

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TP.HCM

KHOA ĐIỆN TỬ - VIỄN THÔNG



BÁO CÁO ĐỒ ÁN ÔN TẬP DESLAB

NĂM HỌC: 2020-2021

TÊN ĐỒ ÁN:

TRÒ CHƠI DINO SỬ DỤNG STM32 VÀ OLED

LỚP: 18DTV-CLC

Lê Thành Nghị

BẢNG THỐNG KÊ CẬP NHẬT TÀI LIỆU KỸ THUẬT

Phiên bản	Mô tả	Thành viên	Ngày
1	Tạo tài liệu	L.T.Nghị	23/04/2021
0.1	Cập nhật phần 3	L.T.Nghị	24/04/2021
0.2	Cập nhật phần 1, 2, 4	L.T.Nghị	25/04/2021
0.3	Cập nhật hình ảnh, chú thích ảnh	L.T.Nghị	26/04/2021

MỤC LỤC

1. MỞ ĐẦU	4
2. GIỚI THIỆU – LÝ THUYẾT	5
2.1 STM32	5
2.2 OLED 128x64 1.3 inch driver SH1106	13
2.3 Nút nhấn	13
3. SƠ ĐỒ KHỐI VÀ HOẠT ĐỘNG	14
4. HÌNH ẢNH	24
4.1 Hình ảnh thiết kế	24
4.2 Hình ảnh trong trò chơi	25
5. THAM KHẢO	34

MỤC LỤC HÌNH

Hình 1 – Phân loại STM32Fxx	5
Hình 2 – Thông số và các chức năng của STM32F103xx	6
Hình 3 – Kiến trúc của vi điều khiển STM32F103xx	7
Hình 4 – Cấu trúc và các thanh ghi của bộ Timer	8
Hình 5 – Mô hình kết nối sử dụng giao tiếp SPI	11
Hình 6 – Mạch nạp ST-LINK V2	12
Hình 7 – OLED SPI	13
Hình 8 – Nút nhấn 6x6mm	13
Hình 9 – Sơ đồ khối main function	14
Hình 10 – Sơ đồ khối dino_work function	15
Hình 11 – Sơ đồ khối cactus_work function	16
Hình 12 – Sơ đồ khối bird_work function	17
Hình 13 – Sơ đồ khối bird_wing_work function	18
Hình 14 – Sơ đồ khối sword_work function	20
Hình 15 – Sơ đồ khối display function	21
Hình 16 – Sơ đồ khối dino_dead function	22
Hình 17 – Kết nối các chân của STM32	23
Hình 18 – Hình ảnh thực tế bộ trò chơi	24
Hình 19 – Hình ảnh giao diện trò chơi	25
Hình 20 – Hình ảnh dino nhảy lên	26
Hình 21 – Hình ảnh dino chết vì va chạm với xương rồng	27
Hình 22 – Hình ảnh kiểm xuất hiện	28
Hình 23 – Hình ảnh kiểm xuất hiện (ban đêm)	29
Hình 24 – Hình ảnh dino cầm kiếm	30
Hình 25 – Hình ảnh dino chém cactus	31
Hình 26 – Hình ảnh chim xuất hiện	32
Hình 27 – Hình ảnh dino chết vì va chạm với chim	33

1. MỞ ĐẦU

Vận dụng những kiến thức đã học được từ những buổi ôn tập trên DESLAB, em có ý tưởng làm ra một bộ trò chơi dino hiển thị lên OLED, sử dụng vi điều khiển STM32.

Hệ thống bao gồm màn hình OLED 128x64 1.3 inch giao tiếp chuẩn SPI để hiển thị hình ảnh trò chơi, vi điều khiển STM32F103RCT6 để xử lý và 2 nút bấm để điều khiển nhân vật trong trò chơi.

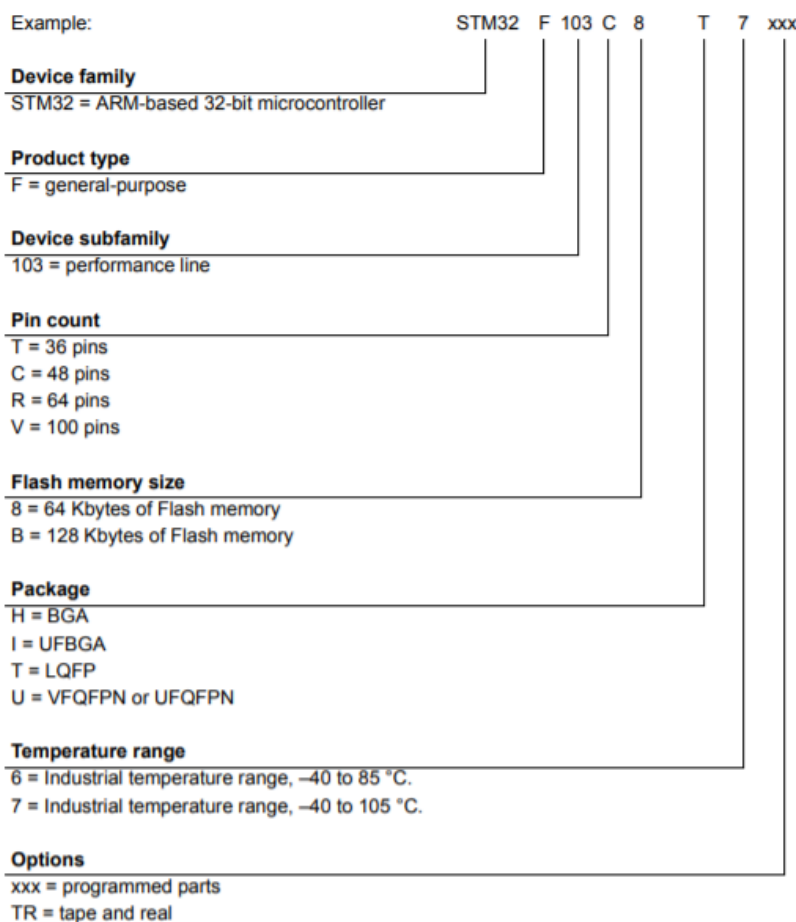
2. GIỚI THIỆU – LÝ THUYẾT

2.1 STM32

STM32 là một họ vi điều khiển 32-bit dựa trên bộ xử lý Arm® Cortex®-M 32-bit được phát triển bởi STMicroelectronics. STM32 cung cấp hiệu năng cao, khả năng xử lý thời gian thực, khả năng xử lý tín hiệu số, điện năng tiêu thụ thấp và khả năng kết nối (Wifi, Bluetooth, LoRa, Ethernet...) trong khi vẫn được sự hội nhập và dễ dàng phát triển.

STM32F103 thuộc họ F1 với lõi là ARM COTEX M3, là vi điều khiển 32 bit, tốc độ tối đa là 72Mhz. Giá thành cũng khá rẻ so với các loại vi điều khiển có chức năng tương tự. Mạch nạp cũng như công cụ lập trình khá đa dạng và dễ sử dụng.

Kit phát triển STM32F103RCT6 ARM Cortex-M3 Mini được thiết kế với các phần thí nghiệm cơ bản và ra chân đầy đủ vi điều khiển STM32F103RCT6.



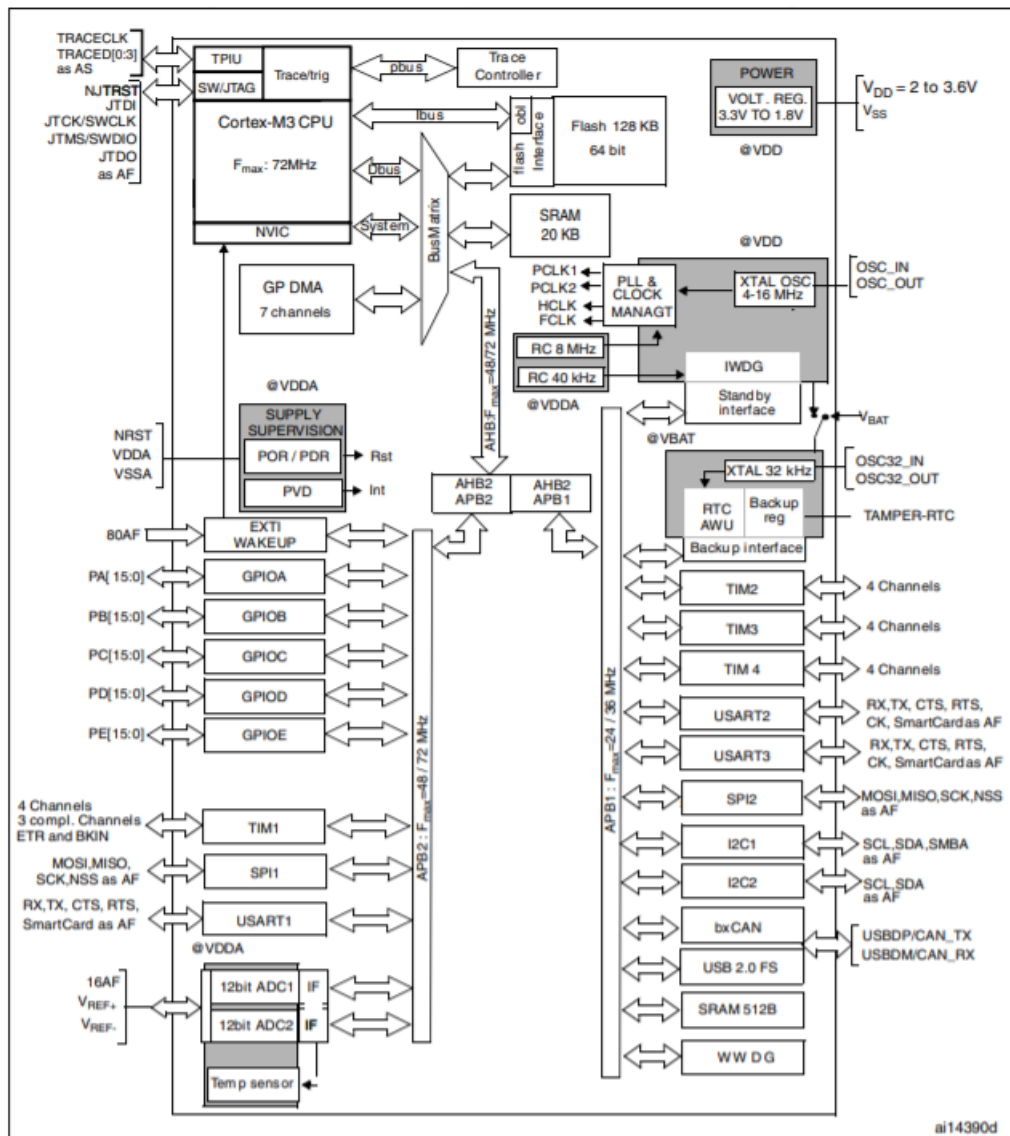
Hình 1 – Phân loại STM32Fxx

Thông số của một số vi điều khiển dòng STM32F103:

Peripheral		STM32F103Tx		STM32F103Cx		STM32F103Rx		STM32F103Vx	
Flash - Kbytes		64	128	64	128	64	128	64	128
SRAM - Kbytes		20		20		20		20	
Timers	General-purpose	3		3		3		3	
	Advanced-control	1		1		1		1	
Communication	SPI	1		2		2		2	
	I ² C	1		2		2		2	
	USART	2		3		3		3	
	USB	1		1		1		1	
	CAN	1		1		1		1	
GPIOs		26		37		51		80	
12-bit synchronized ADC		2		2		2		2	
Number of channels		10 channels		10 channels		16 channels ⁽¹⁾		16 channels	
CPU frequency		72 MHz							
Operating voltage		2.0 to 3.6 V							
Operating temperatures		Ambient temperatures: -40 to +85 °C / -40 to +105 °C Junction temperature: -40 to + 125 °C							
Packages		VFQFPN36		LQFP48, UFQFPN48		LQFP64, TFBGA64		LQFP100, LFBGA100, UFBGA100	

Hình 2 – Thông số và các chức năng của STM32F103xx

Sơ đồ khối các thành phần, kiến trúc của STM32F103xx:



1. $T_A = -40\text{ }^{\circ}\text{C}$ to $+105\text{ }^{\circ}\text{C}$ (junction temperature up to $125\text{ }^{\circ}\text{C}$).
2. AF = alternate function on I/O port pin.

