WiringLMK

From BananaPro/Pi

Contents

What is the WiringLMK

WiringLMK is a GPIO access library written in C language for Banana Pro and LeMaker Guitar Base Board Rev.B. It is modified on the base of the original WiringPi for the BCM2835 used in the Raspberry Pi created by Drogon [1] (https://web.archive.org/web/20200222133427/http://wiringpi.com/). The modification done by LeMaker keeps the WiringLMK API usage the same as the original wiringPi.

Support Hardware

- LeMaker Guitar Base Board Rev.B
- LeMaker Banana Pro/Banana Pi

The following information is useful: LeMaker Guitar:Pin Definition on Base Board BananaPro/Pi:Pin definition

Installation

You can donwload and install the WiringLMK by the following commands:

```
$ cd WiringLMK
$ chmod +x ./build
$ sudo ./build
```

Support API

Setup

Here we introduce you the four ways to initialise wiringLMK Pin Mode using the setup functions, one of the following setup functions must be called at the start of the WiringLMK program for initializations:

```
int wiringPiSetup (void) ;
int wiringPiSetupGpio (void) ;
int wiringPiSetupPhys (void) ;
int wiringPiSetupSys (void) ;
```

where `int wiringPiSetupPhys (void) ;` is identical to the physical pin numbering; while `int wiringPiSetupGpio (void) ;` and `int wiringPiSetupSys (void) ;` equals to Broadcom GPIO pin numbers in Raspberry Pi (GPIO.BCM). The differences are listed in the #Appendix: WiringLMK Pin Setup Table for comparison.

Pin Operations

These are the core functions of WiringLMK library that operates directly on SBCs and support other peripheral modules using these basics:

```
void pinMode (int pin, int mode) ;
```

- This sets the `int mode` of a pin to either `INPUT`, `OUTPUT`, `PWM_OUTPUT` (**PWM** output pin only) or `GPIO_CLOCK` (**CLK** output pin only).

```
void pullUpDnControl (int pin, int pud) ;
```

- When the given pin is set as an input, it would be able to set as pull-up or pull-down resistor mode. The parameter `int pud` should be `PUD_OFF` (no pull up/down), `PUD_DOWN` (pull to GND) or `PUD_UP` (pull to VCC 3.3V).

```
void digitalWrite (int pin, int value) ;
```

- When the given pin is set as an output, this function writes the value `HIGH` or `LOW` (`1` or `0`) to the pin.

```
int digitalRead (int pin) ;
```

- To obtain the current logic level at the given pin, this function can read and return the value as `HIGH` or `LOW` (`1` or `0`).

```
analogRead (int pin) ;
```

```
analogWrite (int pin, int value) ;
```

- The given pin should be able to be configured as additional analog modules such that this function can write the given analog value to the supplied analog pin.

```
int wiringPiISR (int pin, int edge, void (*function)(void)) ;
```

- The interrupt function can be specified, the `int edge` can be set to `INT_EDGE_RISING`, `INT_EDGE_FALLING`, or `INT_EDGE_BOTH` for both edges. The third parameter should be the interrupt handler function.

```
void pwmWrite (int pin, int value) ;
```

- Only for the PWM output pin, it can write the value to the PWM register for the given pin. Here it means the physical PWM, not the software PWM.

```
int softPwmCreate (int pin, int initialValue, int pwmRange) ;
```

- This function creates a software controlled PWM pin, it's suggested to set the second parameter `int pwmRange` to 100, where 0 means off and 100 means fully on for the given Pin.

```
void softPwmWrite (int pin, int value) ;
```

- The initialized soft PWM pin can then updates its PWM value on the given pin (`int value` should be in-range) by this function.

Compilation

Before using the WiringLMK GPIO library, you need to include its header file in your programs: `#include <wiringPi.h>`. You may also need additional #include lines, depending on the modules you are using. To compile an example using WiringLMK, you may have to add

```
 -I/usr/local/include -L/usr/local/lib -lwiringPi
```

to the compile line depending on the running environment, the important parameters you might need are listed: `-lwiringPi`, `-lwiringPiDev`, or `-lpthread`.

Simple Example

Here is the "Hello World" example showing how to make an LED blinks:

```
#include <wiringPi.h>
int main (void)
```

```
{
  pinMode (11, OUTPUT) ;
  for (;;)
  {
    digitalWrite (11, HIGH) ; delay (500) ;
    digitalWrite (11,  LOW) ; delay (500) ;
  }
  return 0 ;
}
```

See Also

How to control the IO on the SBC boards

Appendix: WiringLMK Pin Setup Table

Physical Numbering Scheme	WiringLMK Setup Functions		
wiringPiSetupPhys()	wiringPiSetup()	wiringPiSetupGpio()	wiringPiSetupSys()
1	--	--	--
2	--	--	--
3	8	2	2
4	--	--	--
5	9	3	3
6	--	--	--
7	7	4	4
8	15	14	14
9	--	--	--
10	16	15	15
11	0	17	17
12	1	18	18
13	2	27	27
14	--	--	--
15	3	22	22
16	4	23	23
17	--	--	--
18	5	24	24
19	12	10	10
20	--	--	--
21	13	9	9
22	6	25	25
23	14	11	11
24	10	8	8
25	--	--	--
26	11	7	7
27	30	0	0
28	31	1	1
29	21	5	5
30	--	--	--
31	22	6	6
32	26	12	12
33	23	13	13
34	--	--	--
35	24	19	19
36	27	16	16
37	25	26	26
38	28	20	20

			--
40	29	21	21

Retrieved from "http://wiki.lemaker.org/index.php?title=WiringLMK&oldid=6065"