國產IC開發套件 Filogic 130 (MT7933) 示範案例 雷射槍射擊遊戲



開發板Filogic130 介紹 周邊介面與範例說明



GND

5

1 O T

開發板硬體外觀





2023 © 資訊工業策進會 Institute for Information Industry

擴展連接器

1 O T

101

5

- 為各種傳感器和設備提供類似的引腳 擴展連接器連接性
- 該板有46個與其他接口復用的GPIO根 據用例,用戶可以配置每個I/O功能



GPIO-39 默認為 NC

30

信號名稱	連接器針數	信號名稱	連接器針數
GPIO_0	預留閃光燈	GPIO_27	J87 - 18
GPIO_1	預留閃光燈	GPIO_28	J87 - 20
GPIO_2	預留閃光燈	GPIO_29	J28 - 15
GPIO_3	預留閃光燈	GPIO_30	J28 - 17
GPIO_4	預留閃光燈	GPIO_31	J28 - 19 USB OTG 預留
GPIO_5	預留閃光燈	GPIO_32	J28 - 21 USB OTG 預留
GPIO_6	J28 - 2為 Arduino 預留:SPI0_SCK	GPIO_33	J28 - 23 USB OTG 預留
GPIO_7	J28 - 4為 Arduino 預留:SPI0_CSN	GPIO_34	J28 - 25 USB OTG 預留
GPIO_8	J28 - 6為 Arduino 預留 :SPI0_MISO	GPIO_35	J28 - 18
GPIO_9	J28 - 8為 Arduino 預留:SPI0_MOSI	GPIO_36	J28 - 20
GPIO_10	J28 - 10	GPIO_37	J28 - 22
GPIO_11	J28 - 12	GPIO_38	J28 - 24
GPIO_12	J87 - 5	GPIO_39	J28 - 29 AU_AMP_MUTE 的默認值
GPIO_13	J87 - 7	GPIO_40	J28 - 27
GPIO_14	9 - 78L	GPIO_41	J28-1
GPIO_15	J87 - 11	GPIO_42	J28 - 30為 Arduino 預留:UART1_RX
GPIO_17	J87-13	GPIO_43	J28-3 AcSin 相処
GPIO_18	J87 - 15	GPIO_44	J28 - 28為 Arduino 預留:UART1_TX
GPIO_19	J87 - 17為 Arduino 預留 :12C1_SDA	GPIO_45	J28-5
GPIO_20	J87 - 19為 Arduino 預留 :12C1_SCL	GPIO_46	J28-7
GPIO_21	J87 - 21	GPIO_47	J87-2
GPIO_22	J87 - 23	GPIO_48	J87 - 6 CM33 UART 預留
GPIO_23	J87 - 25	GPIO_49	J87-4
GPIO_24	J87 - 12	GPIO_50	J87 - 8 CM33 UART 預留
GPIO_25	J87 - 14	GPIO_51	J28 - 9
GPIO_26	J87 - 16	GPIO_52	J28 - 11

Filogic SDK 下載

Filogic130 Arduino SDK的各個版本

- 從資策會雲端硬碟下載
- https://drive.google.com/drive/folders/1z7HscksgsalRf HW7T49nVbl60gY-weV7
- ◆ 在SDK v1.0.0裡面, 提供開發板的電路圖

安裝Arduino IDE1.8

5 1018

第一步: 進入Arduino官網 <u>https://www.arduino.cc/en/M</u> <u>ain/Software</u>



ARDUINO 1.8.9

Download the Arduino IDE

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the <u>Getting Started</u> page for Installation instructions.

Windows Installer, for Windows XP and up WINDOWS ZIP file for non admin install

Windows app Requires Win 8.1 or 10

Mac OS X 10.8 Mountain Lion or newer

Linux 32 bits Linux 64 bits Linux ARM 32 bits Linux ARM 64 bits

Release Notes Source Code Checksums (sha512)

Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). Learn more on how your contribution will be used.

\$10

\$25

SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED 32,214,054 TIMES. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!

