LeScratch User Manual

文件编号：

文件版本：A

制　　定：Alessia

审　　核：

批　　准：

生效日期：

版本号	修订日期	修订内容	修订人	生效日期
A	2015/5/23	Initial version	Alessia	
B	2015/11/3	SBC 40 Pins Compatible version	Alessia	

Table of Contents

Foreword

This manual is oriented to beginner users of Banana Pro/LeMaker Guitar Base Board Version.B, which provides detailed operation and usage guidelines of LeScratch educational software suite, on this basis, to facilitate a custom interface for the user. Before the start point, please read the following advices to get ready:

1. Be familiar about the LMK.GPIO library and WiringLMK library, the difference between the GPIO.BCM and wiringPiSetup() is listed in Appendix 3.1. You can also check up the table of pin mapping to the physical ones when you are defining your own pins.
2. By default, this manual use BCM pin definition of LMK.GPIO library. But for some fast read and write frequency of sensors requirements, some modules is using wiringPiSetup() pin definition of WiringLMK library. Currently only in Section 2.7 Humidity and Temperature sensor and Section 2.9 AD / DA convertor, LeScratch uses the wiringPiSetup () mode.
3. The 40 Pins on Banana Pro and LeMaker Guitar Base Board Version. B are fully compatible. Be aware of how many VCC power pins and GND ground pins on the board since you have to know how to power on the sensors. Besides, this manual gives sensor connections in each section and most modules allow users to customize their own data pin.

1 Instruction and Installation

Scratch is designed with learning and education in mind. A wide variety of educators have been supporting Scratch creators since 2007. Scratch is a programming language that makes it easy to create interactive art, stories, simulations, and games – and share those creations online. Scratch is designed especially for ages 8 to 16, but is used by people of all ages. Millions of people are creating Scratch projects in a wide variety of settings, including homes, schools, museums, libraries, and community centers.

LeScratch is a Scratch message handler that runs in the background to let the Scratch client to communicate with hardware. For education purpose, people at any age can earn an easy access to the principles of microcomputers. When people learn to code in Scratch, they learn important strategies for solving problems, designing projects, and sharing ideas. In combined with LeScratch, people get the extended functions to control sensors with its original Scratch functions, which make it more fun and applicable.

The powerful microcomputers Banana Pro/LeMaker Guitar, offers a wealth of hardware interfaces and communication buses. In this manual, I will firstly introduce you how to install and configure LeScratch package, and then I will explain in details about how we extend the functionalities of Scratch 1.4 to control the hardware peripherals and sensors. Finally the demonstration that uses most of the sensor modules: LeScratch Smart House will be introduced, which is built with the LeScratch handler running in cooperation with Scratch 1.4 client on SBCs.

1.1 Pre-requisites

In this manual, we're using system image version "Raspbian For BananaPro v1412" for Banana Pro, and "Ubuntu Mate For Guitar" for LeMaker Guitar; please download the correct version at:

http://www.lemaker.org/product-bananapro-download-16.html

http://www.lemaker.org/product-guitar-download-18.html

Then let's declare project directory under the home directory, all the operations in the following are based on this directory:

```
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
```

For LeMaker Guitar both I2C and SPI modules are automatically enabled. Since I2C driver is installed and enabled by default, you only have to enable the SPI drivers; open the terminal and download the source code of the Python module: py-spidev for both boards, then install py-spidev as following:

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

In this content, we need another Python module that is originally developed to control Raspberry Pi GPIO (General Purpose Input Output). Since the 40 Pins on Banana Pro/LeMaker Guitar are fully compatible with the Raspberry Pi ones, we modified it into LMK.GPIO Python module for Banana Pro. Before installing the new version, please delete the old version first:

```
cd /usr/local/lib/python2.7/dist-packages
ls
sudo rm –rf LMK
sudo rm LMK.*
```

Then download the latest version of LMK.GPIO, open the terminal and write the following commands to install the module:

```
sudo apt-get update
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. To install this library, please following, uninstall the previous installations first and then install the latest one:

```
git clone https://github.com/LeMaker/WiringLMK.git -b bananapro
cd WiringLMK
chmod +x ./build
sudo ./build uninstall
sudo ./build
```

1.1.1 System setup

The extension boards communicate with the Banana Pro/LeMaker Guitar through various interfaces. Both I2C and SPI modules are enabled automatically for LeMaker Guitar. The I2C module is included in the latest Banana Pro distributions and is enabled by default. But the SPI driver should be enabled manually: To temporarily enable the SPI module for the modprobe system, you need to use:

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

Note that once the system is restarted, this change won't work anymore. If you want to permanently enable it, you can use the following command to modify the file ``/etc/modprobe.d/bpi-blacklist.conf``, simply commenting out line ``blacklist spi-sun7i`` as ``#blacklist spi-sun7i``:

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

If you're using other versions of system image, you might find out the blacklist file is removed such that the file you opened above is empty, in this case you can move on.

Now you can load the SPI modules by adding ``spi-sun7i`` and ``spidev`` (one module per line, no separators) to ``/etc/modules`` by the following command:

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

Thus, each time the system boot will automatically load both spi and i2c modules from the /etc/modules.

1.1.2 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) for detailed steps. (Note, you may find that step7: you need to middle-click on the System Browser and select accept)