Hình 3 – Kiến trúc của vi điều khiển STM32F103xx

Trong đề tài này, khi sử dụng STM32, nhóm có sử dụng một số chức năng như ngắt timer, SPI. Các chức năng này sẽ được trình bày rõ hơn như sau.

2.1.1 Timer

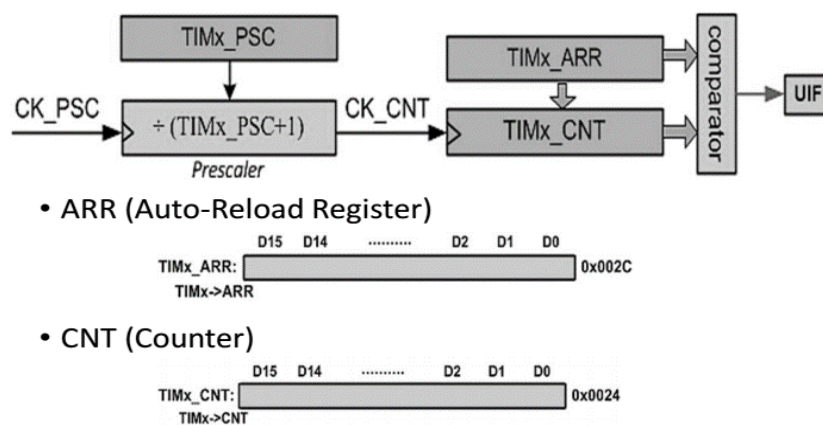
Timer (bộ định thời) là bộ đếm thời gian, được sử dụng hầu như trong tất cả các hệ thống, phục vụ cho các công việc tính toán liên quan đến thời gian. Timer là một module trong chip, được cấp xung clock từ thạch anh bên ngoài chip. Độ chính xác của timer có liên quan mật thiết từ độ chính xác của xung clock ngõ vào (do thạch anh cung cấp).

External sources (nguồn ngoài)

- HSE (High Speed External): Bộ dao động HSE chấp nhận tần số ngõ vào từ 4 – 25 MHz. Nguồn clock chủ yếu là bộ dao động thạch anh hoặc mạch tạo tín hiệu xung.
- LSE (Low Speed External): có chức năng tương tự như bộ dao động HSE nhưng chấp nhận tần số ngõ vào bằng 32.768 kHz.
- LSE được sử dụng để cấp nguồn cho module Real Time Clock (RTC) bên trong STM32.

Internal sources (nguồn nội)

- HSI (High Speed Internal): nguồn clock cố định cấp cho HSI bằng 8Mhz
- LSI (Low Speed Internal): input frequency dao động từ 30-60 kHz. Tần số phổ biến 45 kHz.
- PLL (Phase-Lock-Loop): bộ nhân tần số HSE hoặc HSI lên nhiều lần, nhằm tạo ra giá trị lớn hơn cho clock.



Hình 4 – Cấu trúc và các thanh ghi của bộ Timer

Tần số TIM_CNT:

$$f_{CK_CNT} = \frac{f_{CK_PSC}}{PSC + 1}$$

- fCK_CNT: tần số sau bộ chia Prescaler.
- fCK_PSC: tần số clock đầu vào cấp cho timer.
- PSC: giá trị nạp vào cho bộ chia tần số của timer. Tối đa là 65535, giá trị này được lập trình bằng phần mềm.

Tần số Timer:

$$f_{TIMER} = \frac{f_{SYSTEM}}{(PSC + 1)(Period + 1)}$$

- fTIMER: là tần số Timer, đơn vị là hz.
- fSYSTEM: tần số clock hệ thống được chia cho timer sử dụng, đơn vị là Hz.
- Period: giá trị bộ đếm nạp vào cho timer. Tối đa là 65535.

Các thanh ghi có trong bộ Timer

TIMx_CNT – counter register: Thanh ghi này chứa giá trị của bộ đếm timer. Giá trị tối đa là 65535 (16-bit).

[illegible]

TIMx_PSC – Prescaler register: Thanh ghi này chứa giá trị của bộ chia tần số của timer. Giá trị chia tần số phụ thuộc vào giá trị của thanh ghi.

[illegible]

TIMx_ARR – Auto-reload Register: Thanh ghi này chứa giá trị đích của bộ đếm. Giá trị này sẽ được so sánh với giá trị của thanh ghi CNT (16-bit)

Address offset: 0x2C
Reset value: 0xFFFF

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

TIMx_SR - Status register: Chứa các cờ biểu thị các trạng thái của timer

Address offset: 0x10
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CC4OF	CC3OF	CC2OF	CC1OF	Res.	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			rc_w0	rc_w0	rc_w0	rc_w0	Res.	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

- CCxOF: vượt quá giá trị compare/capture.
- BIF: báo có ngắt đầu vào xảy ra.
- TIF: báo có ngắt do xung trigger.
- COMIF: báo có ngắt do tác động của COM.
- CCxIF: báo có ngắt do compare/capture.
- UIF: báo có ngắt do có sự cập nhật giá trị của bộ đếm timer.

TIMx_CRx - Control register: Thanh ghi điều khiển

Address offset: 0x00
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CKD[1:0]	ARPE	CMS[1:0]	DIR	OPM	URS	UDIS	CEN		
						rw	rw	rw	rw	rw	rw	rw	rw		

- CKD[1:0] : bit liên quan đến việc sinh ra thời gian dead-time và bộ lọc số (ở mode ETR,TIx).
- ARPE: có cho phép tự động load lại khi tràn hay không.
- CMS[1:0] : bit cài đặt mode center-aligned.
- DIR : bit điều khiển timer đếm lên hay đếm xuống.
- OPM: bit sử dụng cho chế độ one-pulse.
- URS: bit này cho phép cập nhật bộ đếm timer theo chế độ điều khiển bằng tín hiệu bên ngoài hoặc bộ đếm bị tràn.
- UDIS: bit này cho phép hoặc không cho phép việc cập nhật timer.
- CEN: bit này cho phép hoặc không cho phép bộ đếm hoạt động.

2.1.2 Ngắt Timer

Ngắt Timer là sự kiện ngắt được phát sinh từ bộ Timer, do đếm tràn Timer hoặc giá trị đếm trong bộ Timer trùng với giá trị của bộ Capture/Compare.

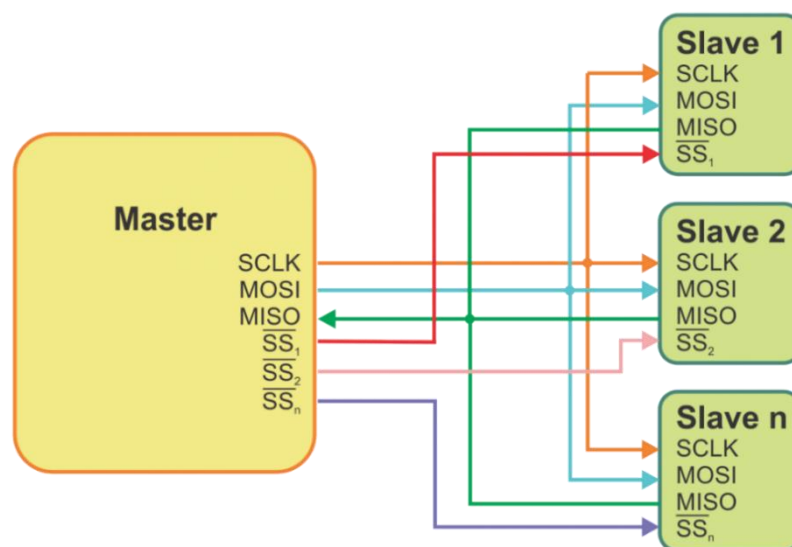
Việc cấu hình các thông số Timer vẫn giống như cấu hình Timer bình thường. Tuy nhiên phải cấu hình thêm NVIC. Trong báo cáo này em sẽ dùng ngắt timer khi tràn.

2.1.3 SPI

SPI (Serial peripheral interface) là một chuẩn truyền dữ liệu nối tiếp ở chế độ song công hoàn toàn (full-duplex) do công ty Motorola giới thiệu. Giao tiếp SPI gồm 4 dây tín hiệu trên đường truyền:

- SCLK – Serial Clock: Dây tín hiệu đồng bộ. Tín hiệu này do thiết bị đóng vai trò master (chủ) cung cấp. Các thiết bị slave (tớ) chỉ nhận tín hiệu clock từ master để đồng bộ hoạt động.
- MOSI – Master Out Slave In: Dây truyền dữ liệu đi từ master vào slave.
- MISO – Master In Slave Out: Dây truyền dữ liệu từ slave đến master.
- CS/SS – Chip Select/Slave Select: Chân chọn slave (cho phép slave hoạt động) do một master có thể giao tiếp với nhiều slave.

⇒ Nhờ vào việc phân định hai luồng dữ liệu vào/ra bằng hai tín hiệu độc lập (MOSI/MISO) nên SPI hỗ trợ truyền nhận dữ liệu theo cơ chế song công hoàn toàn (vừa gửi vừa nhận) mà không bị xung đột dữ liệu trên đường truyền.



Hình 5 – Mô hình kết nối sử dụng giao tiếp SPI

2.1.4 ST-Link V2

ST-LINK V2 là một trình gỡ lỗi (debugger) trong mạch và lập trình (programmer) cho các họ vi điều khiển STM8 và STM32, dùng để, nạp code lên các bộ vi điều khiển STM8 & STM32. Các giao diện module dây đơn (SWIM) và JTAG / gỡ lỗi dây nối tiếp (SWD) được sử dụng để giao tiếp với bất kỳ vi điều khiển STM32 nào trên board ứng dụng. Có thể được giao tiếp với phần mềm Keil nơi chương trình có thể được Flash vào bộ vi điều khiển STM32.



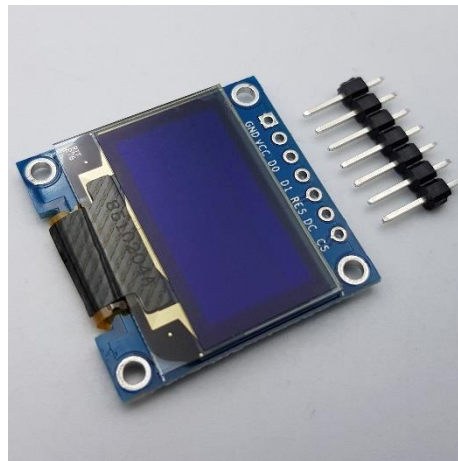
Hình 6 – Mạch nạp ST-LINK V2

2.2 OLED 128x64 1.3 inch driver SH1106

Màn hình OLED SH1106 với kích thước 1.3 inch, cho khả năng hiển thị hình ảnh tốt với khung hình 128x64 pixel, tương thích với hầu hết các vi điều khiển hiện nay thông qua giao tiếp SPI.