\$5

\$3

\$50 OTHER



第二步: 下載Arduino IDE1.8.X版 本

安裝Arduino IDE1.8

第三步:安裝ArduinoIDE

		K A	
🧿 Arduino Setup: License Agreement 🦳 🗌 🗙	💿 Arduino Setup: Installation	Options – 🗆 X	💿 Arduino Setup: Installation Folder 🦳 🗌 🗙
Please review the license agreement before installing Arduino. If you accept all terms of the agreement, click I Agree.	Check the components y you don't want to install.	ou want to install and uncheck the components Click Next to continue.	Setup will install Arduino in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.
SNU LESSER GENERAL PUBLIC LICENSE			
Version 3, 29 June 2007	Select components to install:	✓ Install Arduino software ✓ Install USB driver	Destination Folder
Copyright (C) 2007 Free Software Foundation, Inc. < <u>http://fsf.org/</u> >		Create Start Menu shortcut	
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.		Create Desktop shortcut Associate .ino files	
This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.	Space required: 482.4MB		Space required: 482.4MB Space available: 102.5GB
Cancel Nullsoft Install System v3.0 I Agree	Cancel Nullsoft Insta	Il System v3.0 <back next=""></back>	Cancel Nullsoft Install System v3.0 < Back Install

Filogic 130 環境設定

第四步:Arduino IDE1.8安裝 成功後,啟動進入Arduino IDE1.8環境,點擊檔案→偏好 設定

sketch_jun07a Arduino 1.8.13 — 🗆 🗙					
案 編輯 草稿	碼 工具 說明				
新增 開啟 問助完近	Ctrl+N Ctrl+O	<u>م</u>			
用啟取紅 草稿碼簿 範例 關閉 儲存 另存新檔	> Ctrl+W Ctrl+S Ctrl+Shift+S	{ setup code here, to run once			
頁面設定 列印	Ctrl+Shift+P Ctrl+P				
偏好設定	Ctrl+Comma	main code here, to run repea			
離開	Ctrl+Q				
9 }	35				

Arduino Yún 🕸 COM8

Filogic 130 環境設定

第五步:在開發板管理網址設定: <u>https://www.ideas-hatch.com/filogic_130_arduino_sdk/</u> <u>package_mtk_filogic_130_index.json</u>

 「新聞」		×	
設定 網路			
草稿碼簿的位置:		1	
C:\Users\user\Documents\Arduino	SLAD	證證	
編輯器語言: System Default	(需要重新啟動 Arduino)		
編輯器字型大小: 22		◎ 額外的開發板管理員網址	×
介面縮放率: 🔽 自動 100 🛊 % (需要重新啟動 Arduino)		輸入額外的網址,一行一個	
Theme: Default theme 🗸 (需要重新散動 Arduino)		h.com/filogic_130_arduino_sdk/package_mt	k_filogic_130_index.json ^
顯示詳細輸出: 🗌 編譯 🗌 上傳			3
編譯器警告: 無 🗸			~
□ 顯示行數	🗌 啟用程式碼摺疊功能		>
☑ 上傳後驗證程式碼	🗌 使用外部編輯器	點擊取得非自万做于的支援網址清車	
☑ 散動時檢查有無更新	☑ 驗證或上傳時先存檔		
Use accessibility features			
額外的開發板管理員網址: https://www.ideas-hatch.com/filogic_130_arduin	o_sdk/package_mtk_filogic_130_	index.json	
在偏好設定檔裡還有更多設定值可直接編輯			
C:\Users\user\AppUata\Local\Arduino15\preferences.bt (口能在Arduino主執行之時進行編輯)			
()//)四1工110(0000///約111/2/#72年118冊4号/			
		確定取消	

Filogic 130 環境設定

第六步:安裝開發板Filogic 1.3.0

				◎ 開發板管理員	
💿 Arduino Ardu	uino 1.8.18			類型 全部 → Filogic	
檔案 編輯 草稿碼	工具 說明				
00 B E	自動格式化 封存草稿碼	Ctrl+T		Filogic 130 by MediaTek Labs	
Arduino	修正編碼並重新載入		2次4	此世代也言的两被权: Filogic 130.	
	管理程式庫	Ctrl+Shift+I	亡真不	Online Help More Info	
}	序列埠監控視窗	Ctrl+Shift+M	.nes		100
void loo	序列繪圖家	Ctrl+Shift+L			
{	WiFi101 / WiFiNINA Firmware Updat	er		WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW	1.2.0
{	開發板: "Arduino Uno"	*	開發板管理員		1.0.0
{	序列埠	2	Arduino Yún		
button	取得開發板資訊		Arduino Uno		
if (but	燒錄器: "AVRISP mkll"	2	Arduino Duemilanove or Diecimila		
ii (buc	燒錄Bootloader		Arduino Nano		
{			Arduino Mega or Mega 2560		
digit	talWrite(latchPin,	LOW);	Arduino Mega ADK (
shift	tOut(dataPin, cloc	kPin, MS	Arduino Leonardo		
11			Arduino Leonardo ETH		同時
digit	taiwrite(latchPin,	HIGH);	Arduino Micro (