```
$ sudo scratch
```

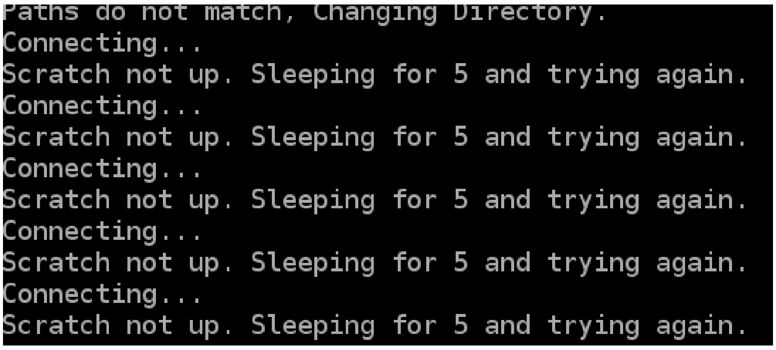
Reboot after setting up the above instructions.

1.2 Running LeScratch

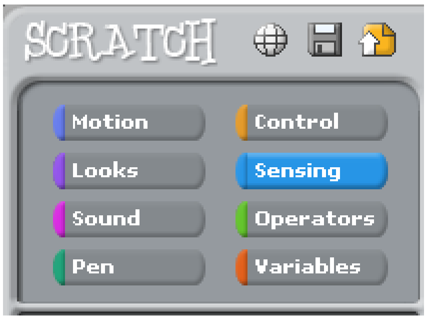
Before starting LeScratch program, it is suggested to start the Scratch 1.4 (saved in Mesh mode) at first such that the connections will be built once the LeScratch script runs. First you need to download LeScratch from LeMaker Github. The file LeScratch.py is implemented with Python can be run easily:

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

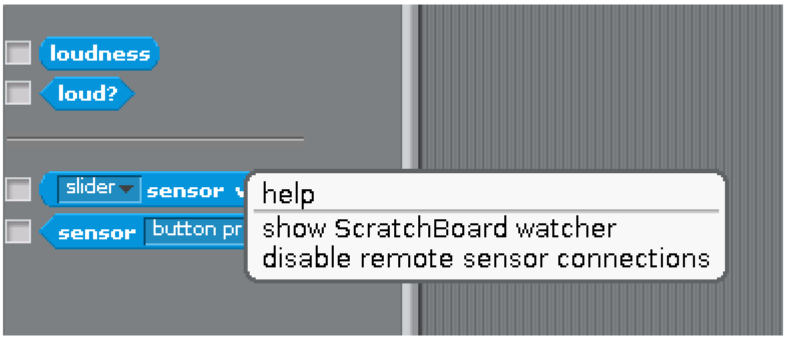
If the following phenomenon happens, it means the connection between LeScratch and Scratch is failed, thus sending or receiving message is not possible：



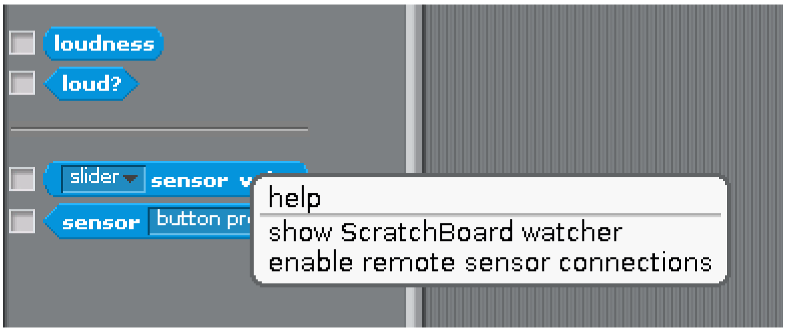
The solution is simple; you need to find the Sensing Block as it shows in the picture:



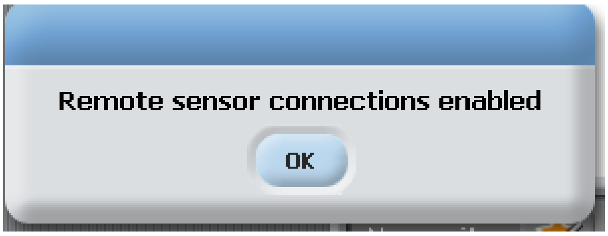
Then right-click on the block <slider>sensor value， select <disable remote sensor connections>.



Again, you right-click on the block <slider>sensor value， select <enable remote sensor connections>.

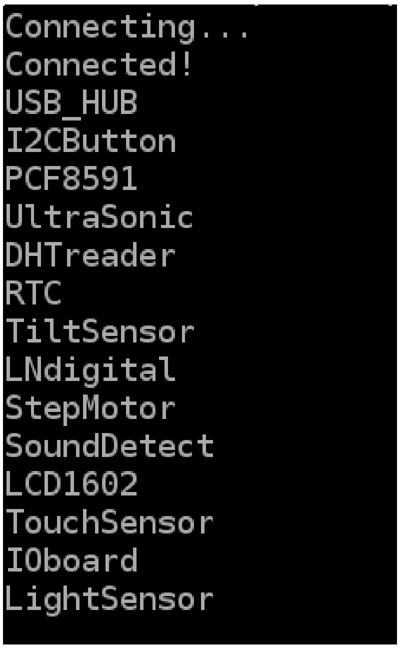


Then you will receive a message as in the following that means the remote sensor connection is built:



When you already run the LeScratch handler, and you load a new demo of Scratch, usually you need to wait a bit until the connection between LeScratch and Scratch is built, as it shows in the below, which means the communication is successful. Otherwise you need to restart the LeScratch handler:

When the LeScratch handler shows the following in the terminal, it means running good.



2 Scratch Commands

2.1 GPIO (General Purpose Input Output)

2.1.1 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 LMK.BCM [4, 5, 6, 12, 13, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27] are usually available. Note that the 40 Pins definition LMK.BCM is compared in the Appendix with other ways of Pins definitions. 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/LeMaker Guitar, and give digital input to its corresponding switch pins. Currently this board is not for public sales, but you can write an email to product@lemaker.org and tell us your needs.

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.

The usages of the GPIO controls:

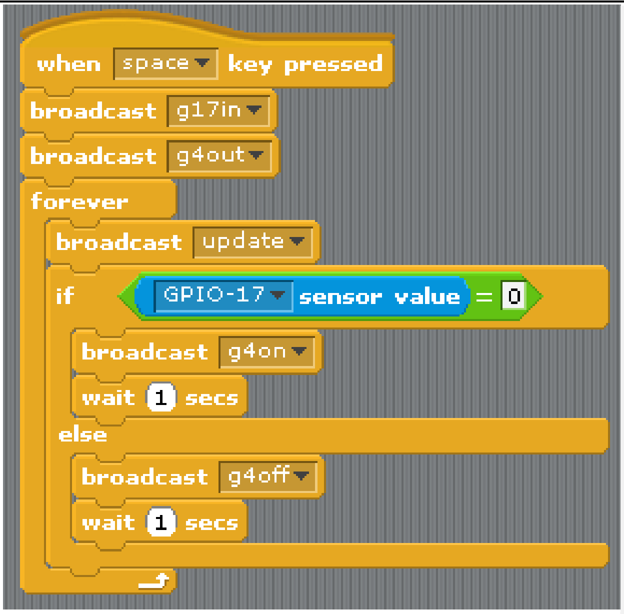
Commands	g[num]in	g[num]out	g[num]on	g[num]off
Functions	Define input PIN	Define output PIN	Set pin to 1/high	Set pin to 0/low

2.1.2 Example: LED GPIO Board

Click on the Green Flag to Run:
Declare GPIO 17 as output

While true
Set GPIO 17 to on/high
Wait for 1 second

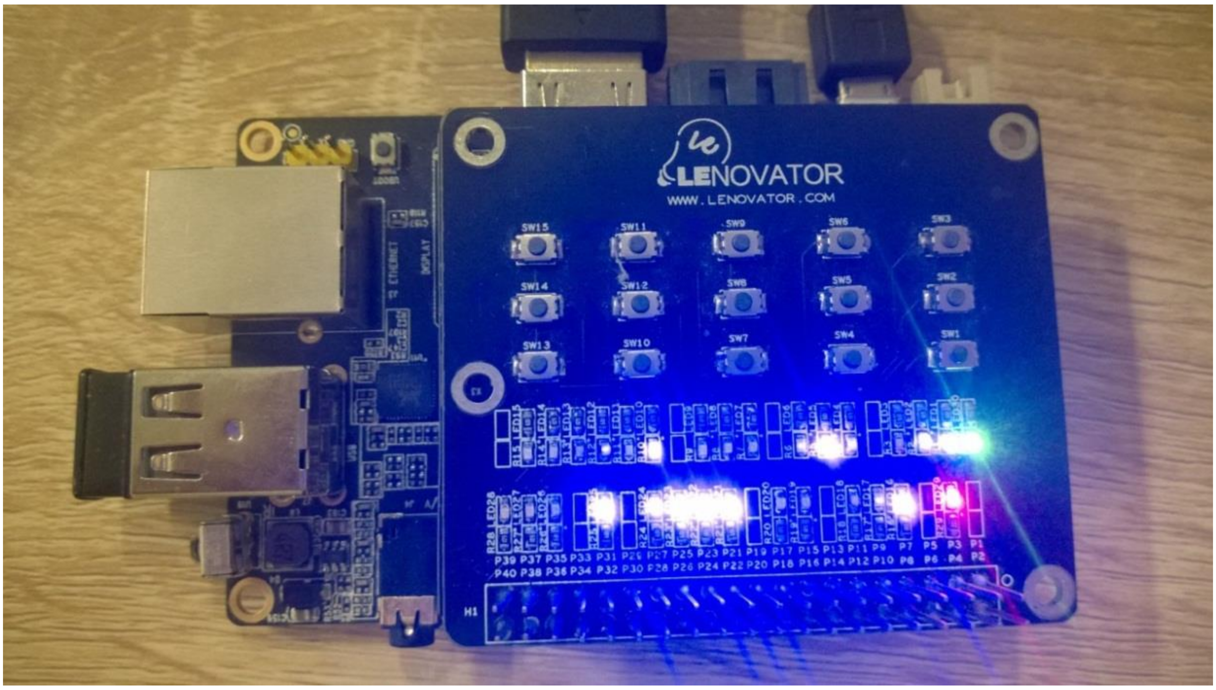
Set GPIO 17 to off/low
Wait for 1 second



When keyboard Space is pressed
Declare GPIO 17 as input
Declare GPIO 4 as output
While true
Update step

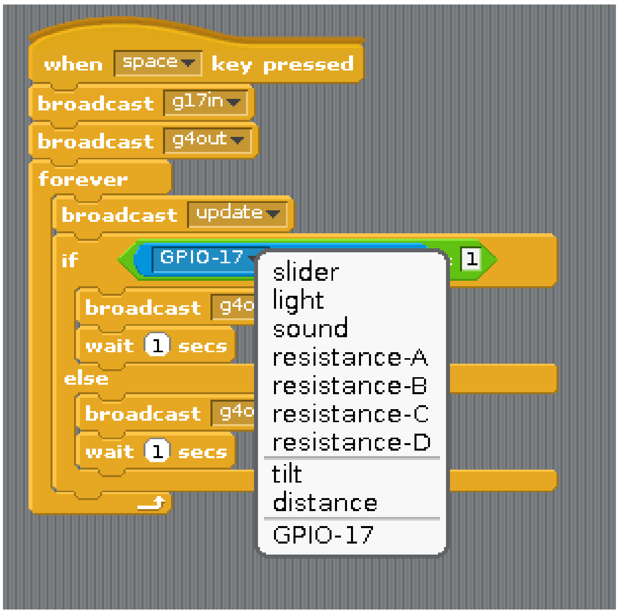
Check if GPIO 17 = 0
Set GPIO 4 to on/high
Wait for 1 second
Else
Set GPIO 4 off/low
Wait for 1 second

Click on the Green Flag to Run:
Declare GPIO 12, 13, 19, 16, 6 as output
While true
Set GPIO 12, 13, 19, 16, 6 to on/high, wait for 1 second
Set GPIO 12, 13, 19, 16, 6 to off/low,
Wait for 1 second

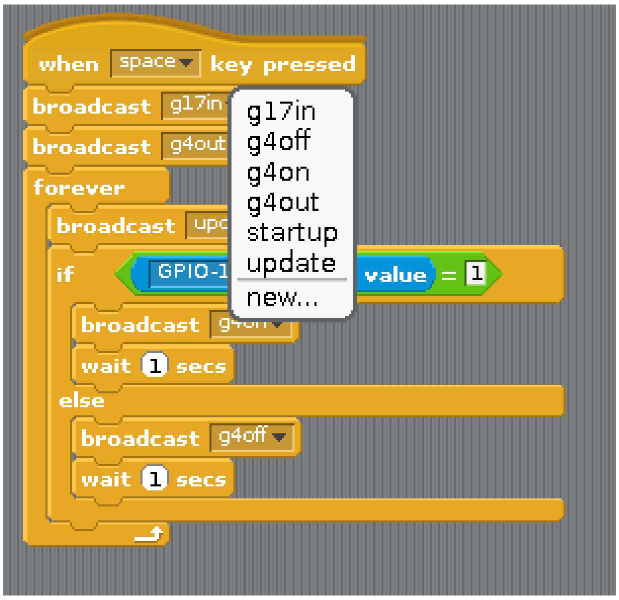


Note that, if you're trying to define your own GPIO input from the available list [4, 5, 6, 12, 13, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27], you will have to run LeScratch once to obtain the GPIO input list.

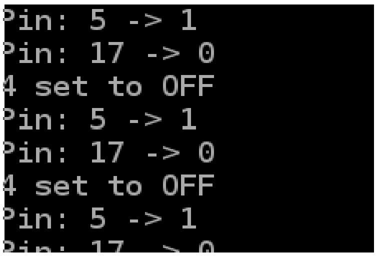
For example, here you add GPIO 5 as input, but you can't find the name GPIO-5 in the sensor list.

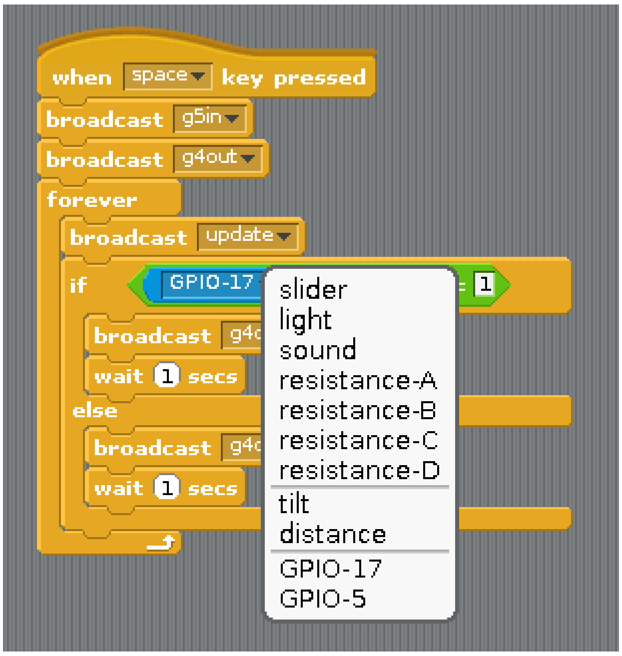


The operations is as simple as below, you create a new broadcast g5in:



Then you run the LeScratch.py to see if the terminal displays Pin 5 etc, if so then it means the GPIO-5 is successfully updated in the Sensor list, you can now choose it from the block:





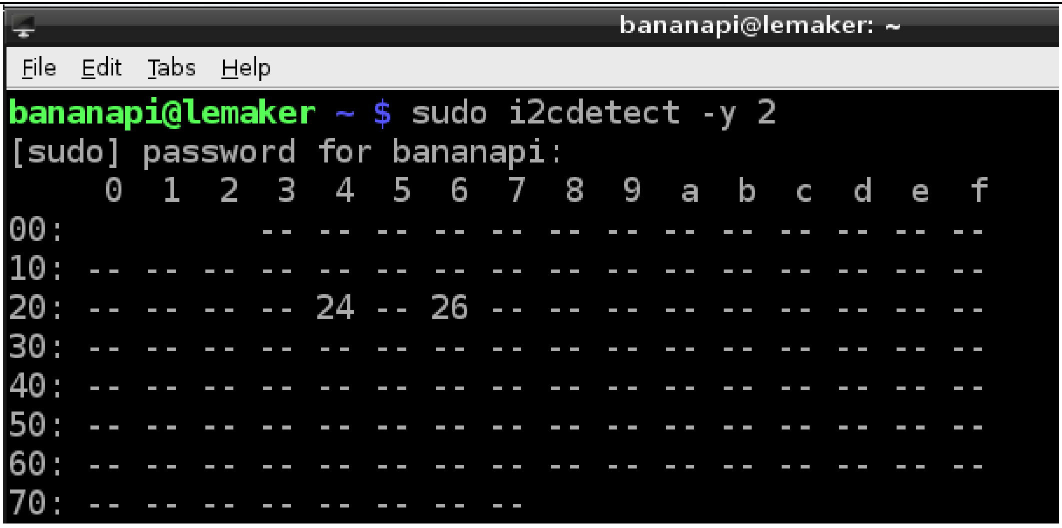
2.2 I2C (Inter-Integrated Circuit)

2.2.1 Instructions

I^{2}C (Inter-Integrated Circuit) is a multi-master, multi-slave, single-ended, serial bus, which is easy to control and has high communication speed. I2C bus has SDA data line and SCL clock line that control the sending and receiving of data. I2C has various addresses for extended devices, once you attach an extension board to Banana Pro/LeMaker Guitar, open the terminal and use i2c-tools to detect its address (The USB Hub-32IO extension board is used in this section):

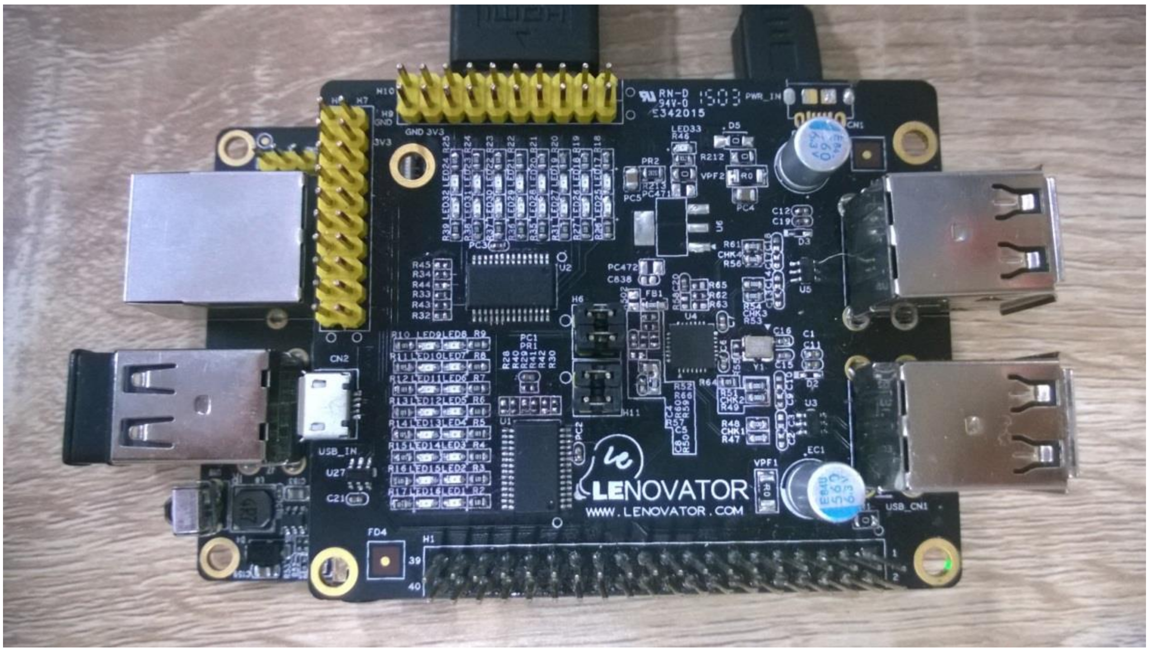
```
$ sudo i2cdetect -y 2
```

If you have problem to connect your device, you might get wrong address or empty results; If succeed, you can find the address of your i2c device displayed as below, in our case, the USB Hub-32IO expansion board has two i2c addresses 0x24 and 0x26:



```
bananapi@lemaker: ~
File  Edit  Tabs  Help
bananapi@lemaker ~ $ sudo i2cdetect -y 2
[sudo] password for bananapi:
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- 24 -- 26 -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

Below picture is the USB-HUB-32IO from Lenovator Purchase link: http://www.lenovator.com/product/26.html:



The USB Hub-32IO expansion board has 4 Port USB Hub & MCP23017 x2 32GPIO to communicate with Banana Pro/LeMaker Guitar such that Scratch can control with extended functions. To specify the I Ȼ address of MCP23017 or other I Ȼ device for Scratch, use the following formats:

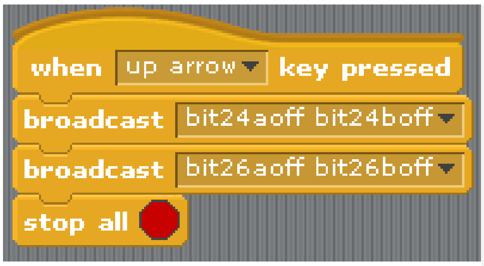
```
Command "i2"+ "address 0x(20-27)" + "a" +"bit (0 to 7)" for Port A
Command "i2"+ "address 0x(20-27)" + "b" +"bit (0 to 7)" for Port B
Command "bit"+ "address 0x(20-27)" + "a" +"bit (7 to 0)" for Port A
Command "bit"+ "address 0x(20-27)" + "b" +"bit (7 to 0)" for Port B
```

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
```

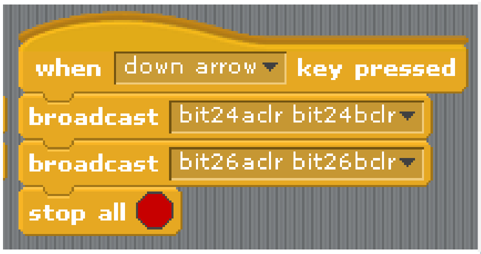
2.2.2 Example: USB Hub-32IO



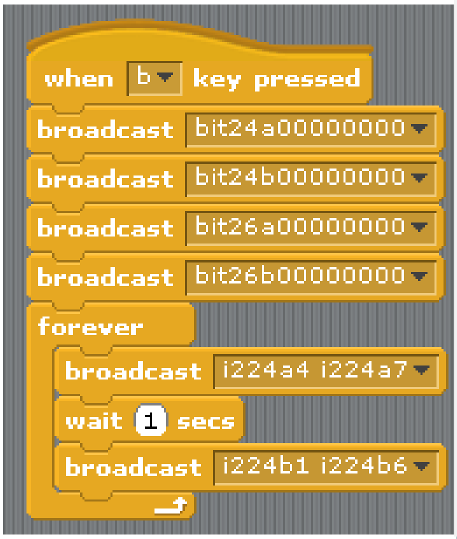
When up arrow is pressed
(address 0x24, 0x26)
Clear port A, B

When keyboard a is pressed
(address 0x24, 0x26)
Clear port A, B

While true
Set address 0x24 port A to 0b10010111
Set address 0x24 port B to 0b01101000



When down arrow is pressed
(address 0x24, 0x26)
Clear port A, B



When keyboard b is pressed
(address 0x24, 0x26)
Clear port A, B

While true
Set address 0x24 port A bit4 and bit7 to 1, wait for 1 second
Set address 0x24 port B to bit and bit6 to 1

2.3 SPI (Serial Peripheral Interface)

2.3.1 Instructions

The Serial Peripheral Interface (SPI) bus is a synchronous serial communication interface specification used for short distance communication, which is in full duplex mode using Master-Slave architecture with a single master. The master device originates the frame for reading and writing. Multiple slave devices are supported through selection with individual slave select lines. There're typically 4 lines for connections: (Purchase link: http://www.lenovator.com/)

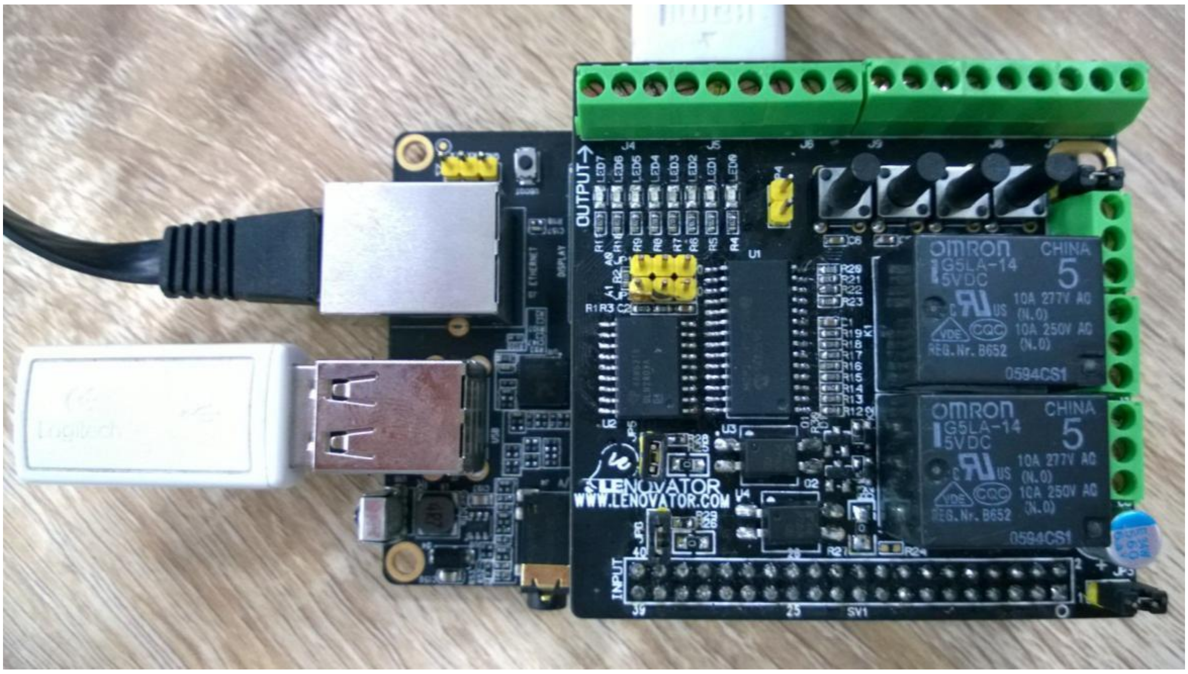
（1） MOSI: Master Device output data, Slave Device input data

（2） MISO: Master Device input data, Slave Device output data

（3） SCLK: Clock generated by Master Device

（4） CS: Chip (Slave Device) select line controlled by Master Device

For example, our expansion board LN Digital has one 16 bits MCP23s17 that communicates with Banana Pro using SPI (Purchase link: http://www.lenovator.com/product/27.html). 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).



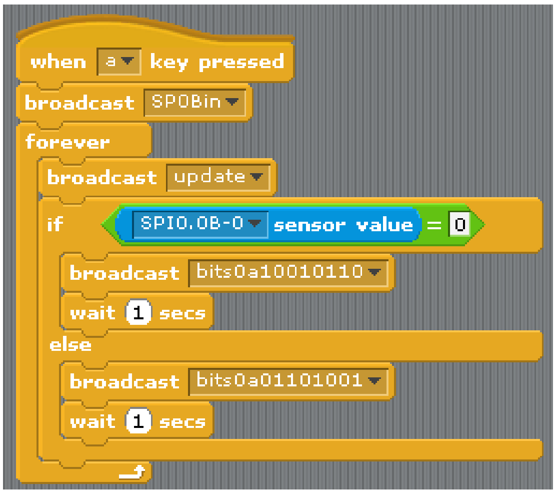
To specify SPI commands in details, you can refer to the following:

```
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
```

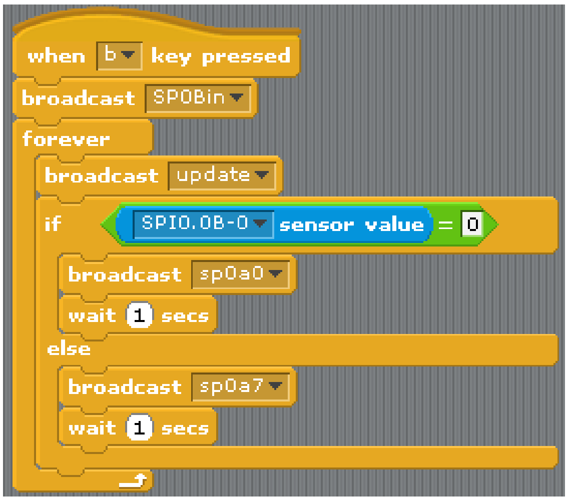
For examples: (address 0-7 = 0x40-4E):

