



追 求 卓 越 创 造 精 品

TO BE BEST

TO DO GREAT

广州友善之臂计算机科技有限公司

Matrix API Reference Manual

Copyright © 2007-2016 FriendlyARM

All rights reserved.



追 求 卓 越 创 造 精 品

TO BE BEST

TO DO GREAT

广州友善之臂计算机科技有限公司

Brief introduction

This manual is a friendly arm of the Guangzhou Computer Technology Co., Ltd. (referred to as "friendly arm") to create and maintain a friendly arm currently does not provide any services to answer and explain the contents of this manual, user feedback in the forums you can encounter the problems and questions, we will amend or your suggestion in a future update, this manual's home date version flag.

Matrix API reference manual library is written and maintained a library written in C language with a friendly arm, mainly to use Matrix accessory module, integrated library is very rich, in addition to the usual GPIO library, the library also includes I2C, SPI library , UART software library and PWM libraries. Since the launch of the module are friendly and Matrix command, so this library is called the Matrix API reference manual library.

Matrix API reference manual library contains a GPIO command line tool, which can be used to set Ou GPIO pins can be used to read the GPIO pin can be used even for the purpose of controlling the GPIO pin Shell script.

We welcome all friends to copy the spread of this manual, but are not allowed to extract some or all of the contents for commercial purposes, offenders accountable, friendly arm retains the right to interpretation and amendment of this manual.

This manual is published by the Guangzhou friendly arm Computer Technology Co., Ltd., please indicate the source, it is inevitable omissions and deficiencies in the manual, we welcome the valuable advice.



追求卓越 创造精品

TO BE BEST

TO DO GREAT

广州友善之臂计算机科技有限公司

Interface Description File Operations

Interface name	Parameters and return value	Function Description
int openHW (String devName, int flags)	Parameters: devName: flags to write data: Open a file, for example, read-write or read-only open Return Value: Successful returns a file descriptor, error return -1.	Open the device.
int ioctlWithIntValue (int fd, int cmd, int value)	Parameters: fd: file descriptor device cmd: ioctl command value: command parameters limit integer Return Value: returns success 0, -1 error.	Execution equipment ioctl operating
int writeHW (int fd, byte [] data)	Parameter Description: FD: data to be written: the data to be written to file descriptor data Return Value: It returns the number of bytes successfully written, an error return -1.	Write data to the device or file opens.
int readHW (int fd, byte [] buf, int len)	Parameter Description: fd: file descriptor data to be read buf: Storing data buffer len: the number of bytes to be read Return Value: Returns the number of bytes read successfully, the error return -1 if the end of file transfer read before, if the read returns 0.	Reading data from the opened file or a device.
int selectHW (int fd, int sec, int usec)	Parameter Description: fd: file descriptor to be queried sen: How long data block waiting (Unit: second) usec: multiple data blocks waiting time (unit: ns, 1000 ns = 1 millisecond)	Query whether the open device or file data is readable.



追求卓越 创造精品

TO BE BEST

TO DO GREAT

广州友善之臂计算机科技有限公司

	<p>Return Value:</p> <p>If fd-readable data, returns 1 if no data is read, returns 0, -1 error.</p>	
void closeHW (int fd)	<p>Parameter Description:</p> <p>fd: file descriptor to be closed</p> <p>Return Value: None</p>	<p>Closes the specified file descriptor</p>

Read ADC conversion results Interface Description

Interface name	Parameters and return value	Function Description
int setI2CSlave (int fd, int slave)	<p>Parameter Description:</p> <p>fd: file descriptor slave I2C devices: I2C device address, such as an EEPROM devices are generally 0x50</p> <p>Return Value: 0 successful return, else return -1</p>	<p>I2C device address is provided to be operated, for example, EEPROM devices are generally 0x50</p>
int setI2CTimeout (int fd, int timeout)	<p>Parameter Description:</p> <p>fd: file descriptor I2C devices</p> <p>timeout: overtime time</p> <p>Return Value: 0 successful return, else return -1</p>	<p>Set the timeout (ioctl I2C_TIMEOUT)</p>
int setI2CRetries (int fd, int retries)	<p>Parameter Description:</p> <p>fd: file descriptor I2C devices</p> <p>retries: number of retries</p> <p>Return Value: 0 successful return, else return -1</p>	<p>Setting the number of retries (ioctl I2C_RETRIES)</p>
int I2CWriteByteTo (int fd, int pos, unsigned char byteData, int wait_ms)	<p>Parameter Description:</p> <p>fd: file descriptor I2C devices</p> <p>pos: Byte positions byteData: wait_ms the data to be written: waiting for a specified time (ms)</p> <p>Return Value: 0 successful return, else return -1</p>	<p>To write a byte of data to the specified location I2C device, and waits for a specified time (ms)</p>