Thông số kỹ thuật

- Điện áp hoạt động: 3.3V đến 5V DC
- Điện năng tiêu thụ: 0,04W
- Kích thước 1.3 inch (128x64px)
- Màu chữ: trắng
- Góc nhìn tối đa: 160°
- Driver: SH1106
- Nhiệt độ làm việc: -30°C đến 80°C
- Chuẩn giao tiếp SPI



Hình 7 – OLED SPI

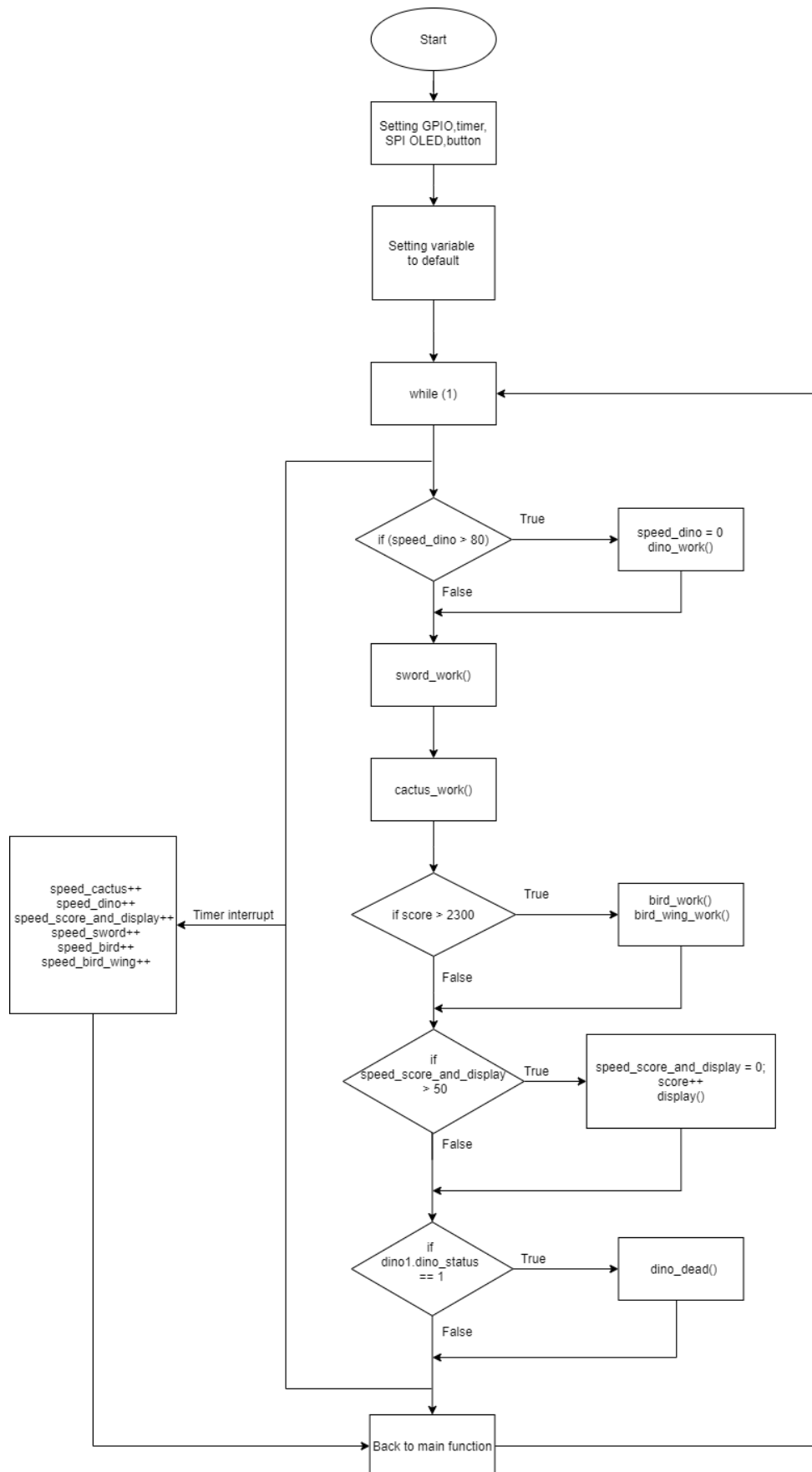
2.3 Nút nhấn

Nút nhấn 6x6mm có thiết kế nhỏ gọn, dễ kết nối và sử dụng, phù hợp cho các ứng dụng điều khiển bằng phím bấm.