Filogic 130 環境設分

第七步:選擇Filogic開發板及 Serial com port

Arduino Ard	uino	1.8.18					
當案 編輯 草稿碼	工具	說明					
		自動格式化	Ctrl+T				
		封存草稿碼					
Arduino		修正編碼並重新載入					
}		管理程式庫	Ctrl+Shift+I				
, 		序列埠監控視窗	Ctrl+Shift+M				
vola 100		序列繪圖家	Ctrl+Shift+L				
{		WiFi101 / WiFiNINA Firmware Updater					
{							
, r		開發板: "Arduino Uno"	;	-	開發板管理員	_	
1		序列埠	3	2	Arduino AVR Boards		
button		取得開發板資訊			MediaTek Filogic Series 🔅		Filogic 130
if(but		燒錄器: "AVRISP mkll"	:	>			
۲. ۲		燒錄Bootloader					

digitalWrite(latchPin, LOW); // 送資料前要先把 latchPin shiftOut(dataPin, clockPin, MSBEIRST, 9); //送出資料,17

Arduino Arduino 1.8.18								
檔案 編輯 草稿碼 工具	. 說明							
🐼 🕢 🖪 🖻	自動格式化	Ctrl+T						
	封存草稿碼							
Arduino	修正編碼並重新載入							
1	管理程式庫	Ctrl+Shift+I						
, , , ,	序列埠監控視窗	Ctrl+Shift+M						
void loo	序列繪圖家	Ctrl+Shift+L						
{	WiFi101 / WiFiNINA Firmware Updater							
{	開發板: "Arduino Uno"	>						
{	序列埠: "COM5 (Arduino Uno)"	3		序列埠				
button	取得開發板資訊		~	COM5 (Arduino Uno)				
if(but	燒錄器: "AVRISP mkll"	>						
{	燒錄Bootloader	1						

實作:雷射槍射擊遊戲展示







電源啟動後,優先 掃描周圍物體和砲 台之間的距離 等待藍牙開啟並 進行操控 操控方式有實體 及手機APP兩種 方式操作







雷射模組 (Laser Rangefinder Module V2)

雷射槍射擊砲台需要上述元件進行開發

CC ID:2ADWC-A/7833CLD A/7833CLD, 722044, 2234

MT7933 HDK Base for SAC_V21 MTK358



按鈕模組



有源蜂鳴器



搖桿模組



1 O T

1018

5





1 O T

2018

5

S





1 O T

1018

5

S

APP介紹

- 玩家使用APP控制砲台時,需要先掃 描周邊藍牙裝置,找到雷射砲台並進 行連線
- 連線成功後,砲台蜂鳴器會發出提示
 音(連線成功),玩家可以透過下方方
 向鍵控制砲台轉動以及攻擊行為
- 攻擊到物體時,蜂鳴器會快頻發出提 示音,並將分數顯示計分顯示區。