追求卓越 创造精品

TO BE BEST

TO DO GREAT

广州友善之臂计算机科技有限公司

int I2CReadByteFrom (int fd, int pos, int wait_ms);	Parameter Description: fd: file descriptor I2C devices pos: Byte position wait_ms: waiting for a specified time (ms) Return Value: 0 successful return, else return -1	Reading a byte of data from the specified location I2C device, and waits for a specified time (ms)

SPI Interface Description

Interface name	Parameters and return value	Function Description
int setSPIWriteBitsPerWord (int spi_fd, int bits)	Parameter Description: spi_fd: file descriptor bits SPI device: word length, in bits Return Value: Returns 0 on success, failure returns negative	SPI devices provided each read word length, in bits. Although most of the SPI interface word length is 8 or 16, there will still be some specific examples. It should be noted that, if this member is zero, then, as the word is used by default 8 long (ioctl SPI_IOC_WR_BITS_PER_WORD)
int setSPIReadBitsPerWord (int spi_fd, int bits)	Parameter Description: spi_fd: file descriptor bits SPI device: word length, in bits Return Value: Returns 0 on success, failure returns negative	SPI devices provided each write word length, in bits (ioctl SPI_IOC_RD_BITS_PER_WORD)
int setSPIBitOrder (int spi_fd, int order)	Parameter Description: spi_fd: file descriptor order SPI devices: transmission SPIEnum.MSBFIRST or SPIEnum.LSBFIRST Return Value: Returns 0 on success, failure returns negative	When the transmission is the first transmission device SPI bit low or high bit, and optional parameters SPIEnum.MSBFIRST SPIEnum.LSBFIRST
int setSPIClockDivider (int spi_fd, int divider)	Parameter Description: spi_fd: SPI device file descriptor divider: Division factor, defined SPIEnum.java passed to the beginning SPI_CLOCK_ constants, for example: SPIEnum.SPI_CLOCK_DIV128	Setting the frequency division factor SPI





追求卓越 创造精品

TO BE BEST

TO DO GREAT

广州友善之臂计算机科技有限公司

readBuff, spi_delay, spi_speed, spi_bits)	int int int	spi_delay: Delay spi_speed: transfer speed spi_bits: Word length, in bits Return Value: Returns 0 on success, failure returns negative	
--	-------------------	--	--

GPIO Interface Description

Interface name	Parameters and return value	Function Description
int initPinGPIO (int board)	Parameters: board: Development Board Selection Return Value: Returns 0 on success, failure returns negative	Boards used for identifying the model
int pintoGPIO (int pin)	Parameter Description: pin: pass GPIO pin numbers Return Value: Returns 0 on success, failure returns negative	Transmitting GPIO for calculating a development board corresponding to the index number of GPIO
int exportGPIOPin (int pin)	Parameters: pin: GPIO pin number Return Value: Returns 0 on success, failure returns negative	Notification system need to export control GPIO pin number, equivalent to execute the command echo pin> / sys / class / gpio / export
int unexportGPIOPin (int pin)	Parameters: pin: GPIO pin number Return Value: Returns 0 on success, failure returns negative	Notification system to cancel export a GPIO pins, equivalent to execute the command echo pin> / sys / class / gpio / unexport
int setGPIOValue (int pin, int value)	Parameters: pin: GPIO pin number value: Incoming GPIOEnum.LOW indicates output low, high pass output indicates GPIOEnum.HIGH Return Value: Returns 0 on success, failure returns negative	High or low output on a pin



追求卓越 创造精品

TO BE BEST

TO DO GREAT

广州友善之臂计算机科技有限公司

int getGPIOValue (int pin)	<p>Parameters: pin: GPIO pin number</p> <p>Return Value: Outputs a low level indicates a successful return GPIOEnum.LOW returns GPIOEnum.HIGH represents output high, failed to return a negative</p>	Query a pin state (high or low)
int setGPIODirection (int pin, int direction)	<p>Parameters: pin: GPIO pin number direction: Incoming GPIOEnum.IN represents the input, GPIOEnum.OUT represents the output</p> <p>Return Value: Returns 0 on success, failure returns negative</p>	Configuration pin function output or input
int getGPIODirection (int pin)	<p>Parameters: pin: GPIO pin number</p> <p>Successful return GPIOEnum.IN represents the input, output represents a return GPIOEnum.OUT failure returns negative</p>	Query pin function (output or input)

ADXL345 chip interface description

Interface name	Parameters and return value	Function Description
int adxl345Read (char * position)	<p>Parameter Description: position: chip data reading module</p> <p>Return Value: Returns 0 on success, failure returns negative</p>	Passed into the file system to read the chip module, the chip read data, the data is calculated and the read data