```
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
```

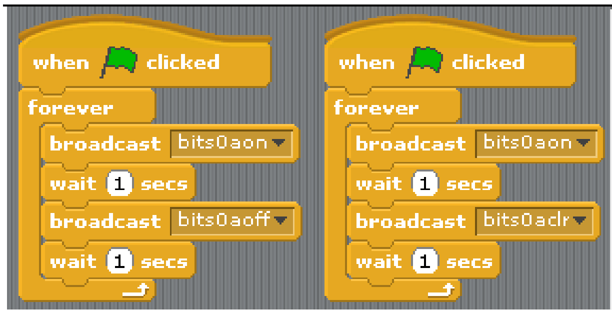
2.3.2 Example: LN Digital (SPI General)



When keyboard a is pressed
Declare address 0x40 port B as input
While true
Update step. Check:
If SPI address 0x40 port B bit0 = 0
Set SPI address 0x40 port A to
0b10010110, wait for 1 second
Else
Set SPI address 0x40 port A to
0b01101001, wait for 1 second



When keyboard b is pressed
Declare address 0x40 port B as input
While true
Update step. Check:
If SPI address 0x40 port B bit0 = 0
Set address 0x40 port A bit0 to 1
Wait for 1 second
Else
Set address 0x40 port A bit7 to 1
Wait for 1 second



Click on the Green Flag to Run:
While true
Set SPI address 0x40 port A to all on, wait for 1 second
Set SPI address 0x40 port A to all off, wait for 1 second

2.3.3 Example: LN Digital (LNDI commands)

The extension board LN Digital allows Scratch to access the following extra hardware:

a) 8 Open-Collector Outputs

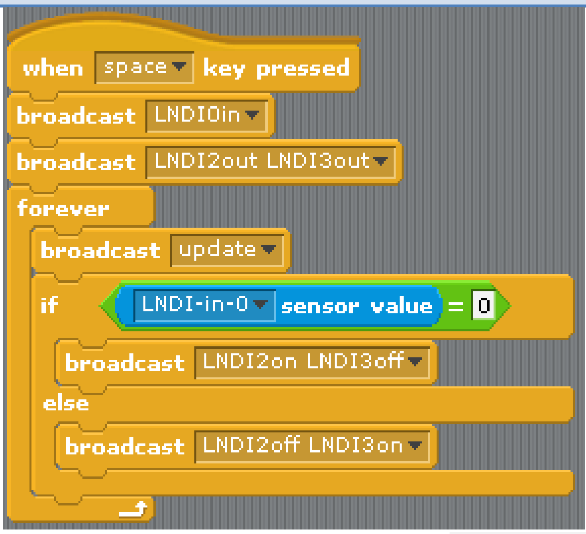
b) 8 LED Indicators

c) 8 Digital Inputs

d) Tactile Switches (The interrupt event is set to listen on the 4 inputs switch.)

e) 2 Changeover Relays (Port A output bit1 – relay 1, Port A output bit2 – relay 2.)

Commands	LNDI[num]in	LNDI[num]out	LNDI[num]on	LNDI[num]off
Functions	Listen to key input	Declare output pin	Set pin to 1/high	Set pin to 0/low



When space is pressed
LNDI1out, LNDI2out, LNDI3out,
Declare LNDI1in,
While true
Update info
If LNDI1in is detected: input button pressed
Set LNDI1, 3 out to on/high
Set LNDI2 out to off/low
Else (no input is detected)
Set LNDI1, 3 out to off/low
Set LNDI2 out to on/high

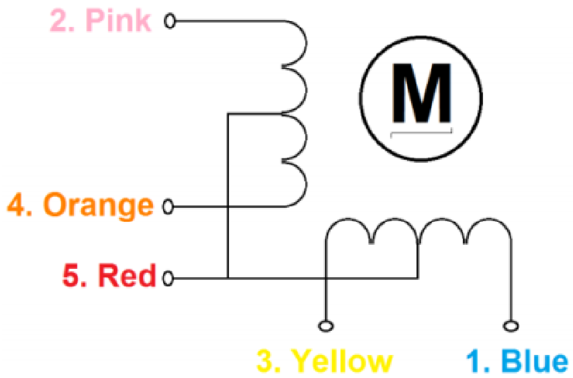
Click the Green Flag to Run:
Declare LNDI1, LNDI2 output
While true

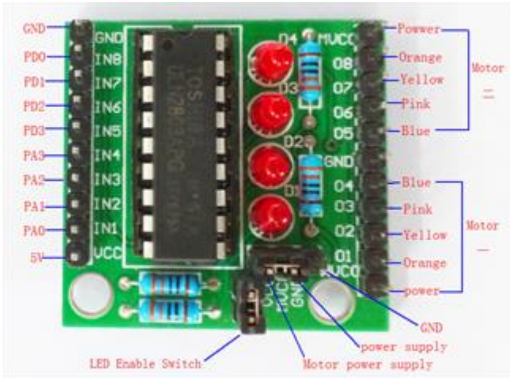
LNDI1out set to on/high, LNDI2 out set to off/low
Wait for 1 second
LNDI1 out set to off/low, LNDI2 out set to on/high
Wait for 1 second

2.4 Step Motor

2.4.1 Technical Specification

The 28BYJ-48 stepper motor is a so-called unipolar motor. A unipolar stepper motor has two or more windings, each with center tap. Each section of windings is switched on for each direction of magnetic field. Since in this arrangement a magnetic pole can be reversed without switching the direction of current, the commutation circuit can be made very simple (e.g., a single transistor) for each winding. In this project the ULN2803A Integrated Circuit is used.





There are a number of ways to drive a stepper motor as shown below:

Wave drive:

In this drive method only a single phase is activated at a time. It is the fastest but is rarely used.

Full step drive:

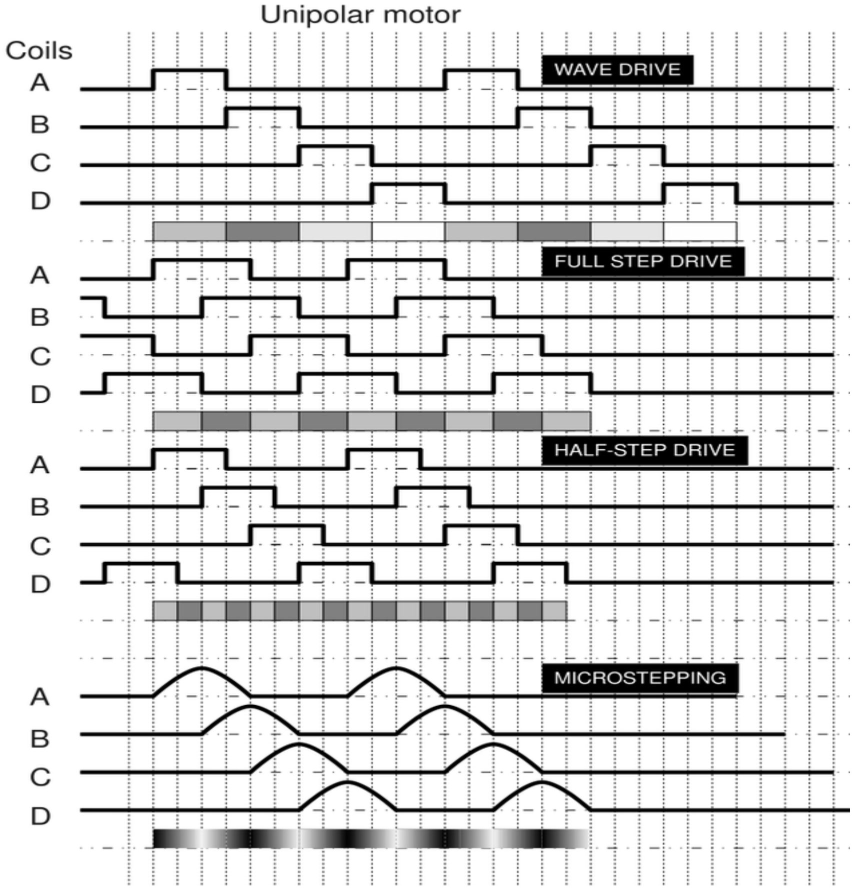
This is the usual method for full step driving the motor. Two phases are always on so the motor will provide its maximum rated torque.

Half stepping:

When half stepping, the drive alternates between two phases on and a single phase on. This increases the angular resolution, but the motor also has less torque.

Micro stepping:

Here the windings are driven with sinusoidal AC waveform. This requires different hardware and isn't used in this project.