#include <Unistep2.h> #include <LBLE.h> #include <LBLEPeriphral.h> //藍牙宣告 //定義一個只有 1 個特徵的簡單 GATT 服務 LBLEService ledService("110C0010-E8F2-537E-4F6C-D104768A1214"); LBLECharacteristicString switchCharacteristic("110C0011-E8F2-537E-4F6C-D104768A1214", LBLE READ | LBLE WRITE); #define BUF SIZE 30 bool bConnStatus = false; uint8 t rData[BUF SIZE]; 程式宣告區 String BTData; //蜂鳴器 1. 匯入藍牙和步進馬達的函 #define buzzPin 13 String LRF feedback ; // Collect the RAW LRF Feedback String measurement ; // Collect the Digital LRF Measurement 示庫 char data; int Bytes_counter = 12; bool ws = true; 2. 宣告各個感測器腳位 bool st = true: bool co = true; bool ch = true: #define LRF Serial2 // LRF module connected on Serial1 of ARDUINO Micro float record[27]; float distance; // 搖桿按鍵輸出 SW 接 Arduino pin 7 float pastdistance; int buttonPin = 47; bool CMM EN = true; // 定義 Arduino 從搖桿按鍵 SW 讀入值為 buttonPress int buttonPress = 0; bool adc = true; int Time, pastTime; //步進馬達 int point = 0; int Xi = 0;String strpoint; int Yi = 0;double dudu = 0.087890625; int xrun, yrun, Xdu, Ydu, X, Y; int Counter = 0; Unistep2 stepperX(23, 22, 21, 20, 4096, 900);// IN1, IN2, IN3, IN4, 總step數, 每步的延遲(in micros) Unistep2 stepperY(19, 18, 17, 15, 4096, 900);// IN1, IN2, IN3, IN4, 總step數, 每步的延遲(in micros) bool che = true; //搖桿 #define xposPin A0 // 雙軸按鍵搖桿 VRx 接 Arduino Analog pin A0 // 雙軸按鍵搖桿 VRy 接 Arduino Analog pin A1 #define yposPin A1 int Xpos = 0; // 定義x軸伺服器位址參數 // 定義丫軸伺服器位址參數 int Ypos = 0;

程式介紹

//雷射	す 模組						
byte	cmd_1[] =	{ 0x80,	0 x 06,	0x03,	0x77 }	;	// 連續測量模式
byte	cmd_2[] =	{ 0x80,	0x06,	0 x 02,	0x78 }	;	// 單次測量模式
byte	cmd_3[] =	{ 0x80,	0 x 06,	0x05,	0x01,	0x74 };	// 雷射點打開
byte	cmd_4[] =	{ 0x80,	0x06,	0 x 05,	0x00,	0x75 };	// 雷射點關閉
byte	cmd_5[] =	{ OxFA,	0x04,	0x09,	0x05,	0xF4 };	// 5公尺
byte	cmd_6[] =	{ 0xFA,	0 x 04,	0x09,	0x0A,	0xEF };	// 10公尺
byte	cmd_7[] =	{ 0xFA,	0x04,	0x09,	0x1E,	0xDB };	// 30公尺
byte	cmd_8[] =	{ 0xFA,	0x04,	0 x 09,	0x32,	0xC7 };	// 50公尺
byte	cmd_9[] =	{ 0xFA,	0x04,	0x09,	0x50,	0xA9 };	// 80公尺
byte	cmd 10[] =	= { 0xFA,	, 0x04,	0x0C,	0x02,	0xF4 };	// 0.1mm 精度
byte		= { 0xFA,	, 0x04,	0x0C,	0x01,	0xF5 };	// 1mm 精度
	-						
byte	cmd 12[] =	= { 0xFA,	, 0x04,	0x0A,	0x00,	0xF8 };	// 2Hz
byte		= { 0xFA,	0x04,	0x0A,	0x05,	0xF3 };	// 5Hz
byte	cmd 14[] =	= { 0xFA,	, 0x04,	0x0A,	0x0A,	0xEE };	// 10Hz
byte	cmd 15[] =	= { 0xFA,	, 0x04,	0x0A,	0x14,	0xE4 };	// 20Hz
_	_						

- 宣告雷射模組控制指令
- 指令1~4:控制雷射測量
 模式與開關
- 指令5~9:設定雷射模組
 量測最大距離
- 3. 指令10、11:設定雷射模 組精度
- 4. 指令12~15:設定雷射模 組量測頻率



```
void setup() {
 Serial.begin(9600);
 delay(100);
 LRF.begin(9600);
 delay(500);
                                           //開啟雷射點
 LRF.write(cmd_3, sizeof(cmd_3));
 delay(500);
 LRF.write(cmd 7, sizeof(cmd 7));
                                          //設定量測距離80公尺
 Serial.println("LRF Range Set to 30m");
 delay(500);
                                          //設定量測精度0.1mm
 LRF.write(cmd_10, sizeof(cmd_10));
 Serial.println("LRF Resolution Set to O.lmm");
 delay(500);
                                           //Arduino 從搖桿按鍵讀入電壓
 pinMode(buttonPin, INPUT);
 pinMode(xposPin, INPUT);
 pinMode(yposPin, INPUT);
                                          //設定當按鍵沒按時,輸出為高電壓
 digitalWrite(buttonPin, HIGH);
 //步進馬達
 pinMode(23, OUTPUT);
 pinMode(22, OUTPUT);
 pinMode(21, OUTPUT);
 pinMode(20, OUTPUT);
 pinMode(19, OUTPUT);
 pinMode(18, OUTPUT);
 pinMode(17, OUTPUT);
 pinMode(15, OUTPUT);
 //蜂鳴器
 pinMode(buzzPin, OUTPUT);
 Measure();
 Measure();
 Measure();
```

```
程式初始化
```

