LMK.GPIO

From BananaPro/Pi

Contents

What is the LMK.GPIO

LMK.GPIO provides a class to control the GPIO on Banana Pro and LeMaker Guitar Base Board Rev.B written in Python language. It is modified on the base of the original RPi.GPIO for the BCM2835 used in the Raspberry Pi [1] (https://web.archive.org/web/20200214112749/https://pypi.python.org/pypi/RPi.GPIO). Note that the current release does not support SPI, I2C, hardware PWM or serial functionality on the LMK.GPIO yet. The modification done by LeMaker keeps the LMK.GPIO API usage the same as the original RPi.GPIO.

Support Hardware

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

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

LEMAKER HOMEDOWNLOADFORUMBLOGPRODUCT

You can donwload and install the LMK.GPIO by the following commands:

```
$ sudo apt-get install python-dev
$ git clone https://github.com/LeMaker/LMK.GPIO.git
$ cd LMK.GPIO
$ python setup.py install
$ sudo python setup.py install
```

Support API

Setup

Here we introduce you how to initialise LMK.GPIO Pin Mode using the setup function `GPIO.setmode(GPIO.MODE)`, which must be called at the start of the LMK.GPIO program for initializations: the physical Pin numbering scheme

```
GPIO.setmode(GPIO.BOARD)
GPIO.setmode(GPIO.BCM)
```

where `GPIO.BOARD` is identical to the physical pin numbering; while `GPIO.BCM` equals to Broadcom GPIO pin numbers. The differences are listed in the #Appendix: LMK.GPIO Pin Setup Table for comparison.

Pin Operations

These are the core functions of LMK.GPIO library that operates directly on SBCs and can support more peripheral modules using these basics:

```
GPIO.setup(int pin, int mode)
```

- This sets the `int mode` of a pin to either `GPIO.IN` (as input) or `GPIO.OUT` (as output).

```
GPIO.setup(int pin, GPIO.IN, 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 `GPIO.PUD_OFF` (no pull up/down), `GPIO.PUD_DOWN` (pull to GND) or `GPIO.PUD_UP` (pull to VCC 3.3V).

```
GPIO.output(int pin, int value)
```

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

```
GPIO.input(int pin)
```

```
GPIO.add_event_detect(unsigned int pin, unsigned int edge,callback = void (*func_my_callback), unsigned int boun
```

- The interrupts are regarded as events that happened as `GPIO.RISING`, `GPIO.FALLING` or `GPIO.BOTH` for both edges in the second parameter of function. The third parameter should be the interrupt event callback function.

```
p = GPIO.PWM(int pin, int frequency)
```

- To creates a software controlled PWM pin, it's also suggested to set the second parameter `int frequency` to 100 (in theory, `frequency > 0` is sufficient) by this function. The initial value of this soft PWM is specified by `.start(int initialValue)`, and can be stopped by function `.stop()`. The duty cycle can be updated by `.ChangeDutyCycle(int dutyCycle)` where `int dutyCycle` should also be in-range.

Compilation

Python language doesn't require to compile, but you have to import it as `import LMK.GPIO` in the very beginning of the program.

Simple Example

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

```
#!/usr/bin/env python
import LMK.GPIO as GPIO
import time
#LED Mode BOARD
PIN_NUM = 7
GPIO.setmode(GPIO.BOARD)
while True:
        try:
                GPIO.setup(PIN_NUM, GPIO.OUT)
        except:
                print("Failed to setup GPIO %d", PIN_NUM)
        GPIO.output(PIN_NUM, True)
        time.sleep(0.5)
        GPIO.output(PIN_NUM, False)
        time.sleep(0.5)
```

See Also

How to control the IO on the SBC boards

Appendix: LMK.GPIO Pin Setup Table

LEMAKER HOMEDOWNLOADFORUMBLOGPRODUCT

Physical Numbering Scheme	LMK.GPIO	Raspberry Pi
LMK.GPIO.BOARD	**LMK.GPIO.BCM**	**BCM Chip Serial Number**
1	**--**	**3.3V**
2	**--**	**5V**
3	**2**	**2/GPIO02**
4	**--**	**5V**
5	**3**	**3/GPIO03**
6	**--**	**GND**
7	**4**	**4/GPIO04**
8	**14**	**14/GPIO14**
9	**--**	**GND**
10	**15**	**15/GPIO15**
11	**17**	**17/GPIO17**
12	**18**	**18/GPIO18**
13	**27**	**27/GPIO27**
14	**--**	**GND**
15	**22**	**22/GPIO22**
16	**23**	**23/GPIO23**
17	**--**	**3.3V**
18	**24**	**24/GPIO24**
19	**10**	**10/GPIO10**
20	**--**	**GND**
21	**9**	**9/GPIO09**
22	**25**	**25/GPIO25**
23	**11**	**11/GPIO11**
24	**8**	**8/GPIO8**
25	**--**	**GND**
26	**7**	**7/GPIO07**
27	**0**	**0/GPIO00**
28	**1**	**1/GPIO01**
29	**5**	**5/GPIO05**
30	**--**	**GND**
31	**6**	**6/GPIO06**
32	**12**	**12/GPIO12**
33	**13**	**13/GPIO13**
34	**--**	**GND**
35	**19**	**19/GPIO19**
36	**16**	**16/GPIO16**
37	**26**	**26/GPIO26**
38	**20**	**20/GPIO20**

40	21	21/GPIO21