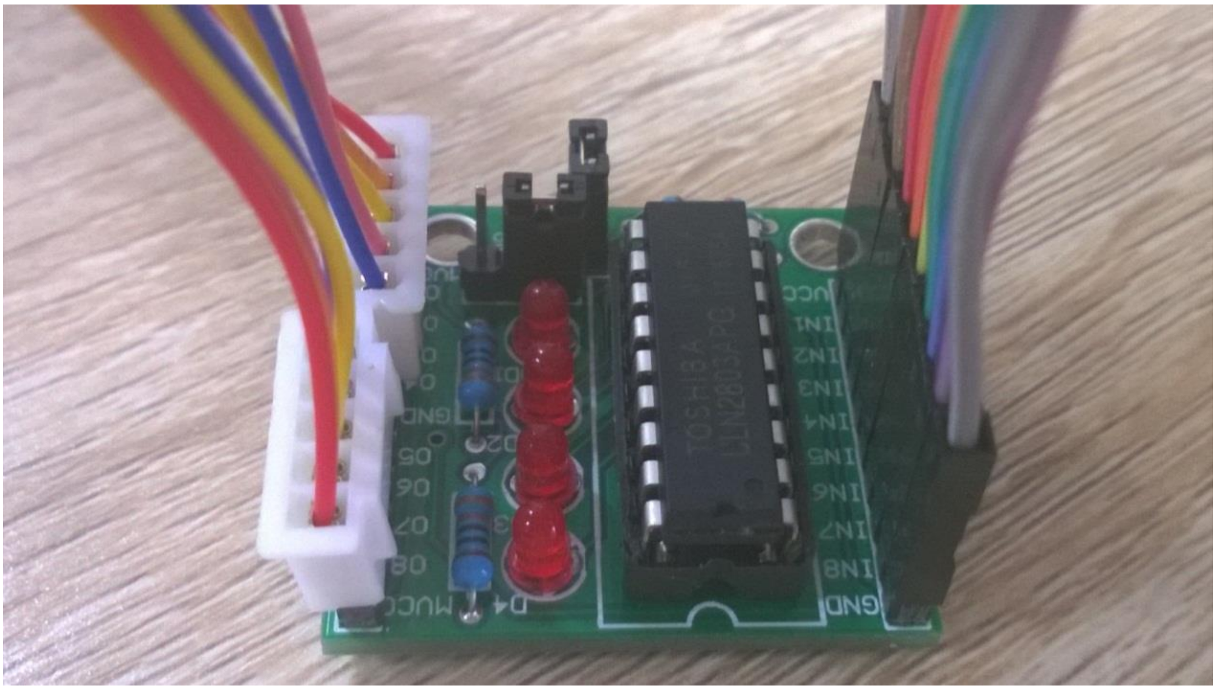


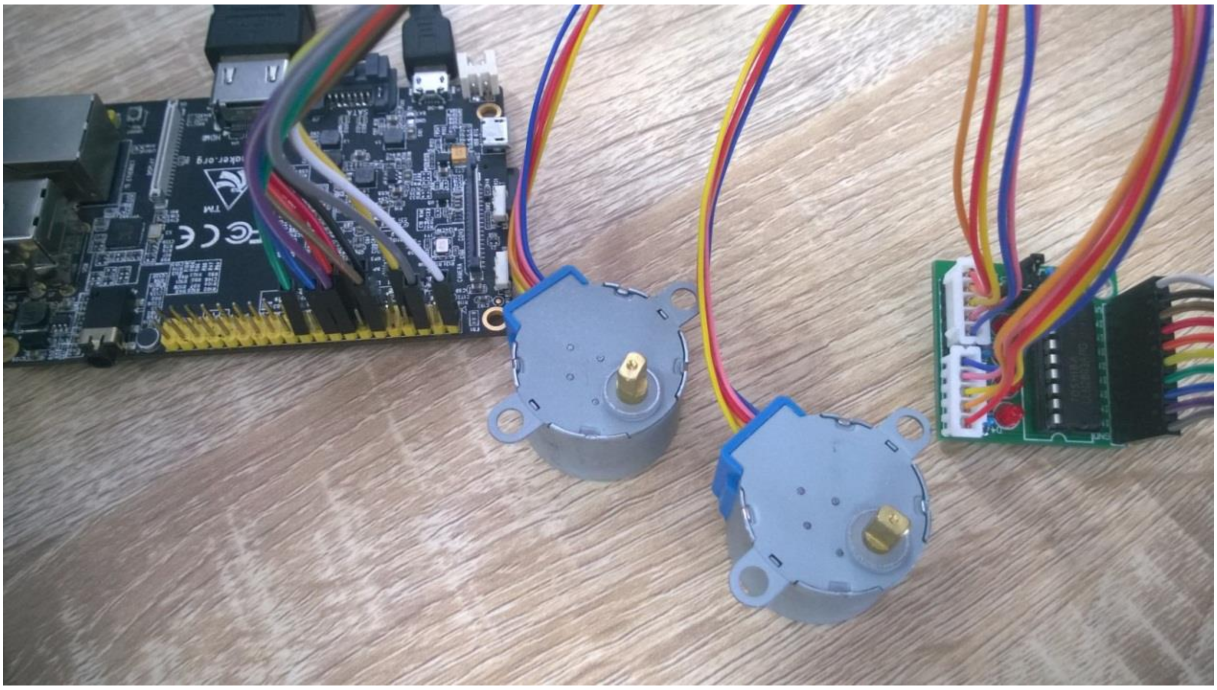
2.4.2 Example: Step Motor

Note that in this example, the LMK.BCM pin definition is applied (please refer to the Appendix 3.1 for details). Remember firstly to run LeScratch and then you start Scratch client; and, always remember to stop Scratch running blocks when you want to restart LeScratch handler.

By default, two step motor are connected by GPIO group Motor (17,18,27,22) and Motor(4,25,24,23). The index uses the LMK.GPIO BCM definition. The connection from motors to ULN2803A is simply red wires to VCC and then others are fixed. The input IN1~IN8 are connected to GPIO (17,18,27,22) and (4,25,24,23), or equally PIN number (11, 12, 13, 15) and (7, 22, 18, 16).

GND	IN1	IN2	IN3	IN4	IN5	IN6	IN7	IN8	VCC
PIN 6	PIN 23	PIN 26	PIN 36	PIN 37	PIN 7	PIN 22	PIN 18	PIN 16	PIN 2

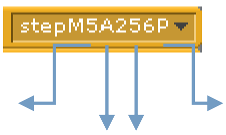




Command "stepM"+ " A/B" + "init"   for motor initialization

Command "stepM"+ "speed" + "A/B" + "steps" + "P/N" give speed, steps, direction

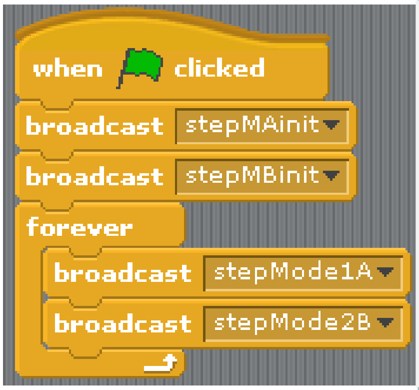
Examples: Gear ratio = 1/64; Step angle = 5.625 degrees /64（360/5.625 = 64）



Speed 5/1000   P: CLOCKWISE, N: ANTICLOCKWISE

MotorA   Steps

Click on the Green Flag to Run:
Init motor A (17,18,27,22) and
motor B (4,25,24,23)
While true
Turn motor A at speed 5/1000 for 256 steps, in the direction clockwise (Positive P)
Turn motor A at speed 5/1000 for 256 steps, in the direction anti-clockwise (Positive N)
Same for step motor B

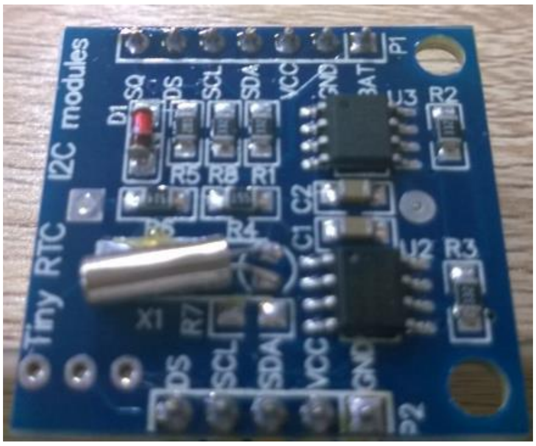


Click on the Green Flag to Run:
Init motor A (17,18,27,22) and
motor B (4,25,24,23)
While true
Set motor A working on Mode 1: Full Step Drive
Set motor B working on Mode 2: Wave Drive
Mode 1 speed 2.5, Mode 2 speed 2.5/3

2.5 RTC (Real Time Clock)

2.5.1 Technical Specification

DS1307 serial real-time clock (RTC) is a low power, full binary-coded decimal (BCD) clock/calendar. Address and data are transferred serially through an I2C, bidirectional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator.



DS1307 has a built-in power-sense circuit that detects power failures and can switches automatically to the backup supply. Timekeeping operation continues while the part operates from the backup supply.

2.5.2 Example: RTC

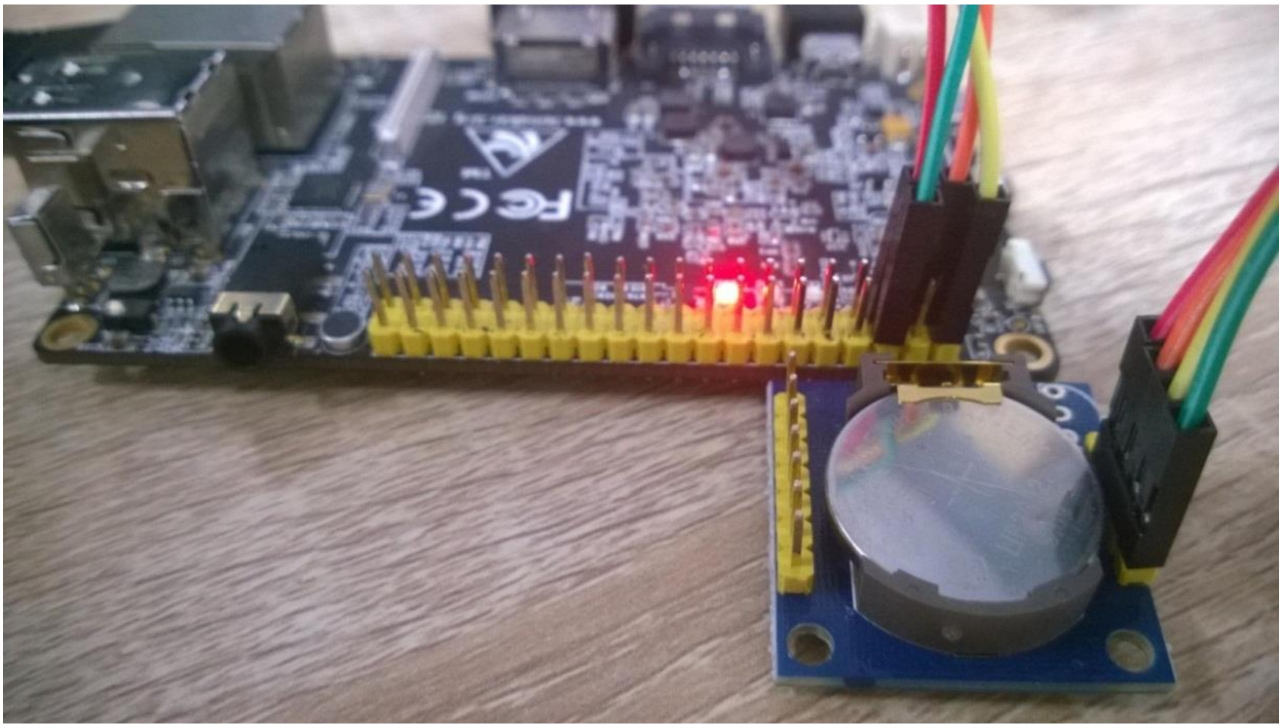
Note that in this example, the LMK.BCM pin definition is applied (please refer to the Appendix 3.1 for details). Remember firstly to run LeScratch and then you start Scratch client; and, always remember to stop Scratch running blocks when you want to restart LeScratch handler.

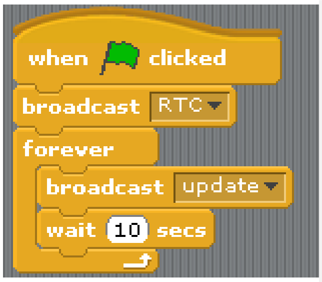
The RTC module is simply provide Scratch with the same exact date and time as the Banana Pro/LeMaker Guitar system: second, minute, hour, day, month, and year. This information can also be displayed on the right side if you choose the value from the Sensor list on the "Sensing" block on the left. RTC module use I2C connections to the Banana Pro/LeMaker Guitar, simply choose one of the I2C pins to connect:

VCC	SCLK	SDA	GND
PIN 4	PIN 5	PIN 3	PIN 6