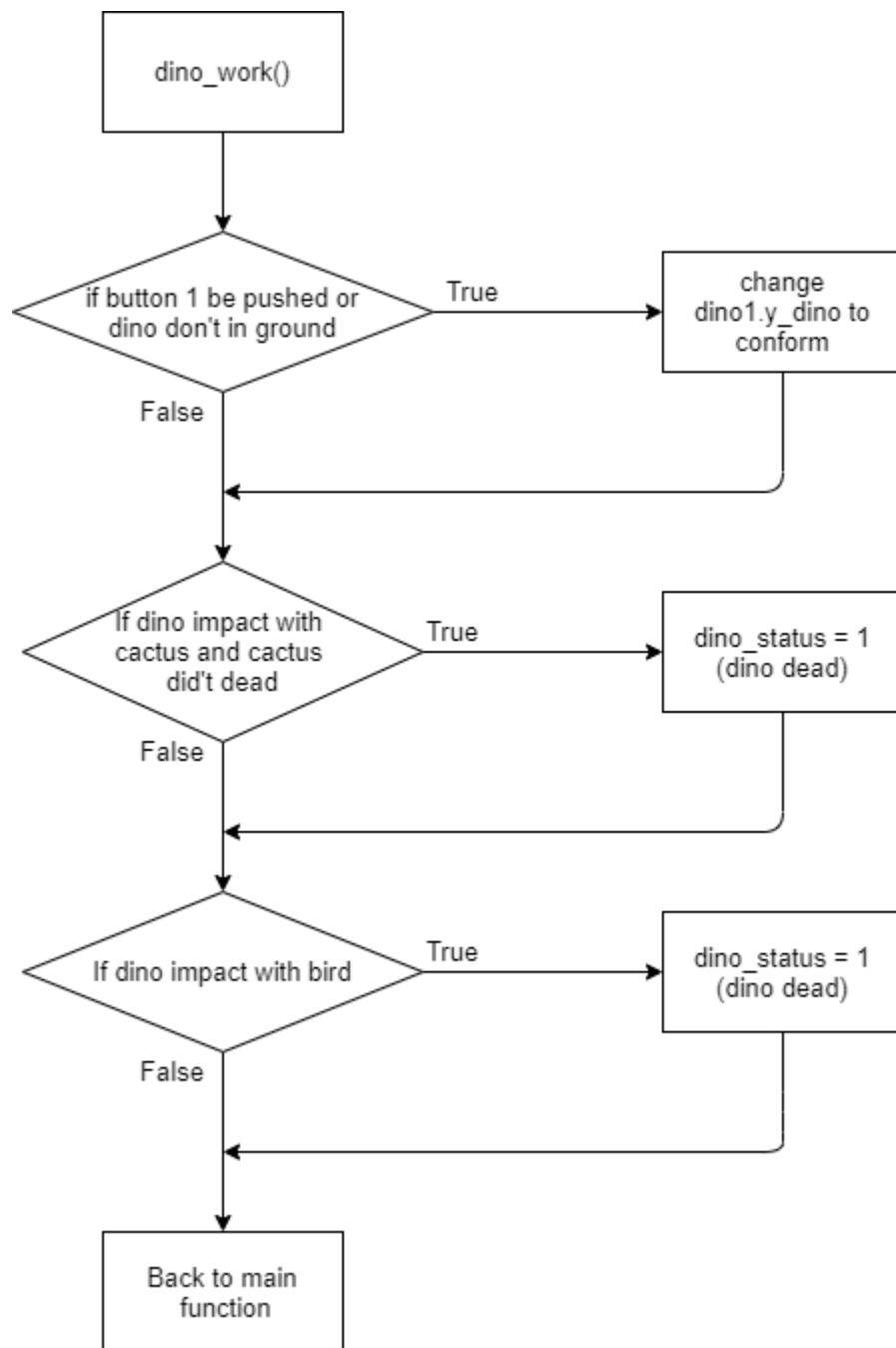


Hình 8 – Nút nhấn 6x6mm

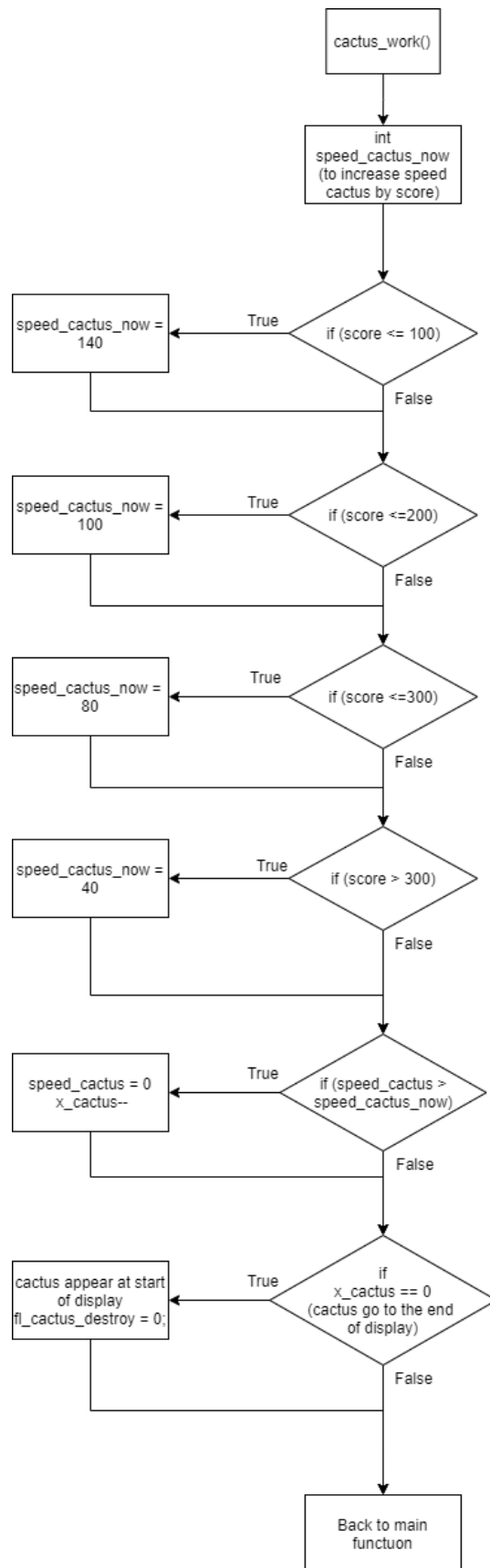
3. SƠ ĐỒ KHỞI VÀ HOẠT ĐỘNG



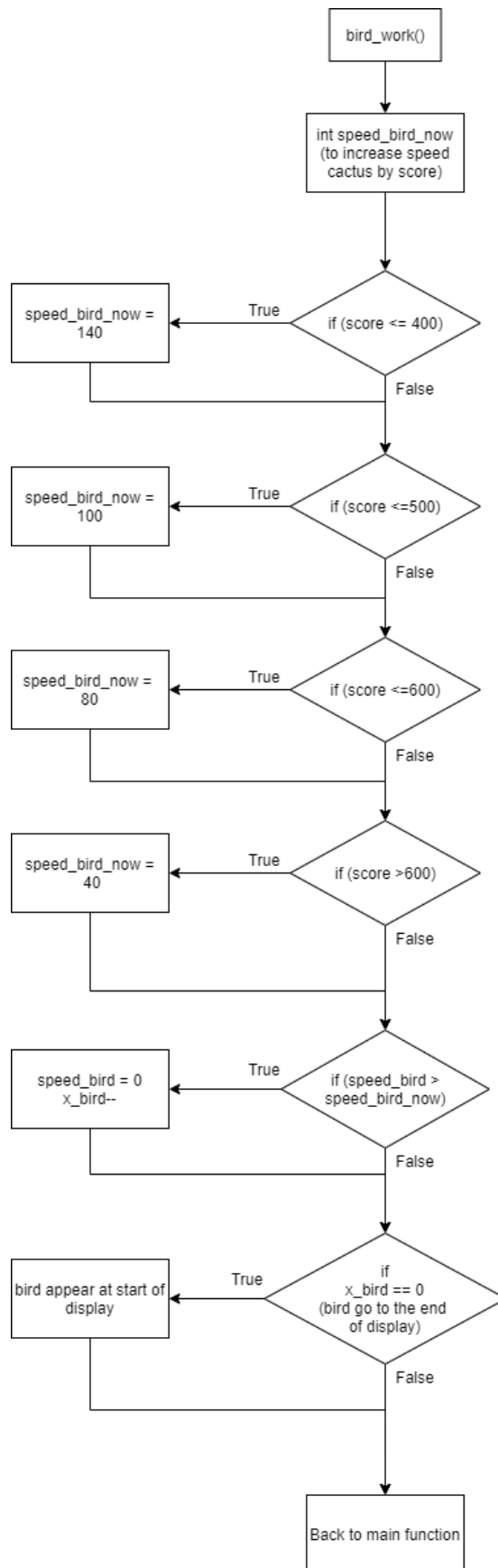
Hình 9 – Sơ đồ khối main function



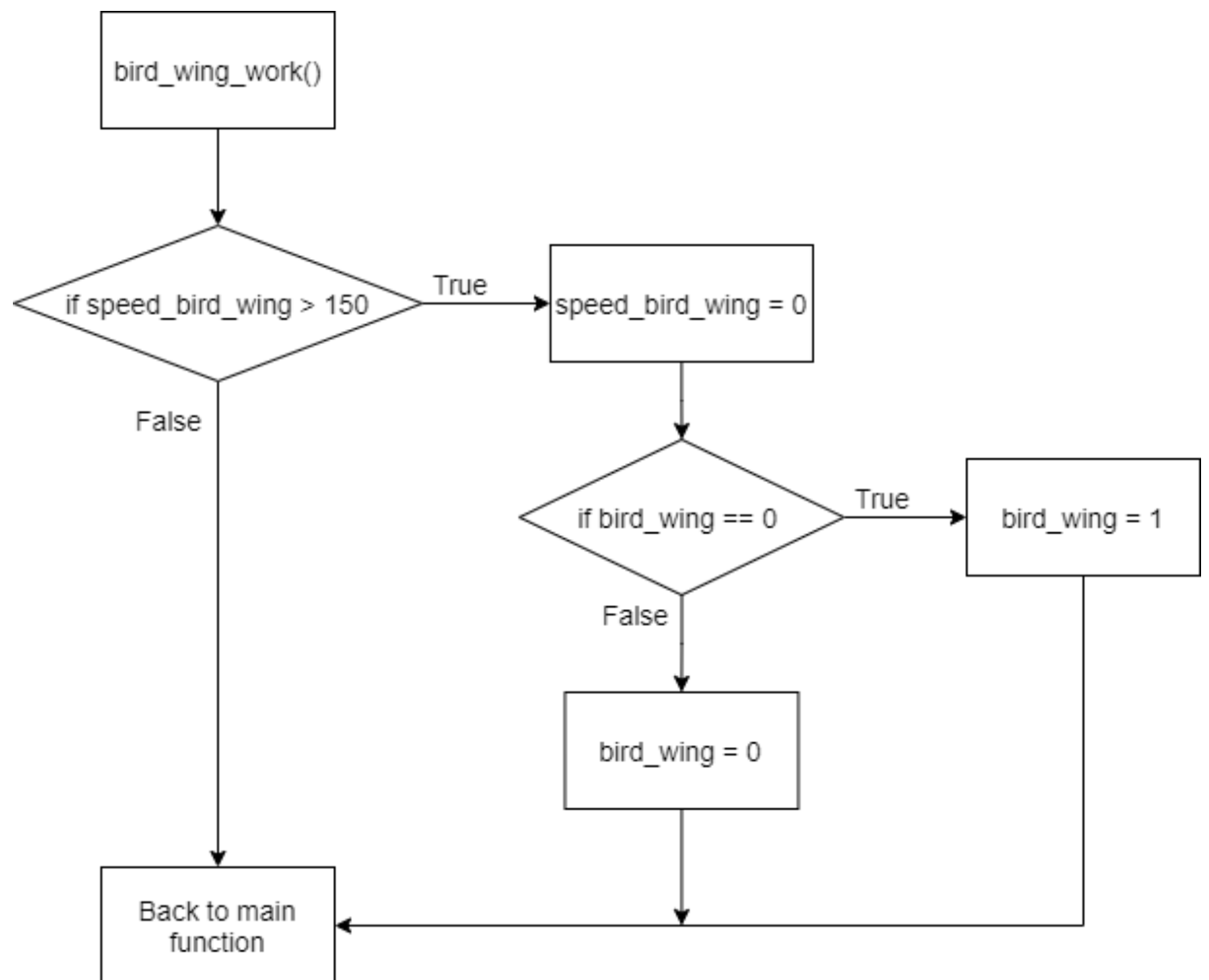
Hình 10 – Sơ đồ khối `dino_work function`



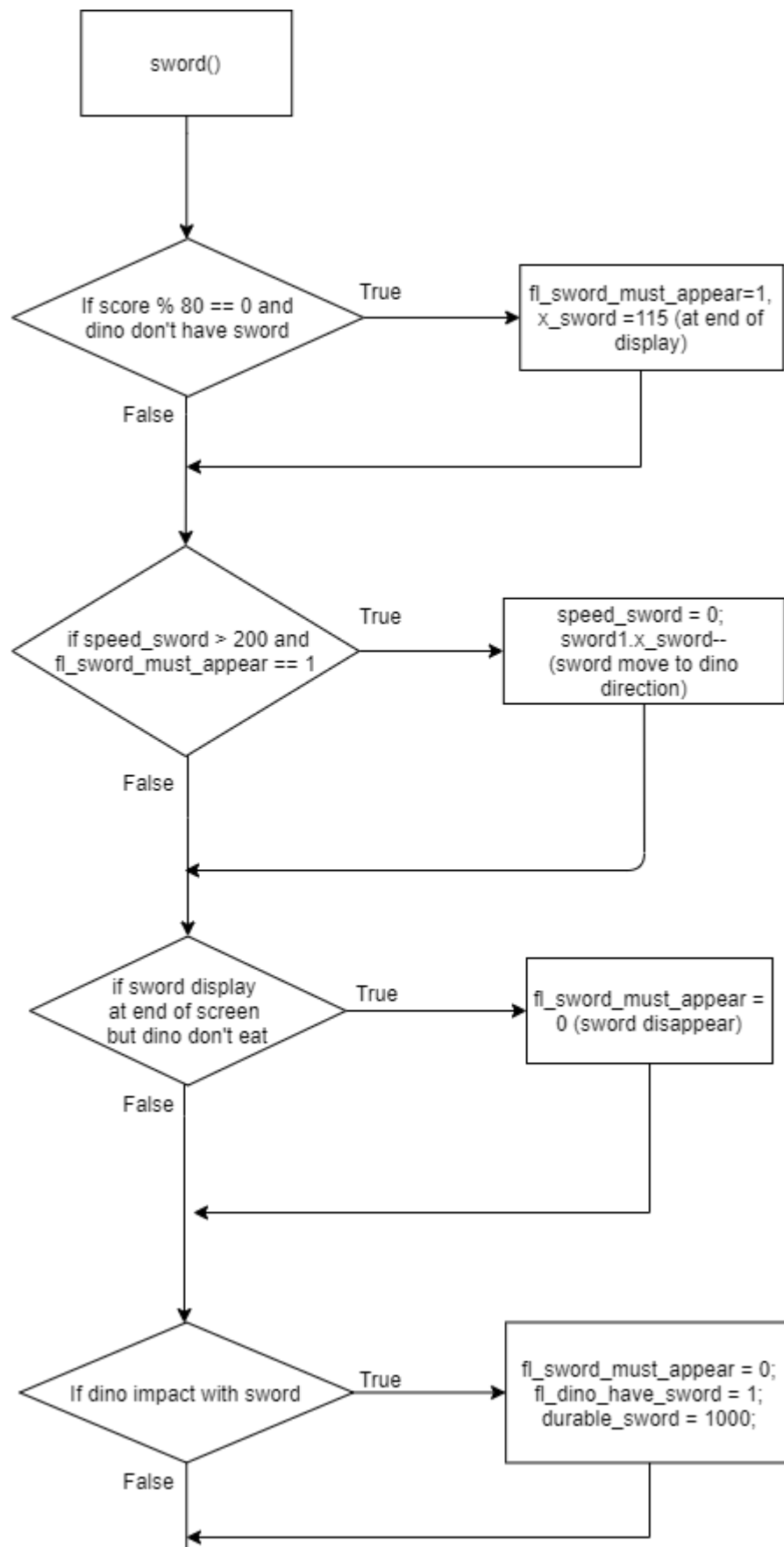
Hình 11 – Sơ đồ khối `cactus_work` function

