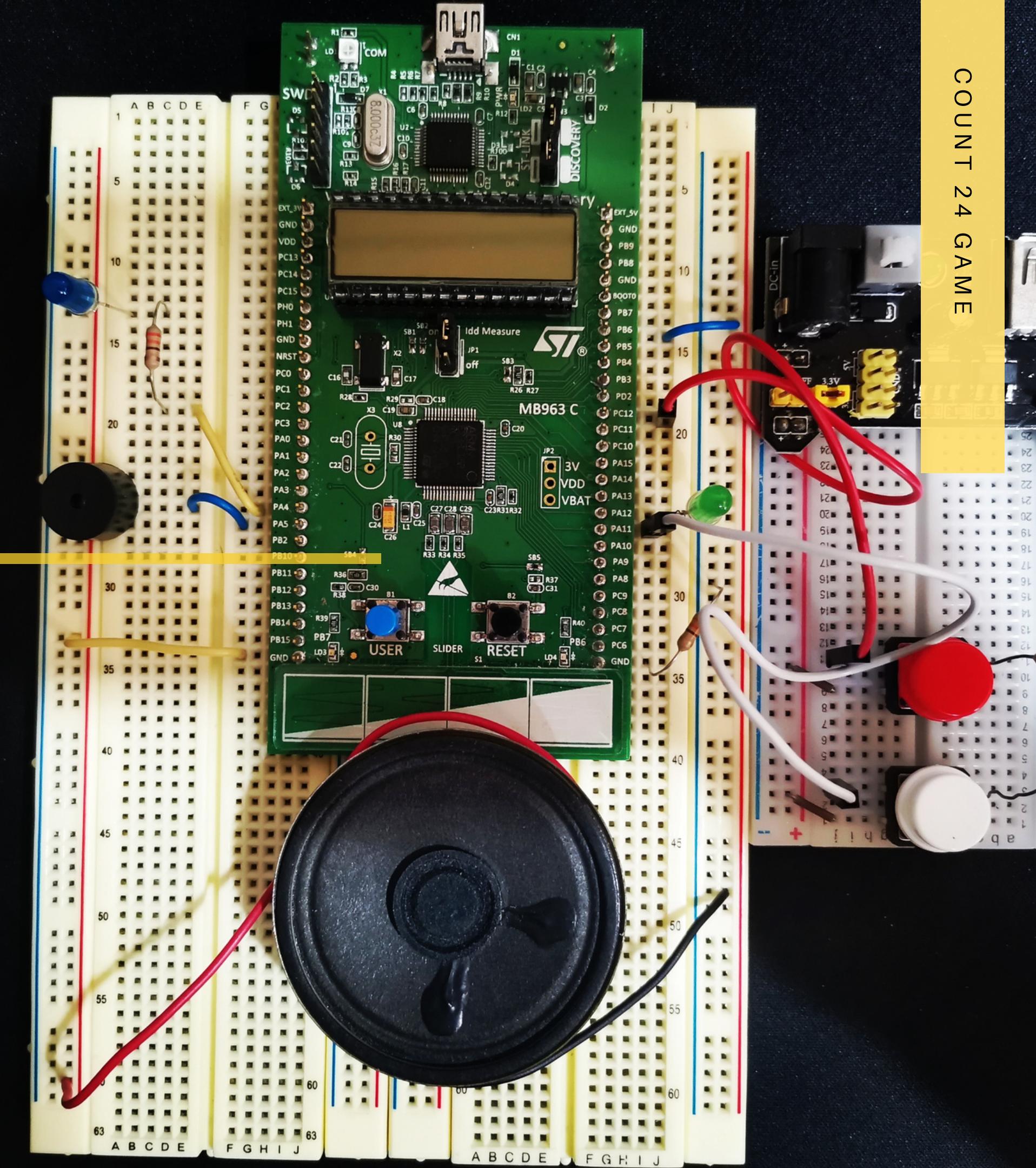
You're receiving notifications because you're watching this repository.

Conclusion

Conclusion and Recommendation

เรียนรู้พื้นฐานและนำไปประยุกต์ใช้งาน

8





Thank you