- 1. 設定雷射模組
- 2. 設定步進馬達接腳輸出



while (Counter < 27) { stepperX.run(); Scan(); //回歸130度 while (ws == true) { if (Counter > 28) { ws = false;

stepperX.run(); reset();

//設定蓮續測量

LRF.write(cmd 15, sizeof(cmd 15)); // Set LRF Module Frequency to 20Hz delay(100); LRF.write(cmd_9, sizeof(cmd_9)); //設定量測距離80公尺 delay(100); LRF.write(cmd 10, sizeof(cmd 10)); //設定量測精度0.1mm delay(100); //初始化 BLE 子系統 LBLE.begin(); while (!LBLE.ready()) { delay(100); Serial.println("BLE ready"); Serial.print("Device Address = ["); Serial.print(LBLE.getDeviceAddress()); Serial.println("]"); // 配置我們的廣告數據。 // 開始廣告 // 在這種情況下,我們只需創建一個代表一個廣告的廣告 // 具有設備名稱的可連接設備 LBLEAdvertisementData advertisement; advertisement.configAsConnectableDevice("OvOcc"); // 配置我們設備的通用訪問配置文件的設備名稱 // 通常這與廣告數據中的名稱相同。 LBLEPeripheral.setName("OvOcc");

程式初始化 3. 掃描周圍距離值 4. 開啟藍牙

// 添加特性到 ledService ledService.addAttribute(switchCharacteristic);

// 向 GATT 服務器添加服務(外設) LBLEPeripheral.addService(ledService);

// 啟動 GATT 服務器 - 現在 // 可連接 LBLEPeripheral.begin();

LBLEPeripheral.advertise(advertisement); Serial.print("conected="); Serial.println(bConnStatus); Unistep2 stepperX(23, 22, 21, 20, 4096, 10);// IN1, IN2, IN3, IN4, 總step數, 每步的延遲(in micros) Unistep2 stepperY(19, 18, 17, 15, 4096, 10);// IN1, IN2, IN3, IN4, 總step數, 每步的延遲(in micros) digitalWrite(buzzPin, HIGH);//有源蜂鳴器響起 delay(200); digitalWrite(buzzPin, LOW);//有源蜂鳴器響起

程式介紹

void loop() {
 //確認藍牙是否已經連接
 if (bConnStatus != LBLEPeripheral.connected()) {
 bConnStatus = LBLEPeripheral.connected();
 Serial.print("conected=");
 Serial.println(bConnStatus);
 digitalWrite(buzzPin, HIGH);//有源蜂鳴器響起
 delay(200);
 digitalWrite(buzzPin, LOW);//有源蜂鳴器響起

}

//App傅送資料
if (switchCharacteristic.isWritten()) {
 const String value = switchCharacteristic.getValue();
 Serial.print("收到了: "); Serial.println(value);
 BTData = value;

}

stepperX.run(); stepperY.run(); Xpos = analogRead(xposPin); Ypos = analogRead(yposPin); buttonPress = digitalRead(buttonPin); //讀入搖桿按鍵電位 Movehorizontally(); //Shooting(); Continuousmeasurement(); /*Serial.print("Xpos = "); Serial.println(Xpos); Serial.print("Ypos = "); Serial.println(Ypos);*/

```
LRF_feedback = "";
Bytes_counter = 0;
```

主程式區

- 1. 判斷藍牙是否連接至APP
- 2. 接收APP的資料
- 3. 讀取搖桿X、Y軸數值,控制
 步進馬達轉動
- 4. 讀取按鍵電位·控制雷射模 組是否做動