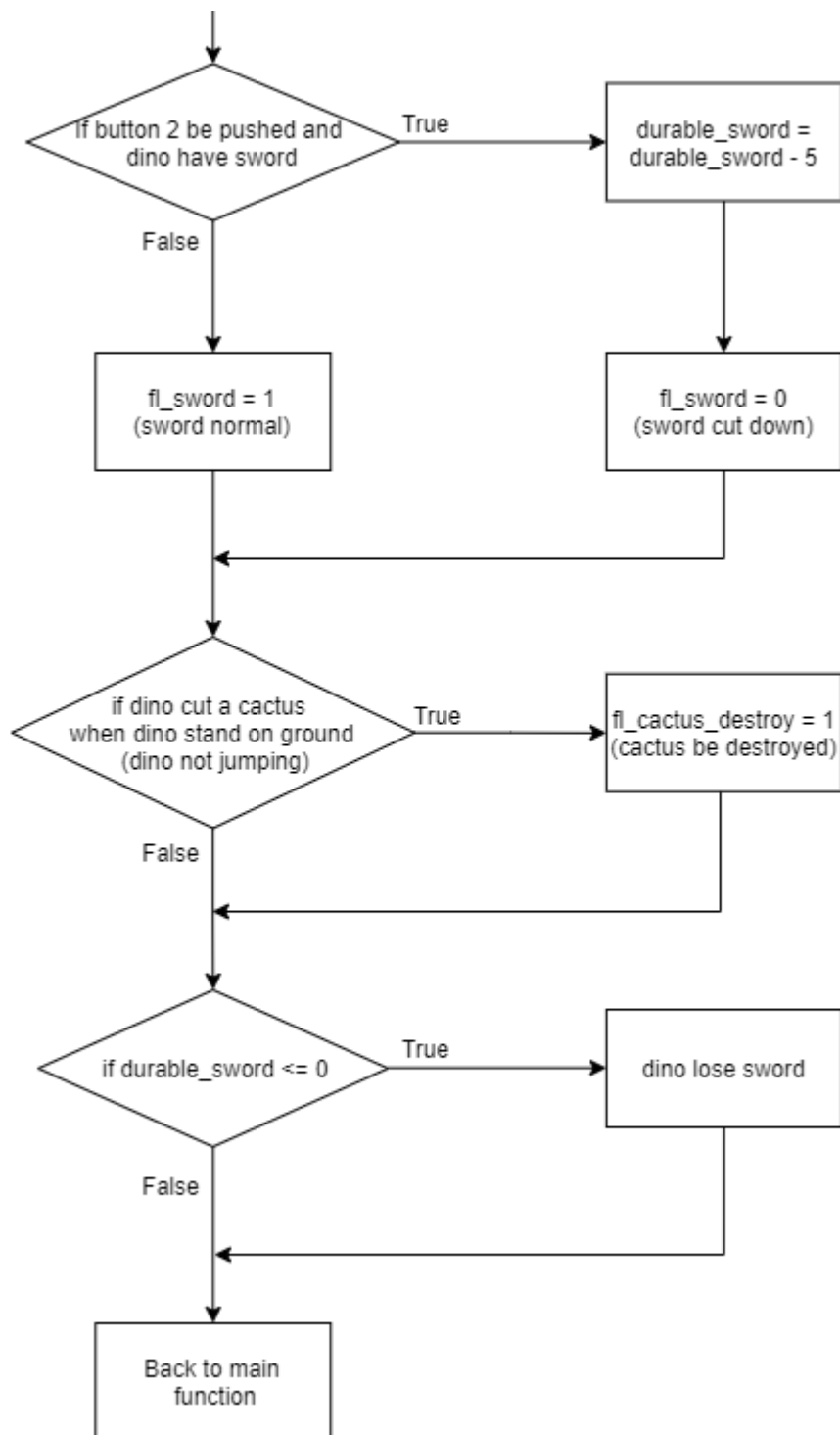


Hình 12 – Sơ đồ khối `bird_work` function

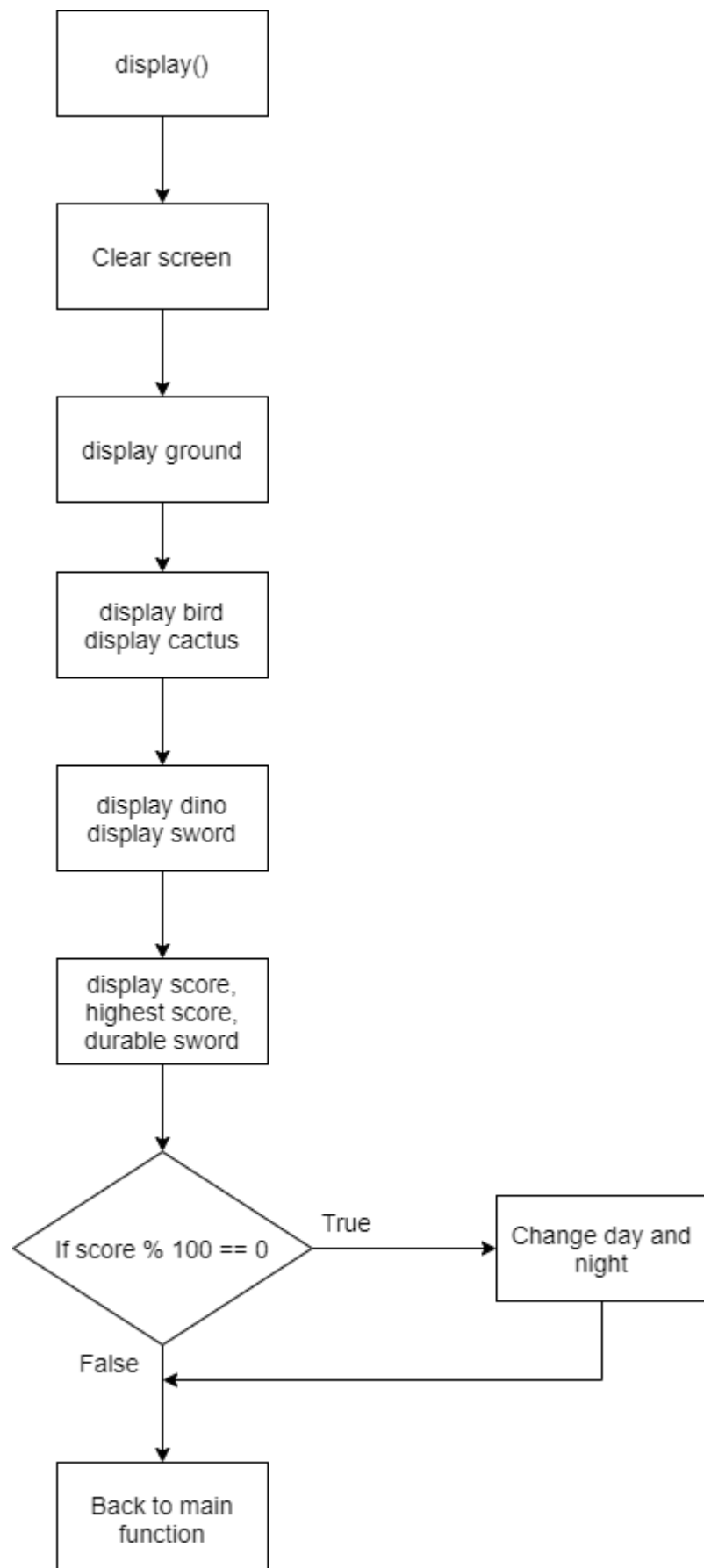


Hình 13 – Sơ đồ khối `bird_wing_work` function

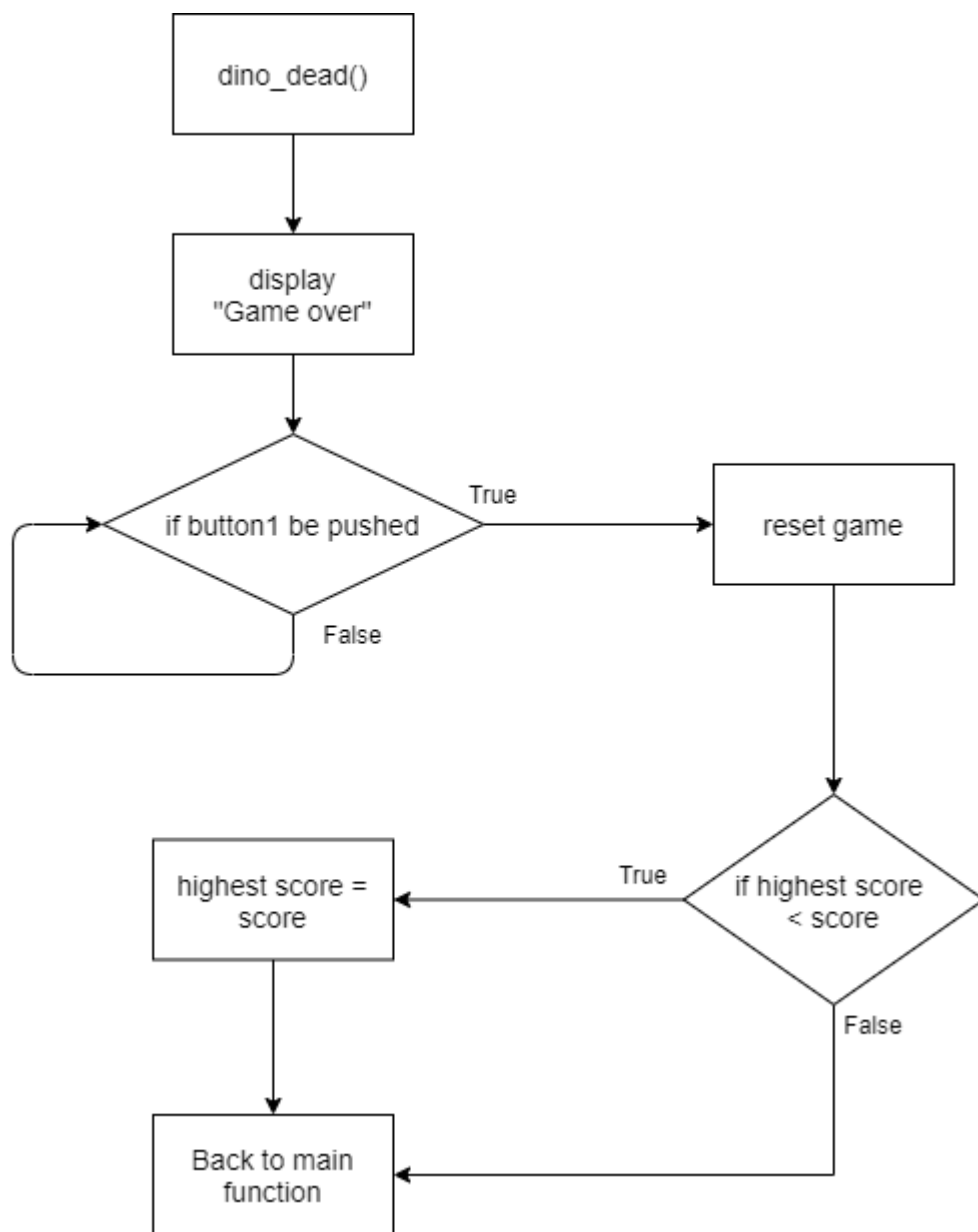




Hình 14 – Sơ đồ khối `sword_work` function

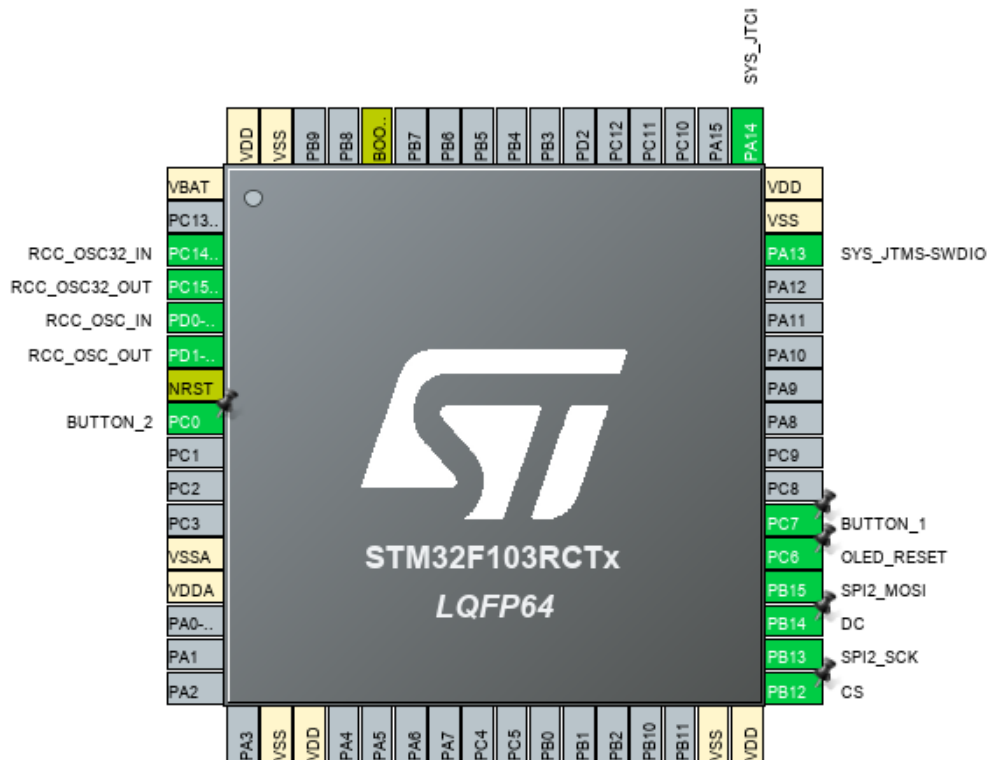


Hình 15 – Sơ đồ khối *display function*



Hình 16 – Sơ đồ khối `dino_dead` function

Cấu hình các chân trên CubeMX:



Hình 17 – Kết nối các chân của STM32

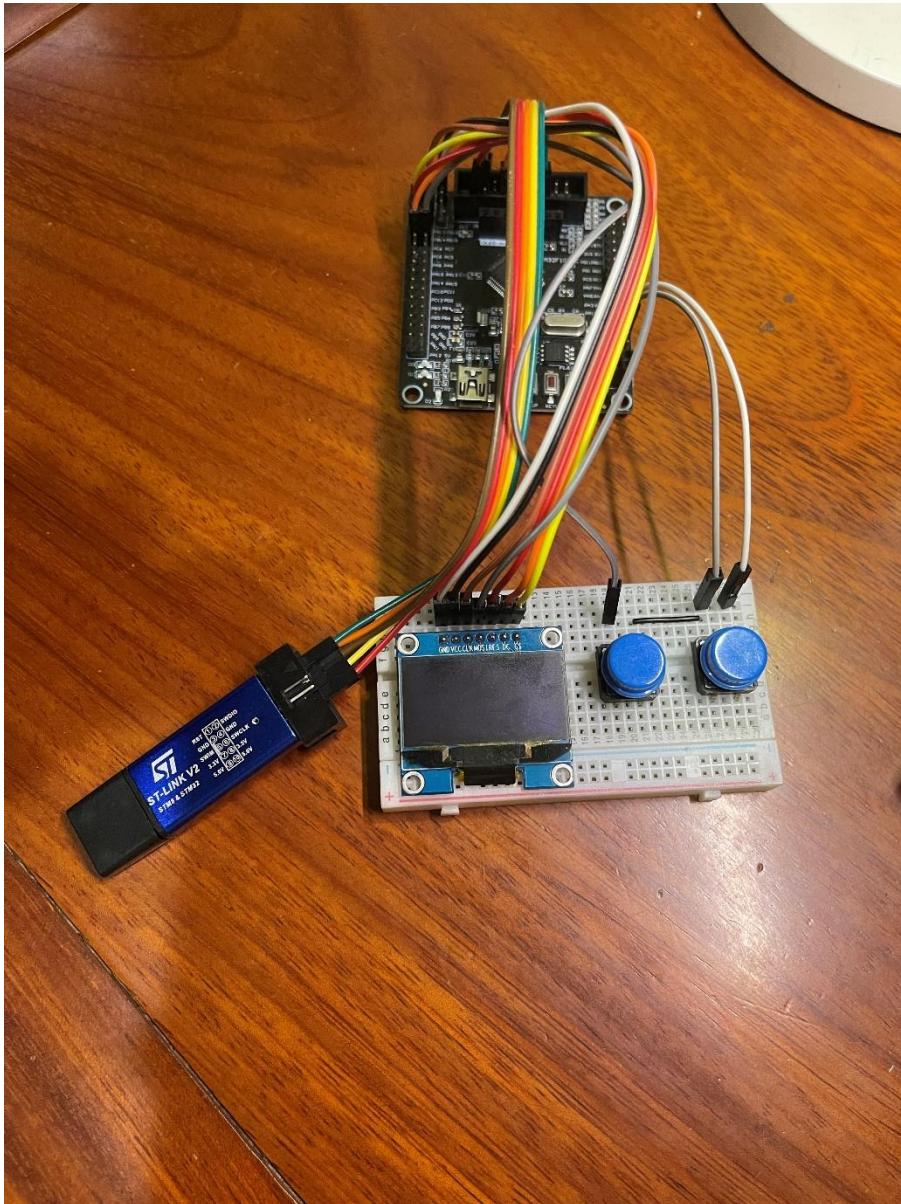
Trong đó:

- PB15: Chân giao tiếp MOSI chuẩn SPI với OLED.
- PB14: Chân giao tiếp DC chuẩn SPI với OLED.
- PB13: Chân giao tiếp SCK chuẩn SPI với OLED.
- PB12: Chân giao tiếp CS chuẩn SPI với OLED.
- PC6: Chân giao tiếp RESET chuẩn SPI với OLED.
- PC7: Chân kết nối với nút nhấn 1.
- PC0: Chân kết nối với nút nhấn 2.

4. HÌNH ẢNH

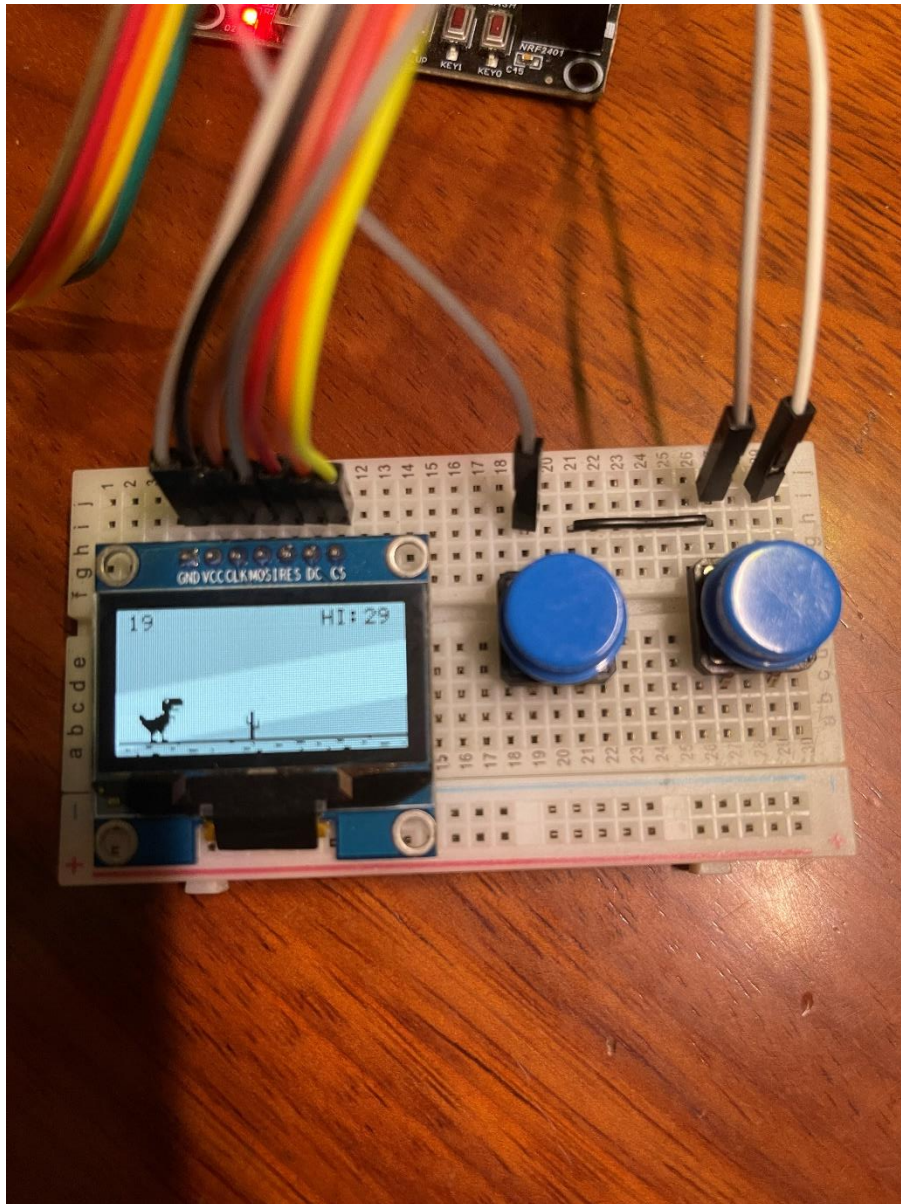
Em đã xác định sản phẩm sẽ là một bộ trò chơi game điều khiển bằng 2 nút bấm đơn giản, lắp trên breadboard. Với các linh kiện có trong hệ thống và quá trình hoạt động đã được mô tả ở những phần trước, phần này sẽ bao gồm nhiều hình ảnh về thiết kế và việc hiển thị các thành phần trong trò chơi.