BMP180 chip interface description

Interface name	Parameters and return value	Function Description
int bmp180Read (int type, int * data)	<p>Parameter Description: type: identifying temperature readings or barometric pressure data: calculated temperature value or pressure value</p>	<p>Passed into the file system to read the chip module, the chip read data,</p> <p><u>Perform calculations on data reads and</u></p>



	Return Value: Returns 0 on success, failure returns negative	according to
--	---	--------------

common

Interface name	Parameters and return value	Function Description
int boardInit ()	Parameter Description: Return Value: Returns the successful development board Model	Initialization development board GPIO

gpio sensor

Interface name	Parameters and return value	Function Description
int sensorInit (struct sensor * dev, int num)	Parameters: dev: number of sensors transfer: transmitting a sensor device num Return Value: Successful returns a file descriptor, error return -1	Sensor initialization access Development Board
void sensorRead (int devFD, char * buf, int len)	Parameter Description: devFD: buf to the file descriptor reading sensor: a buffer for storing data len: the number of bytes to be read Return Value: It returns the size in bytes of success, -1 error	Reading sensor values
void sensorDeinit (int devFD)	Parameter Description: devFD: a sensor to read the file descriptor Return Value: After reading successfully close the file descriptor	Stop or closing sensor



hmc5883

Interface name	Parameters and return value	Function Description
int hmc5883HWInit (int devFD)	Parameter Description: devFD: To write data to a file descriptor Return Value: Successful return value is written, an error return -1.	Chip Initialization data access hmc5883
double hmc5883Read (int devFD)	Parameter Description: devFD: To write data to a file descriptor Return Value: Successful return value is written, an error return -1.	Value calculated compass direction
int hmc5883Init (int i2cDev)	Parameter Description: i2cDev: chip to be written using i2c device Return Value: Successful return value is written, an error return -1.	Open i2c device, set hmc5883
void hmc5883DeInit (int devFD)	Parameter Description: devFD: To close the file descriptor data	Close the file descriptor

led

Interface name	Parameters and return value	Function Description
int getLedState (int ledID)	Parameter Description: ledID: To write data led pin numbers Return Value: LED successful return pin number, an error return -1.	Get LED pin number
int setLedState (int ledID, int ledState)	Parameter Description: ledID: To write data pins led ID ledState: LED display state Return Value: Successful return value is written, an error return -1.	LED display state is provided



追求卓越 创造精品

TO BE BEST

TO DO GREAT

广州友善之臂计算机科技有限公司

LCD1602

Interface name	Parameters and return value	Function Description
int mcpWriteCmd4 (int devFD, unsigned char command)	Parameter Description: devFD: an instruction to write: To write data command file descriptor Return Value: Successful return value is written, an error -1	Write instruction
int mcpWriteCmd8 (int devFD, unsigned char command)	Parameter Description: devFD: an instruction to write: To write data command file descriptor Return Value: Successful return value is written, an error -1	Write eight instructions
int mcpWriteData4 (int devFD, unsigned char data)	Parameter Description: devFD: data to be written: the data to be written to file descriptor data Return Value: Successful return value is written, an error -1	Write data
int mcpWriteData8 (int devFD, unsigned char data)	Parameter Description: devFD: data to be written: the data to be written to file descriptor data Return Value: Successful return value is written, an error -1	Write eight bits of data
int mcplnit (int i2cDev)	Parameter Description: i2cDev: i2c device descriptor Return Value: Successful return value is written, an error -1	Chip initialization data
void mcpDelnit (int devFD)	Parameter Description: devFD: To write data to a file descriptor	Close the file descriptor
int LCD1602KeyDispChar (int devFD, unsigned char x, unsigned char y, unsigned char data)	Parameter Description: devFD: data to be written: the file descriptor to write data x: x-y array of strings: String data row y Return Value: Successful return value is written, an error -1	In the display character row x column y
int LCD1602KeyDispChar (Parameter Description:		In the first line x y column to start writing characters



追求卓越 创造精品

TO BE BEST

TO DO GREAT

广州友善之臂计算机科技有限公司

int devFD, unsigned char x, unsigned char y, unsigned char * str)	devFD: the file descriptor to write data x: x-th column string y: y-line string str: the address data to be written Return Value: Successful return value is written, an error -1	
int LCD1602KeyDispLines (int devFD, char * line1, char * line2)	Parameter Description: devFD: To write data descriptor line1: first line2 LCD display data: display data of the second line of LCD Return Value: Successful return value is written, an error -1	Number of rows displayed character string
int LCD1602KeyInit (i2c dev)	Parameter Description: i2cDev: i2c device descriptor Return Value: Successful return value is written, an error -1	Initialization LCD1602
int LCD1602KeyClear (int devFD)	Parameter Description: devFD: To clear the file descriptor data Return Value: Successful return value is written, an error -1	Clear written data
Void LCD1602KeyDelnit (int devFD)	Parameter Description: devFD: To close the file descriptor data	Close the file descriptor
int LCD1602GetKey (int devFD)	Parameter Description: devFD: To write data to a file descriptor	Or data device LCD1602