Open the terminal and use i2c-tools to detect its address (should be 0x68):

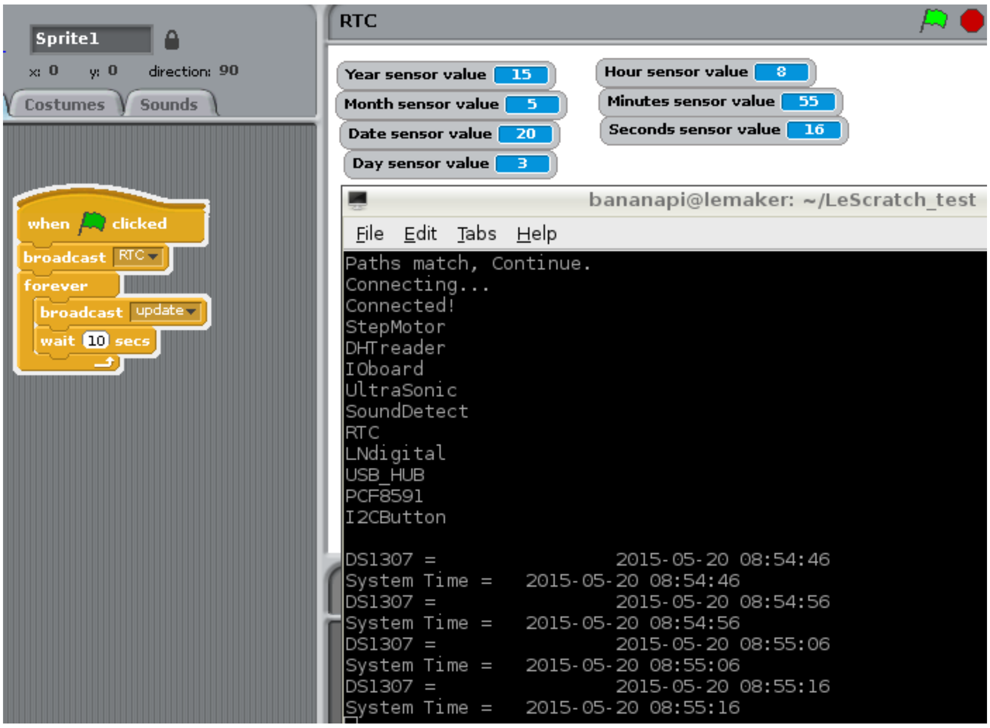
```
$ sudo i2cdetect -y 2
```





Click on the Green Flag to Run:
Declare to read from RTC module
While true
Update the Time and Date information
every 10 seconds (also display on the terminal)

The result is shown as the following:



2.6 UltraSonic Sensor

2.6.1 Technical Specification

Working voltage	5 VDC
Static current	<2 mA
Output signal	Electric frequency signal, high level 5V, low level 0V
Sensor angle	< 15 degrees
Detection distance	2cm-450cm
Precision	~2 mm
Trigger signal	10us TTL impulse
Echo signal	output TTL PWL signal

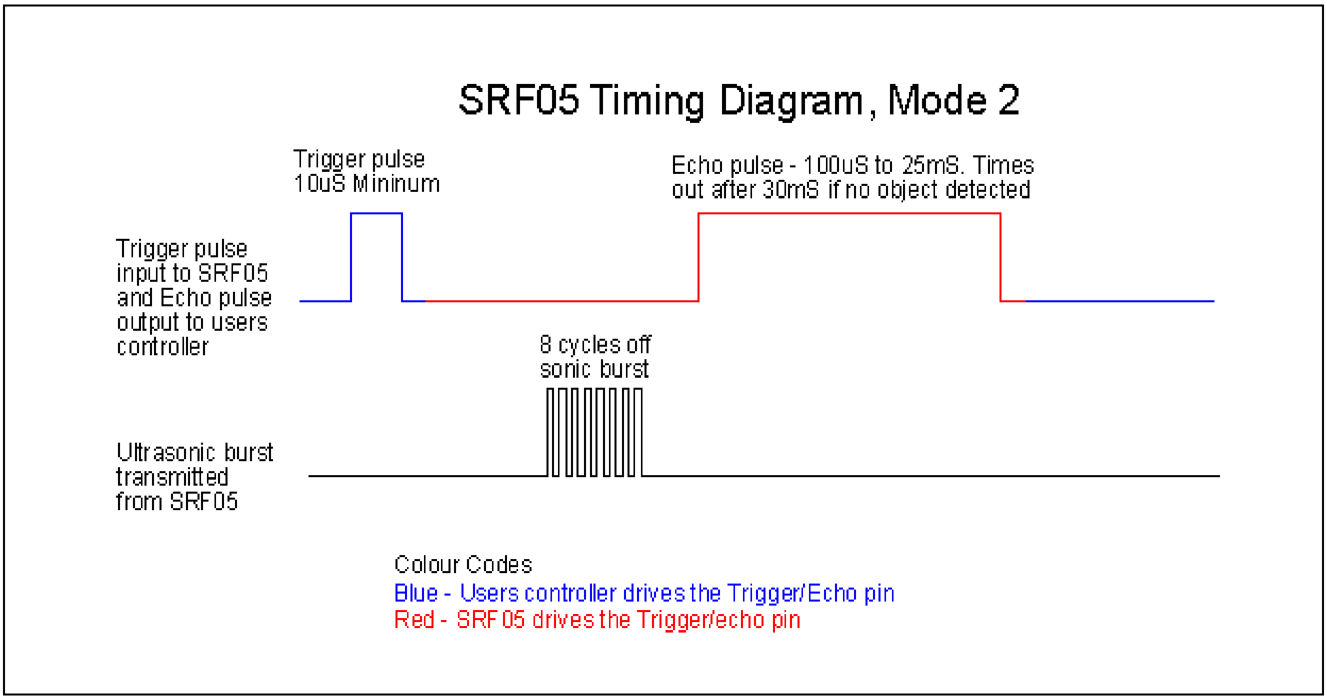
Pins	1. VCC 2. trig(T) 3. echo(R) 4. OUT 5. GND

In short an ultrasonic sensor works as it is described below:

a) Send a pulse signal to I/O TRIG which is at least 10us long, this will activate the module to start detecting;

b) The ultrasonic module will automatically send eight 40khz square waves, and will automatically detect when there is a reflect signal;

c) When there is an reflect signal back, the ECHO I/O will output a high level, the duration of the high-level signal is the time from untral sonic launch to return. As a result, the Measured distance = (T (Time of High Level output) * (340m/s)) / 2. The reason for the division by two is that since this is an echo it has traveled both to and from the object. Note the speed of sound is dependent of the temperature so keep that in mind if you need accuracy.

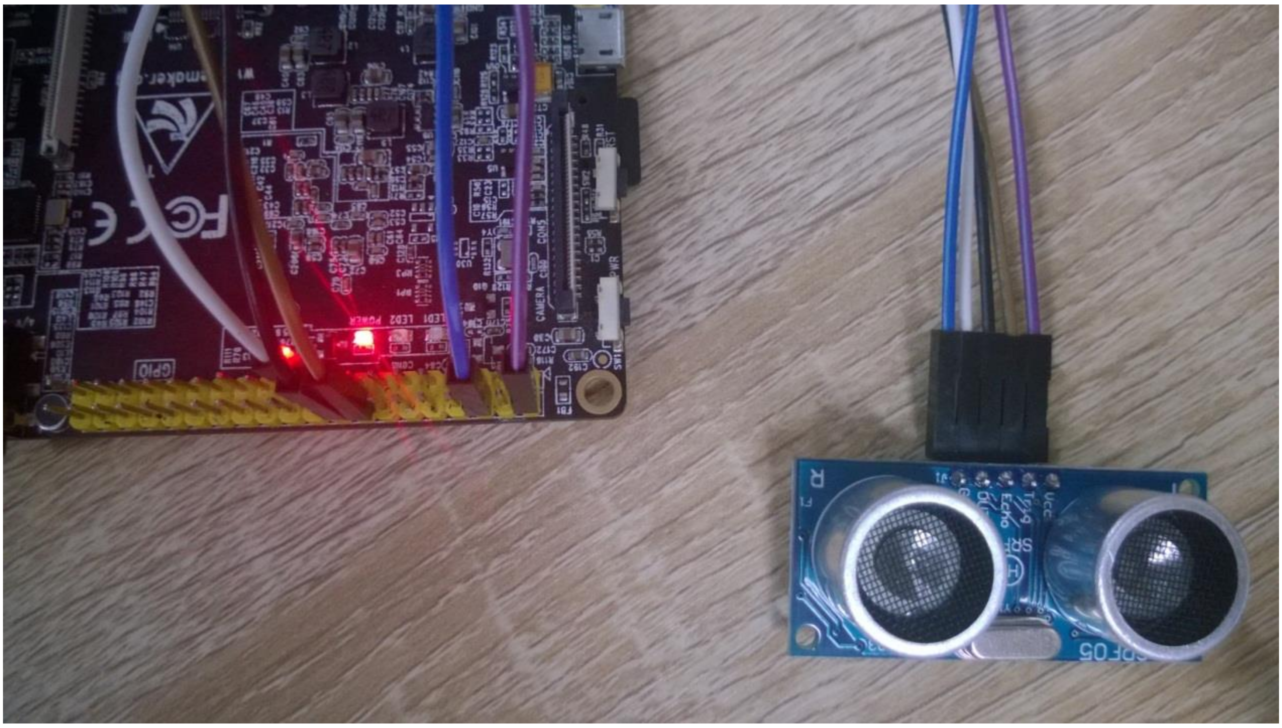


2.6.2 Example: UltraSonic Sensor

Note that in this example, the LMK.BCM pin definition is applied (please refer to the Appendix 3.1 for details). Remember firstly to run LeScratch and then you start Scratch client; and, always remember to stop Scratch running blocks when you want to restart LeScratch handler.

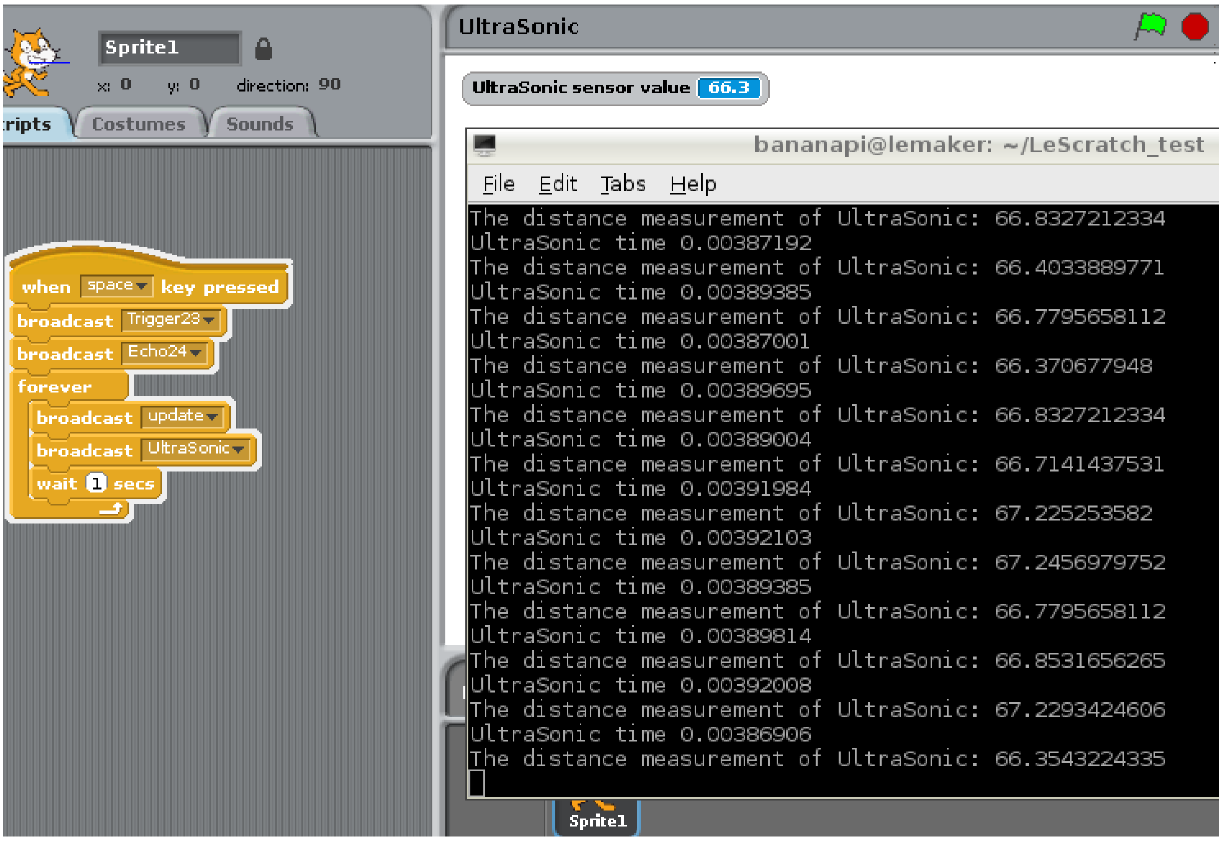
This module allows you to connect any digital pins to Trigger and Echo, the example is as following:

VCC	Trigger	Echo	GND
PIN 1	PIN 16	PIN 18	PIN 6



When keyboard space is pressed
Declare Trigger Pin on GPIO23
Declare Echo Pin on GPIO24
While true
Update the distance calculation by UltraSonic sensor per every second
(The frequency can be faster or slower as required)

The result is shown as the following:



2.7 Humidity and Temperature sensor

2.7.1 Technical Specification

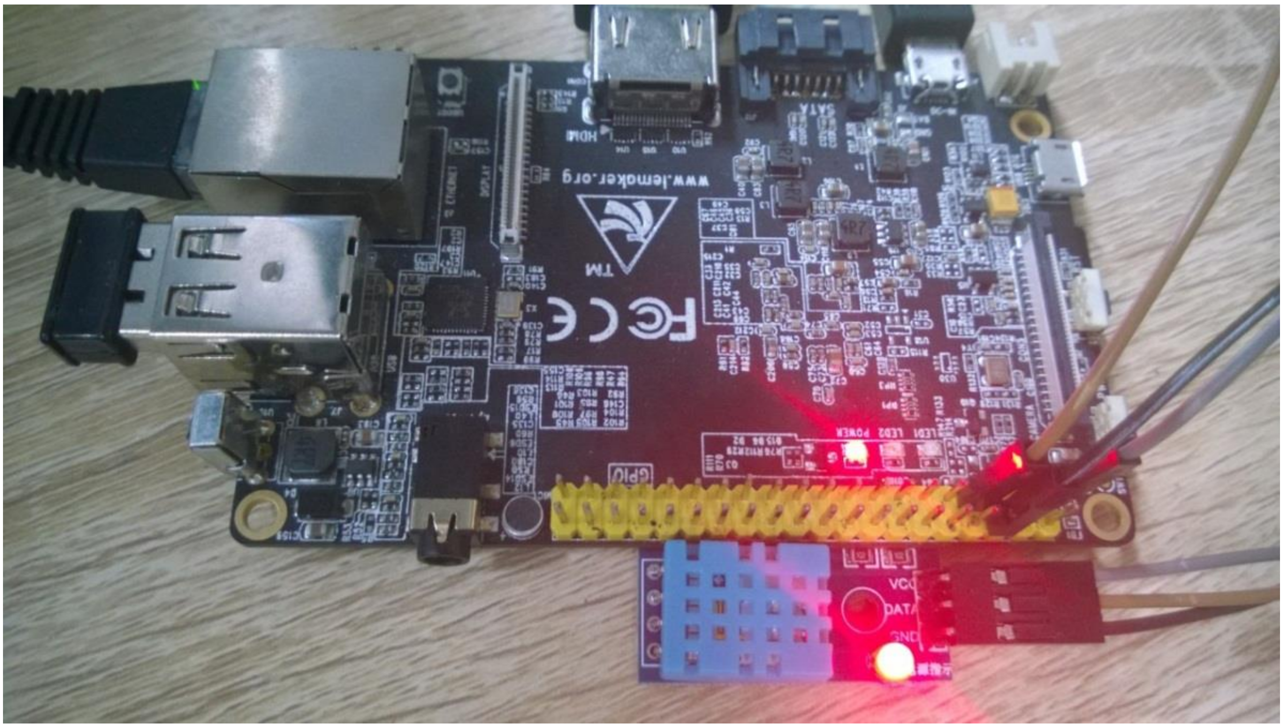
The DHT11 is a basic digital humidity and temperature sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed). It’s fairly simple to use, but requires very careful timing to read data:

1. 3 to 5V power and I/O
2. 2.5mA max current use during conversion (while requesting data)
3. Good for 20-90%RH humidity readings with ±5%RH accuracy
4. Good for 0 to 50 ℃ (122 ℉) temperature readings ±2 ℃ (±35.6 ℉) accuracy
5. No more than 1 Hz sampling rate (once every 1 seconds), data pin connect to GPIO

Currently DHT extension block supports DHT11, DHT22, AM2302, which uses WiringLMK API and pin definitions, which speed up the data transfer using 1-wire communication.

2.7.2 Example: DHT Sensor

