

```
$ cd ~  
$ mkdir my_LeScratch
```

LeScratch needs to pre-install some packages: python-dev, python-setuptools, i2c-tools, python-smbus:

```
$ sudo apt-get install python-dev  
$ sudo apt-get install python-setuptools python-smbus i2c-tools
```

Since I2C driver is installed and enabled by default, you only have to install the SPI drivers, now open the terminal and write the following commands to install:

```
$ cd ~/my_LeScratch  
$ git clone https://github.com/doceme/py-spidev  
$ cd py-spidev  
$ sudo python setup.py install
```

Please also remember to download the latest version of LMK.GPIO for Banana Pro and LeMaker Guitar as in the below, before installing the new version, please delete the old version first:

```
$ cd /usr/local/lib/python2.7/dist-packages  
$ ls  
$ sudo rm -r f LMK  
$ sudo rm LMK.*
```

Then open the terminal and write the following commands to install the LMK.GPIO library:

```
$ cd ~/my_LeScratch  
$ git clone https://github.com/LeMaker/LMK.GPIO  
$ cd LMK.GPIO  
$ python setup.py install  
$ sudo python setup.py install
```

LeScratch communicates with some sensors using wiringLMK library. In case of you don't have the package, please following:

```
$ cd ~/my_LeScratch  
$ git clone https://github.com/LeMaker/WiringLMK.git  
$ cd WiringLMK  
$ chmod +x ./build  
$ sudo ./build
```

System setup

The expansion boards communicate with the SBCs through various communication interfaces. LeMaker Guitar enables SPI and I2C interfaces by default. The I2C module is included in the latest Banana Pro distributions and is enabled by default. But the SPI driver should be enabled manually by:

```
$ sudo modprobe spi-sun7i
```

LeScratch User Guide

From BananaPro/Pi

Contents

- 1 Introduction
- 2 Pre-requisites
 - 2.1 System setup
 - 2.2 Scratch Mesh setup
 - 2.3 Running LeScratch
- 3 Scratch Command Examples
 - 3.1 Example 1: LED Board
 - 3.1.1 Instructions
 - 3.1.2 GPIO commands
 - 3.2 Example 2: LN USB Hub
 - 3.2.1 Instructions
 - 3.2.2 I2C commands
 - 3.3 Example 3: LN Digital
 - 3.3.1 Instructions
 - 3.3.2 SPI commands
 - 3.4 Example 4: LN IO Board
 - 3.4.1 Instructions
 - 3.4.2 AD/DA commands
 - 3.4.3 LCD commands
- 4 See Also
- 5 Appendix: LeScratch User Manual

Introduction

Scratch@MIT education software: Scratch 1.4 is designed with learning and education in mind. It can run in cooperation with LeScratch, which is the Scratch message handler that runs in the background on Banana Pro to let Scratch to communicate with hardware peripherals. For education purpose, people at any age can earn an easy access to the principles of microcomputers and sensors with LeScratch, which is implemented in Python to extend the functionalities that sending and receiving commands among all the peripherals. Download link:
<https://github.com/LeMaker/LeScratch/>

Pre-requisites

```
$ sudo nano /etc/modprobe.d/bpi-blacklist.conf
```

Then you still have to load the SPI modules by adding "spi-sun7i" and "spidev" to "/etc/modules":

```
$ sudo nano /etc/modules
```

Scratch Mesh setup

To modify Scratch Mesh and save it permanently, you need to run scratch with sudo permission. Please refer to [http://wiki.scratch.mit.edu/wiki/Mesh_\(1.3_Mesh_by_Modification_of_Scratch\)](http://wiki.scratch.mit.edu/wiki/Mesh_(1.3_Mesh_by_Modification_of_Scratch)) for detailed steps. (Note, you may find that step7: you need to middle-click on the System Brower and select accept)

```
$ sudo scratch
```

Reboot after setting up the above instructions.