OLED

Interface name	Parameters and return value	Function Description
int OLEDInit (int cmdDatPin, int resetPin)	Parameter Description: cmdDatPin: To write commands pin resetPin: reset pin Return Value: Successful return value is written, an error return -1.	OLED initialization
void OLEDDelnit (int devFD)	Parameter Description: devFD: To close the file descriptor data	Close the file descriptor



追求卓越 创造精品

TO BE BEST

TO DO GREAT

广州友善之臂计算机科技有限公司

void OLEDDisp8x16Char (in t devFD, int x, int y, char ch)	<p>Parameter Description:</p> <p>The character display: devFD: the file descriptor to write data x: y LCD display data row: column ch LCD display data</p> <p>Return Value:</p> <p>Successful return value is written, an error -1</p>	<p>LCD device arranged to write a character position</p>
void OLEDDisp8x16Char (in t devFD, int x, int y, char ch [])	<p>Parameter Description:</p> <p>String displayed: devFD: the file descriptor to write data x: y LCD display data row: column ch LCD display data</p> <p>Return Value:</p> <p>Successful return value is written, an error -1</p>	<p>Write a whole string of character string data to the LCD device</p>
int OLEDScreen (int devFD)	<p>Parameter Description:</p> <p>devFD: To clear the file descriptor data</p> <p>Return Value:</p> <p>Successful return value is written, an error -1</p>	<p>Close the file descriptor</p>

PCF8591

Interface name	Parameters and return value	Function Description
int pcf8591Read (int channel, int * value)	<p>Parameter</p> <p>channel: transmission of the file descriptor value:</p> <p>value analog input channels</p> <p>Return Value:</p> <p>Successful return value is written, an error return -1.</p>	<p>Chip read value pcf8591</p>

PWM

Interface name	Parameters and return value	Function Description
int pwmtoGPIO (int pwm)	<p>Parameter Description:</p> <p>pwm: definitions used are several road pwm</p> <p>Return Value:</p> <p>Successful return value is written, an error return -1.</p>	<p>Pwm pin definitions used in gpio</p>



追求卓越 创造精品

TO BE BEST

TO DO GREAT

广州友善之臂计算机科技有限公司

int PWMPlay (int pwm, int freq, int duty) Description:	Parameter pwm: definitions used are several road pwm freq: frequency duty: the duty cycle Return Value: Successful return value is written, an error return -1.	Pwm output
int PWMStop (int pwm) Description:	Parameter pwm: definitions used are several road pwm Return Value: Successful return value is written, an error return -1.	Stop output pwm

Temperature sensor ds18b20

Interface name	Parameters and return value	Function Description
int ds18b20Read (char * temperature)	Parameter Description: temperature: the temperature values acquired Return Value: Successful return value is written, an error return -1.	Reading module to collect temperature values

PCF8574

Interface name	Parameters and return value	Function Description
int pcf8574WriteCmd4 (in t devFD, unsigned char command)	Parameter Description: devFD: writing data file descriptor command: writing instruction Return Value: Successful return value is written, an error return -1.	Write to the module instruction
int pcf8574WriteCmd8 (in t devFD, unsigned char command)	Parameter Description: devFD: writing data file descriptor command: writing instruction Return Value: Successful return value is written, an error return -1.	Write eight instructions
int pcf8574WriteData4 (i nt devFD, unsigned)	Parameter Description: devFD: file descriptor write data	Write data



追求卓越 创造精品

TO BE BEST

TO DO GREAT

广州友善之臂计算机科技有限公司

char data)	command: write data Return Value: Successful return value is written, an error return -1.	
int pcf8574WriteData8 (i nt devFD, unsigned char data)	Parameter Description: devFD: writing data file descriptor command: writing data Return Value: Successful return value is written, an error return -1.	Write eight bits of data
int pcf8574Init (int i2cDev)	Parameter Description: i2cDev: write open i2c device Return Value: Successful return value is written, an error return -1.	I2c device initialization
void pcf8574DeInit (int devFD)	Parameters: devFD: writing data descriptor Return Value: Successful return value is written, an error return -1.	Close pcf8574 equipment