4.1 Hình ảnh thiết kế

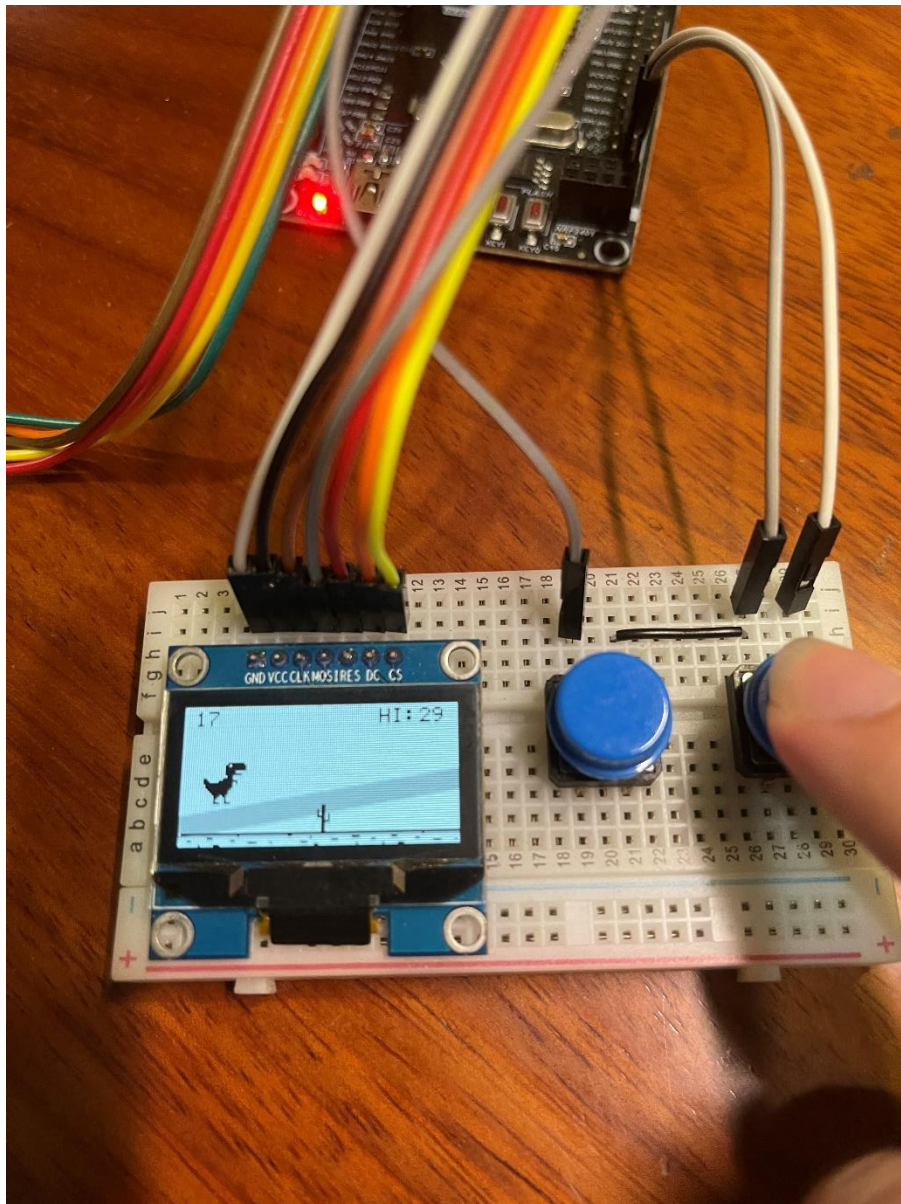


Hình 18 – Hình ảnh thực tế bộ trò chơi

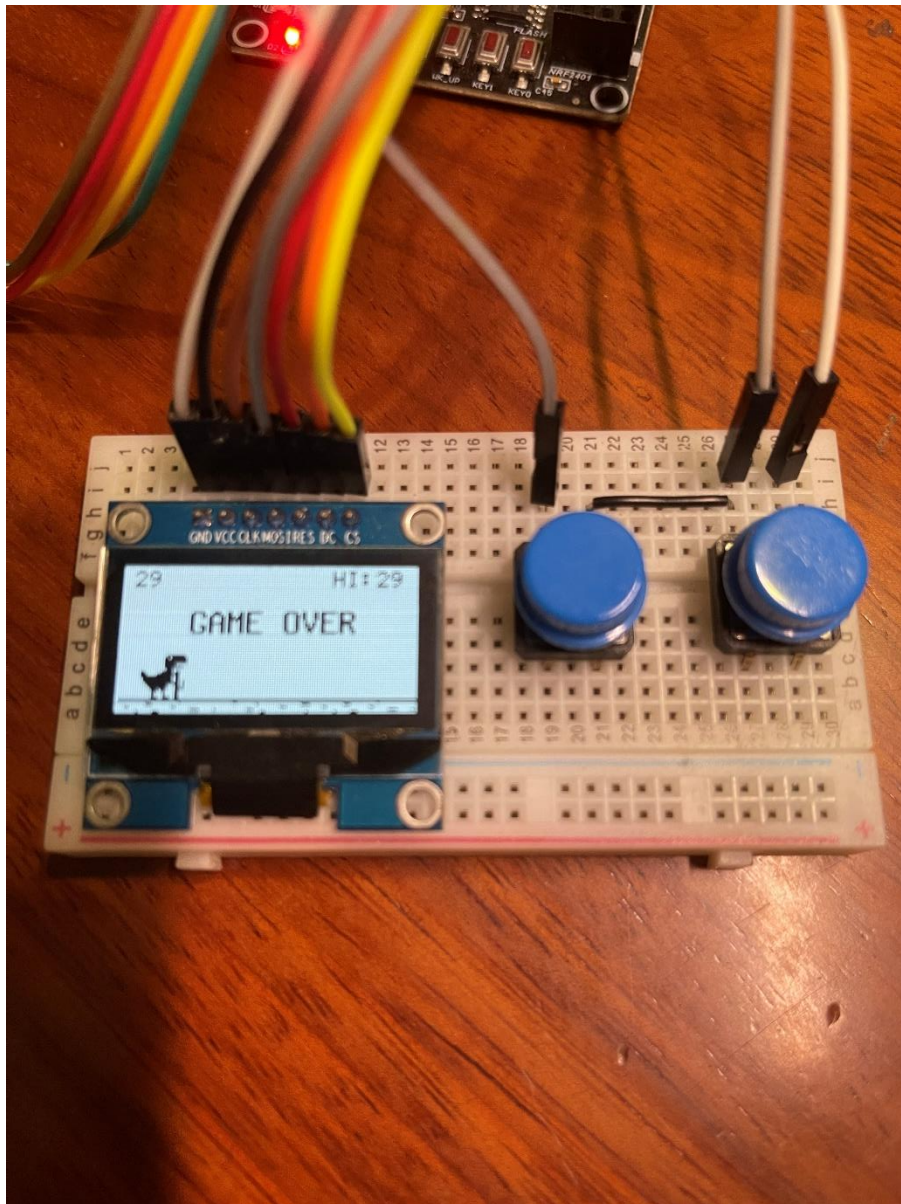
4.2 Hình ảnh trong trò chơi



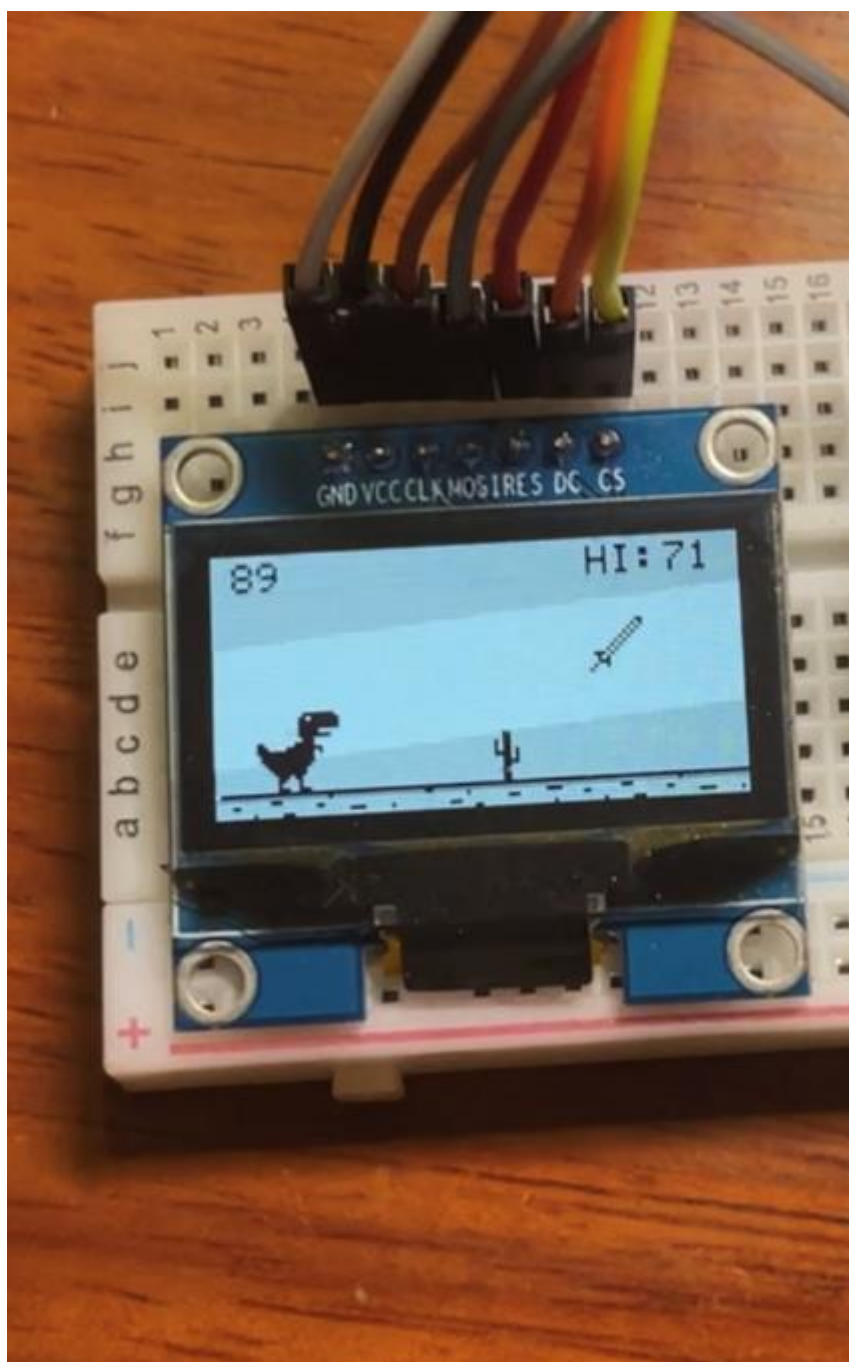
Hình 19 – Hình ảnh giao diện trò chơi



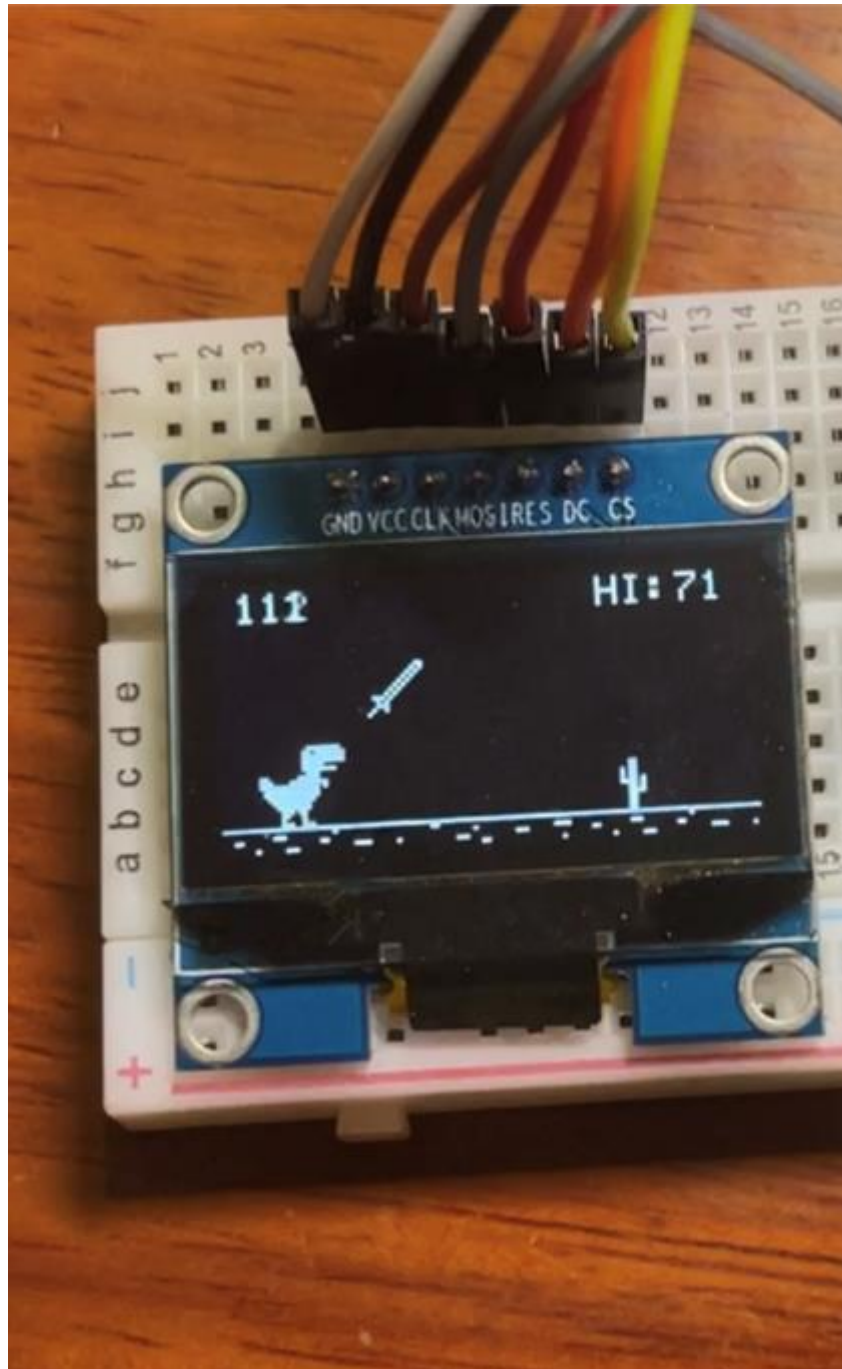
Hình 20 – Hình ảnh dino nhảy lên



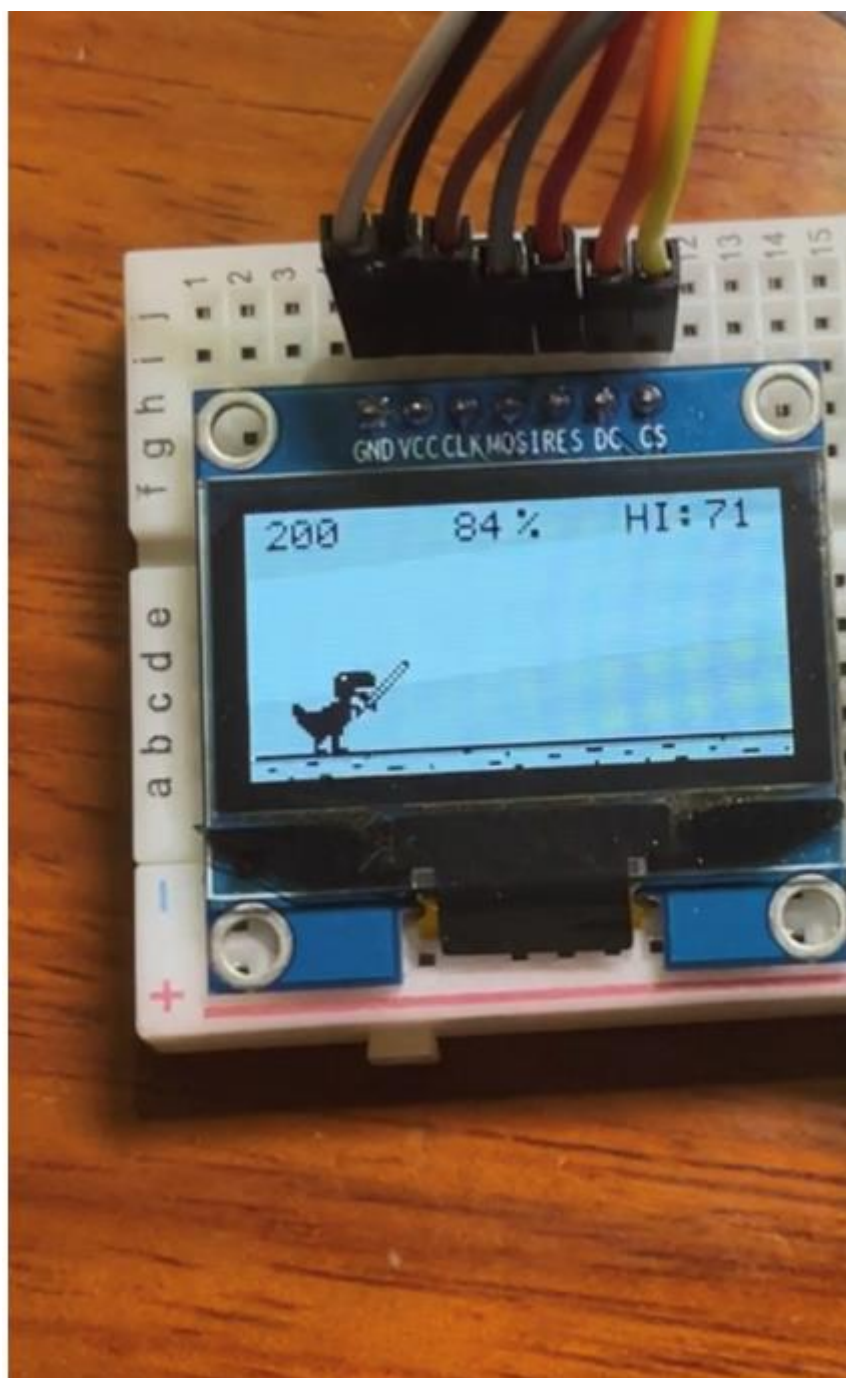
Hình 21 – Hình ảnh dino chết vì va chạm với xương rồng



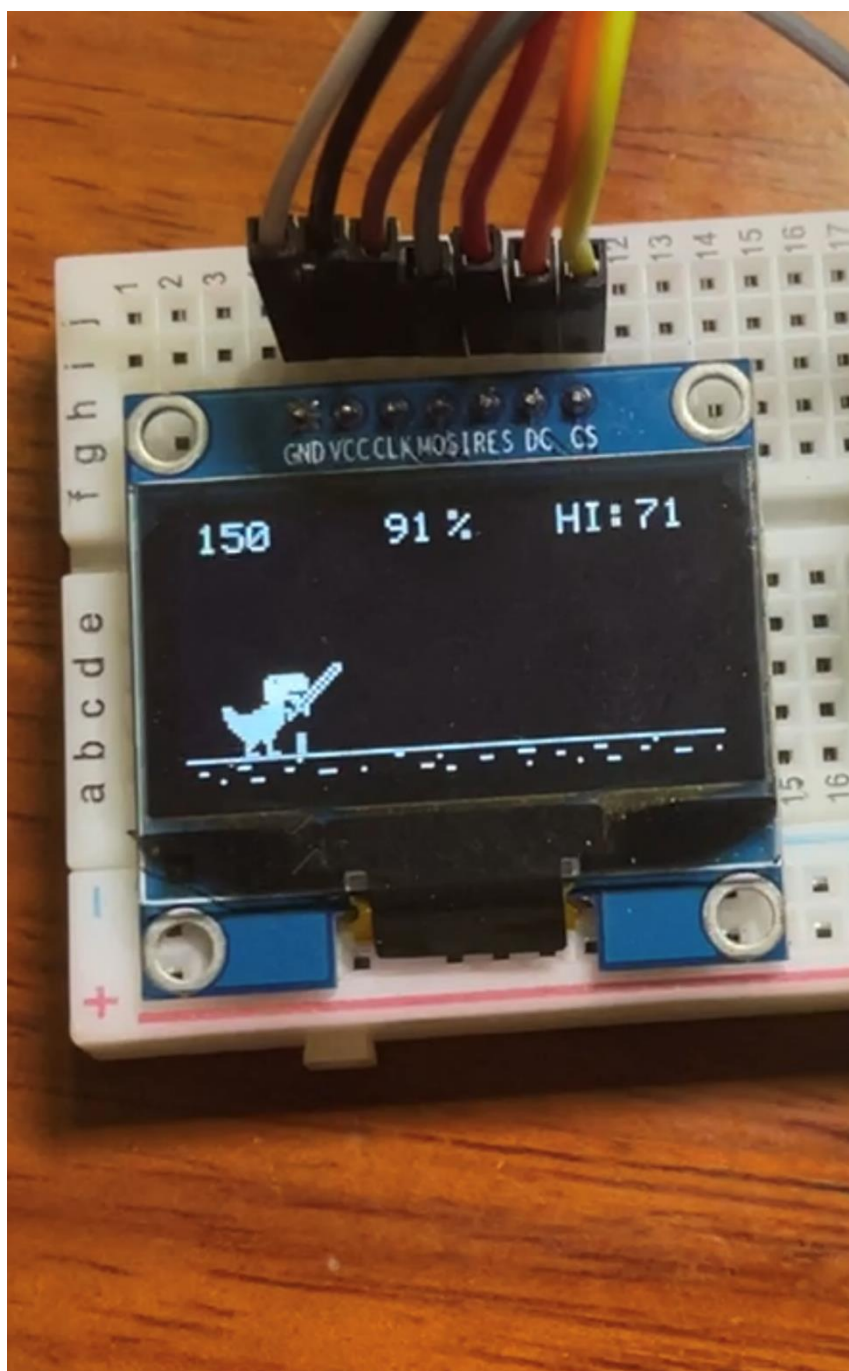
Hình 22 – Hình ảnh kiểm xuất hiện