Running LeScratch

Before starting LeScratch program, it is suggested to start the Scratch 1.4 (saved in Mesh mode) at first such that the connections will be built once the LeScratch script runs. The file LeScratch.py is implemented with Python can be run easily:

```
$ cd ~/my_LeScratch
$ git clone https://github.com/LeMaker/LeScratch
$ cd LeScratch
$ sudo python LeScratch.py
```

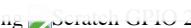
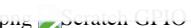
Scratch Command Examples

Example 1: LED Board

Instructions

General-purpose input/output (GPIO) is a generic pin on an integrated circuit whose behavior can be controlled by the user at run time, including whether it is an input or output pin. GPIO pins have no special purpose defined, and go unused by default. Number out of the list defined by RPi.BCM [4, 5, 6, 12, 13, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27] are usually available. The GPIO board has 40 LEDs and 15 switches that can be used to indicate the status of all the 40 pins on Banana Pro and LeMaker Guitar Base Board Rev.B, and give digital input to its corresponding switch pins. In the first place, remember to declare which GPIOs are going to be used by: broadcast g[num]in and g[num]out, "in" means input pin and "out" means output pin. broadcast g[num]on, g[num]off, "on" means set output to 1/high and "off" means set to 0/low.

Commands	g[num]in	g[num]out	g[num]on	g[num]off
Functions	Define input PIN	Define output PIN	Set pin to 1/high	Set pin to 0/low

Example 2: LN USB Hub

Instructions

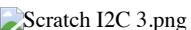
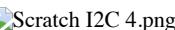
The USB Hub expansion board has 4 Port USB Hub & 23017 x2 32GPIO to communicate with SBCs such that Scratch can control with extended functions. To specify the I2C address for Scratch, use the following formats:

Command "i2+" + "address 0x(20-27)" + "a" +"bit (0 to 7)" for Port A Command "i2+" + "address 0x(20-27)" + "b" +"bit (0 to 7)" for Port B Command "bit"+ "address 0x(20-27)" + "a" +"bit (7 to 0)" for Port A Command "bit"+ "address 0x(20-27)" + "b" +"bit (7 to 0)" for Port B

In details, you can refer to the following examples: i221a1 => i2c address 0x21 Port A bit 1 ON i222b4 => i2c address 0x22 Port B bit 4 ON bit22b01010101 => address 0x22 port B from bit 7 to 0, output => 0b01010101 bit21a01010101 => address 0x21 port A from bit 7 to 0, output => 0b01010101 bit21aon => address 0x21 Port A all ON, 0b11111111 bit21boff => address 0x21 Port B all OFF/clear, 0b00000000 bit22aoff => address 0x22 Port A all OFF/clear

I2C commands



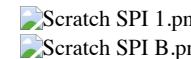
Example 3: LN Digital

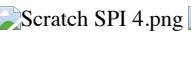
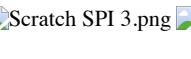
Instructions

LN Digital has one 16 bits mcp23s17 that communicates with SBCs using SPI. Mcp23s17 has 8 various addresses that allow extending 8 boards at the same time. LN Digital has been configured as 8 bits port A and port B or 16 bits. Each port (A/B) can be configured as either input or output. By default it is configured as port A output (1 to 8), port B input (1 to 8). To specify SPI commands in details, you can refer to the following: Command "sp"+ "address (0-7)" + "a" +"bit (0 to 7)" for Port A Command "sp"+ "address (0-7)" + "b" +"bit (0 to 7)" for Port B Command "bits"+ "address (0-7)" + "a" +"bit (7 to 0)" for Port A Command "bits"+ "address (0-7)" + "b" +"bit (7 to 0)" for Port B

For examples: (address 0-7 = 0x40-4E): sp0a1 => spi address 0x40 Port A bit 1 ON sp1b4 => spi address 0x42 Port B bit 4 ON bits2b01010101 => address 0x44 port B from bit 7 to 0, output => 01010101 bits3a01010101 => address 0x46 port A from bit 7 to 0, output => 01010101 bits4aon => address 0x48 Port A all ON, 0b11111111 bits5boff => address 0x4A Port B all OFF/clear, 0b00000000 bits6aoff => address 0x4B Port A all OFF/clear

SPI commands






Instructions

IN IO board has exported a GPIO_CON (GPIO0~7), eight LEDs that corresponds to GPIO0~7, four switches, a LCD slot (for LCD1602 and LCD12864), two groups of SPI and UART interfaces, and one groups of I2C interface. Here the PCF8591 and LCD1602 are used as examples of LN IO board. PCF8591 is a single-chip, single-supply low power 8-bit CMOS data acquisition device with four analog inputs and one analog output.

AD/DA commands



LCD commands



See Also

How to control the IO on the SBC boards

Appendix: LeScratch User Manual

Please refers to the LeScratch User Manual for the full usage version of LeScratch. The `./doc/` and `./examples/` are helpful for beginners to LeScratch. <https://github.com/LeMaker/LeScratch>

Retrieved from "http://wiki.lemaker.org/index.php?title=LeScratch_User_Guide&oldid=5988"