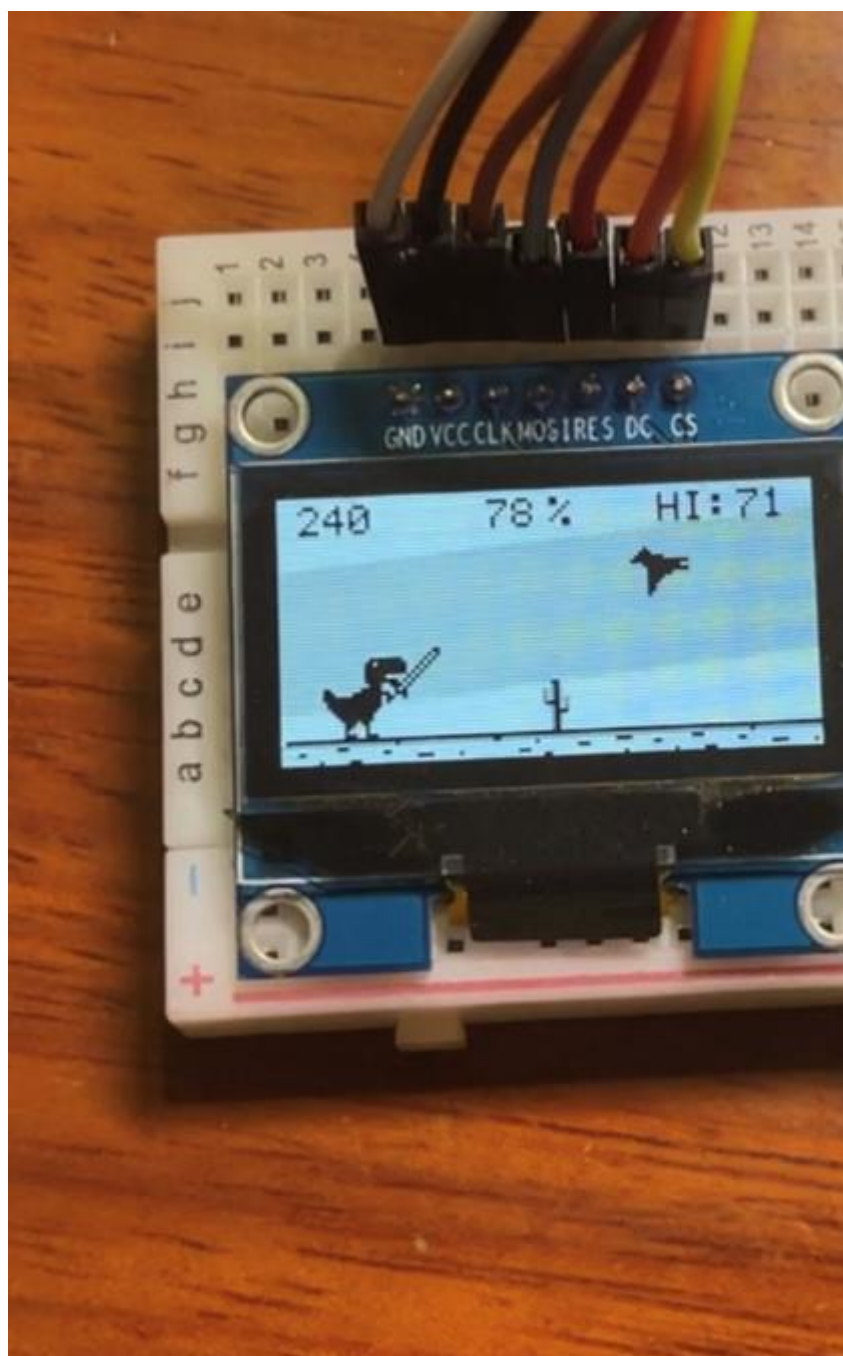
Hình 23 – Hình ảnh kiểm xuất hiện (ban đêm)



Hình 24 – Hình ảnh dino cầm kiếm



Hình 25 – Hình ảnh dino chém cactus



Hình 26 – Hình ảnh chim xuất hiện



Hình 27 – Hình ảnh dino chết vì va chạm với chim

5. THAM KHẢO

- STMicroelectronics, STM32F103 Reference manual RM0008, Rev 20, 2018.

Trích từ: <https://www.st.com/resource/en/datasheet/stm32f103c8.pdf>

- Github, SH1106 driver 1.3 OLED display for STM32 using HAL, 2019.

Trích từ: <https://github.com/desertkun/SH1106>