Note that in this example, the wiringPiSetup( ) pin definition is applied (please refer to the Appendix 3.1 for details). Remember you have to firstly run LeScratch and then you start Scratch client; and, always remember to stop Scratch running blocks when you want to restart LeScratch handler.

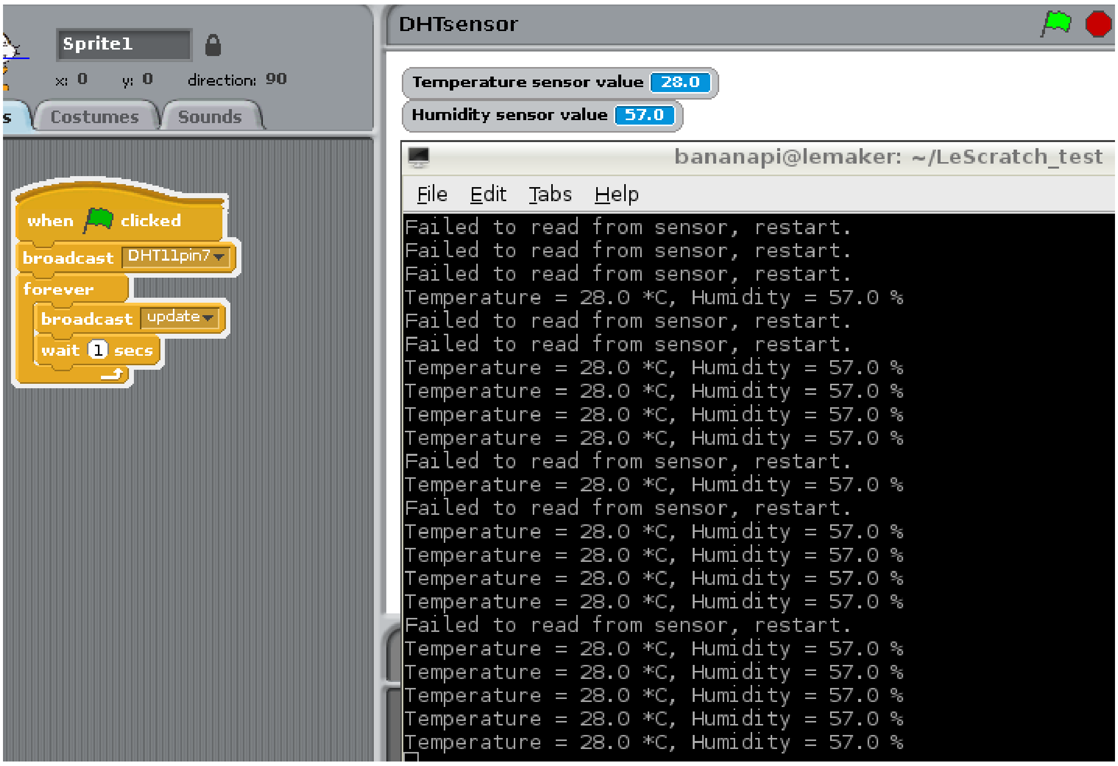


This module allows you to connect any digital pins to DATA pin, the example is as following:

VCC	DATA	GND
PIN 1	PIN 7	PIN 6

Define DHT sensor use PIN7 (wiringPiSetup( )),
DHT11 means the sensor type, now it supports DHT11, DHT12, AM2302
While true (constantly check and update DHT)
Update the Temperature and Humidity
every 2 seconds

The result is shown as the following: Note that sometimes the read from sensors may fail due to the fact that 1-wire communication verification goes wrong, but this does not affect its function and usage.

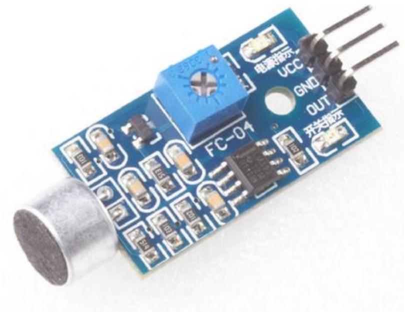


2.8 Sound Detect Sensor

2.8.1 Technical Specification

The sound detect sensor can be used to detect the sound intensity of the environment, but note that it can only identify the existence of sound according to the vibration principle, it can’t recognize specific loudness or sound frequency. It’s sensitive to sound intensity thus it can check if any sound happens around. The details are listed:

1. Working Voltage: 3.3V-5V
2. Output: digital switch output 0/low, 1/high, can be connected to GPIO
3. Adjustable sensitivity (Digital potentiometer adjustment: blue cube in the picture)
4. With fixed bolt hole, easy installation, PCB size: 3.2cm * 1.7cm



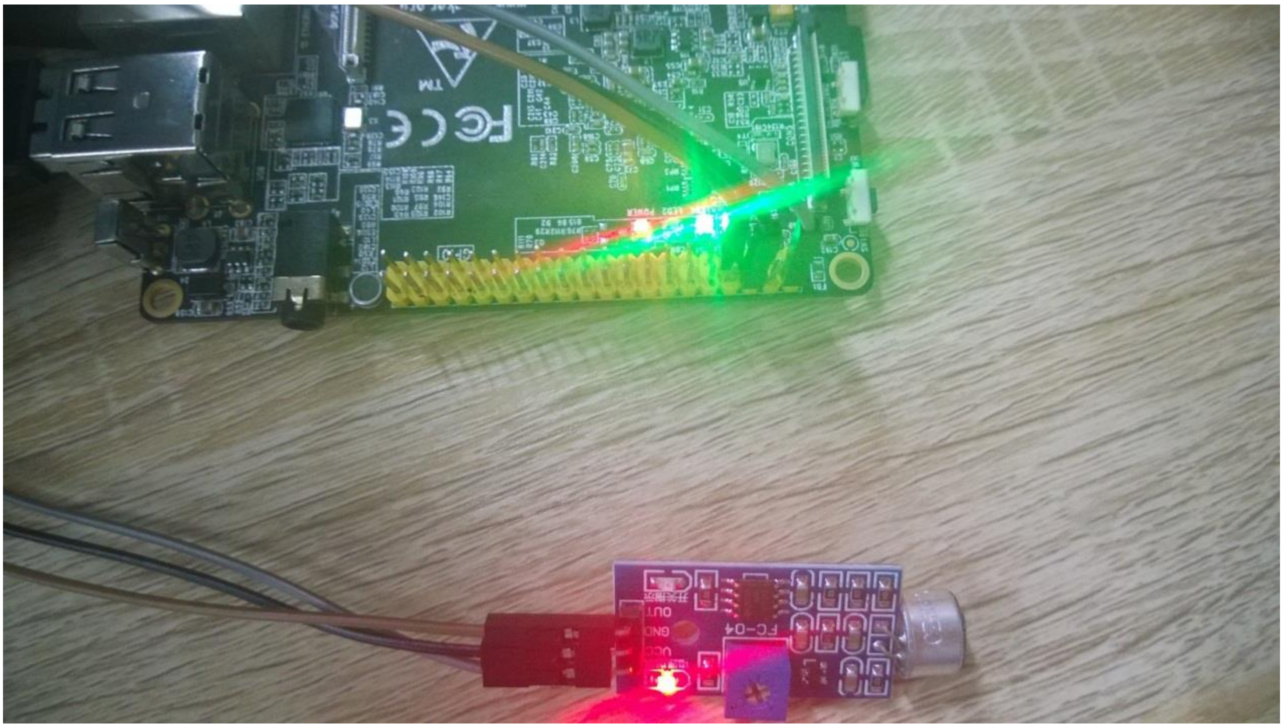
When the sound intensity from the environment is lower than the threshold given by the digital potentiometer adjustment, OUT pin is 1/high, if the intensity is bigger than the threshold, OUT pin is 0/low. By checking if the OUT pin is high/low, we'll know if there's sound. Digital OUT pin can drive the relay as a voice-activated switch.

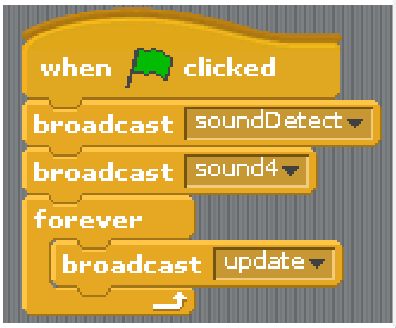
2.8.2 Example: Sound Detect Sensor

Note that in this example, the LMK.BCM pin definition is applied (please refer to the Appendix 3.1 for details). Remember firstly to run LeScratch and then you start Scratch client; and, always remember to stop Scratch running blocks when you want to restart LeScratch handler.

This module allows you to connect any digital pins to OUT signal, the example is as following:

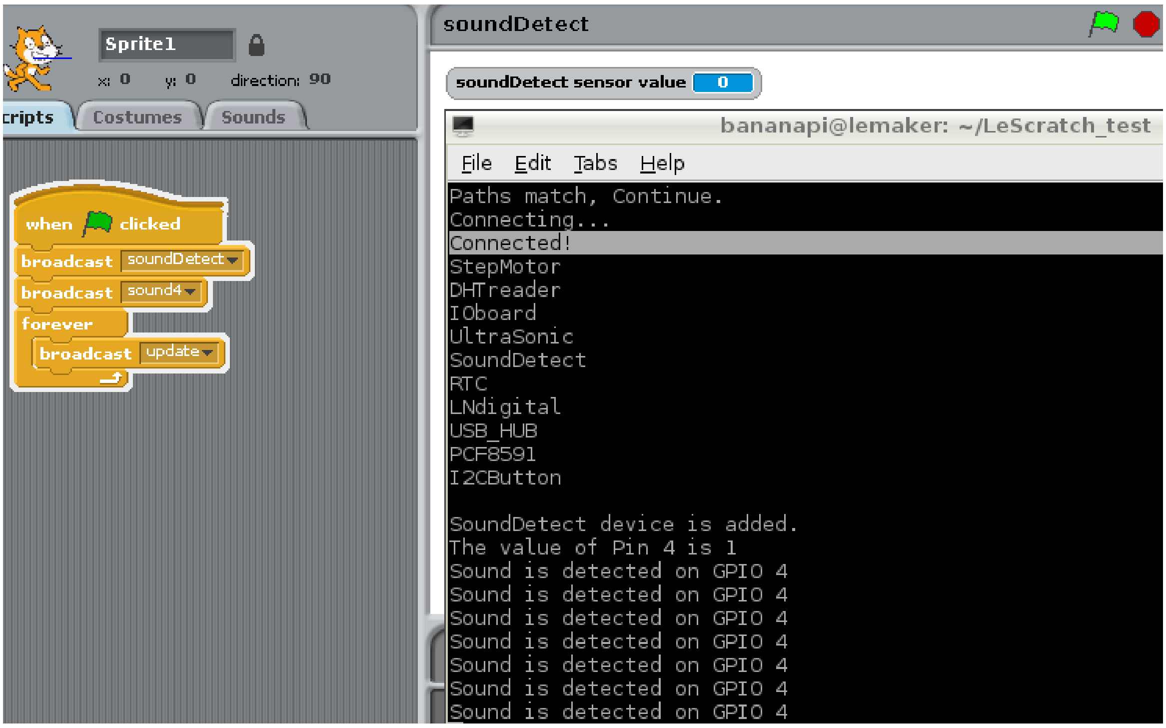
VCC	OUT	GND
PIN 1	PIN 7	PIN 6





Click on the Green Flag to Run:
Declare sound detect sensor
OUT pin connect to GPIO4
While True
Constantly check for updates

The result is shown as the following:



2.9 AD/DA Converter

2.9.1 Technical Specification

The PCF8591 is a single-chip, single-supply low power 8-bit CMOS data acquisition device with four analog inputs, one analog output and a serial I2C-bus interface. Three address pins A0, A1 and A2 are used for programming the hardware address, allowing the use of up to eight devices connected to the I2C-bus without additional hardware. Address, control and data to and from the device are transferred serially via the two-line bidirectional I2C-bus. PCF8591 is featured as following:

a) Single power supply; Operating supply voltage 2.5 V to 6 V; Low standby current

b) Serial input/output via I2C-bus

c) Address by 3 hardware address pins

d) Sampling rate given by I2C-bus speed

e) 4 analog inputs programmable as single or differential inputs: AIN0~AIN3

f) Analog voltage range from VSS to VDD

g) On-chip track and hold circuit

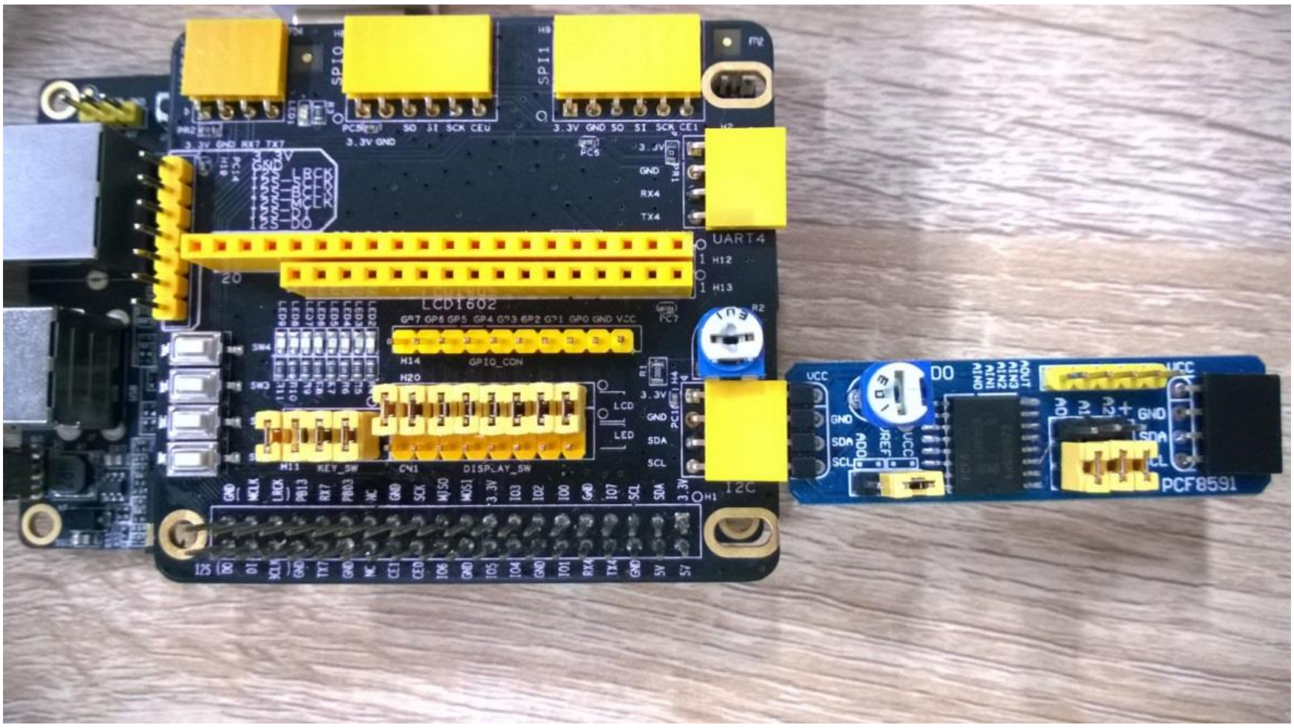
h) 8-bit successive approximation A/D conversion

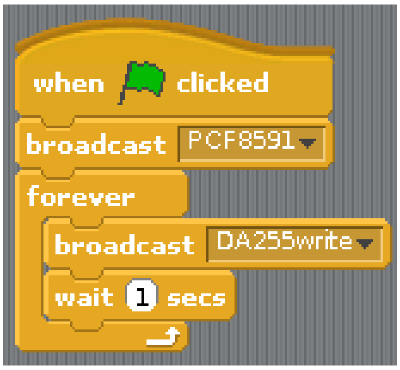
i) Multiplying DAC with one analog output

2.9.2 Example: AD/DA Convertor

Note that in this example, the wiringPiSetup( ) pin definition is applied (please refer to the Appendix 3.1 for details). Remember you have to firstly run LeScratch and then you start Scratch client; and, always remember to stop Scratch running blocks when you want to restart LeScratch handler.

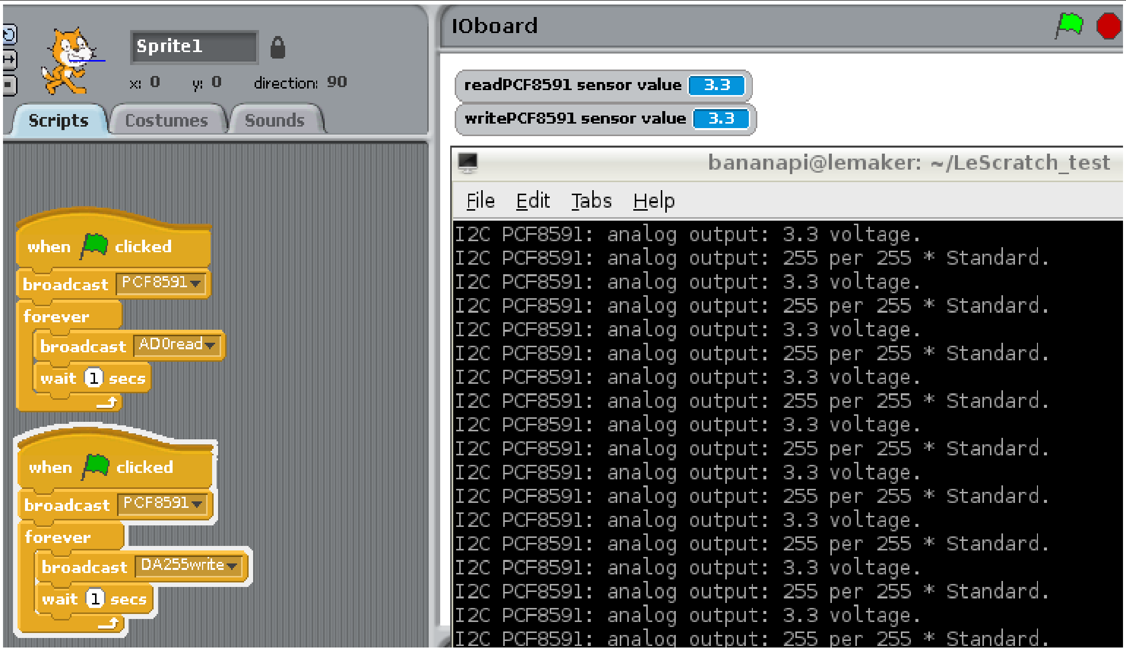
This module simply uses IO board to connect (there're both AD and DA examples):

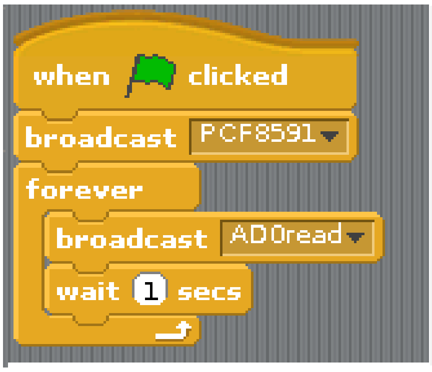




Click on the Green Flag to Run:
Declare PCF8591 AD/DA Convertor
By default the extension address A2A1A0 = 000
While True
DA255 Digital to Analog, given digital num, the analog output equals to 255/255 *3.3V
Analog Output Formula: num/255 *3.3 V

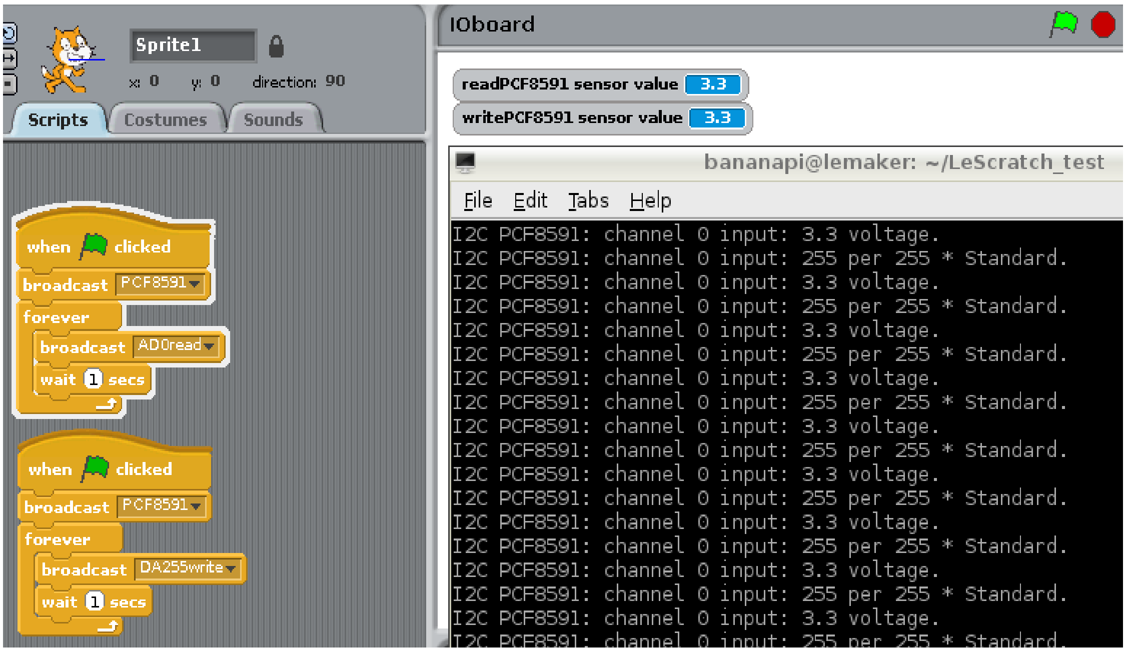
In this DA conversion above, given digital value 255 (DA255write, range 0~255), it is converted by DA calculation and then output the analog value from AOUT pin; use a multimeter to measure, range 0 ~3.3V (here it reads voltage 3.3V). The result is shown as the following:





Click on the Green Flag to Run:
Declare PCF8591 AD/DA convertor
By default the extension address A2A1A0 = 000
While True
AD0 reads analog channel 0 (connects to VCC 3.3V)
Analog input is converted into digital value
Analog Input Formula: num/255 *3.3 V

In this AD conversion above, we connect analog channel 0 connects to VCC 3.3V, after AD calculation it will be converted into digital values 255 (digital range 0~255). The result is shown as the following:



2.10 Photoresistor

2.10.1 Technical Specification

Photoresistor module is sensitive to ambient light intensity, generally used to detect ambient brightness and intensity. Under the conditions that this module light intensity does not reach the threshold, the digital output DO port outputs high level, when the ambient light intensity exceeds a set threshold, the module output a low DO; DO can be directly connected to the control board, by detecting high/low of the DO pin, we measure the changes of light intensity in the environment; If connect the analog output AO to the AD modules and conduct the AD conversion, you can get a more accurate ambient light intensity value. The sensor sensitivity can be adjustable (through digital potentiometer adjustment), the following parameters are listed:

1. Working voltage 3.3V-5V
2. The power indicator (red led) and digital switching output indicator (green led)
3. The digital potentiometer adjustment for sensor sensitivity (blue cube on the board)
4. Output: (a) analog voltage output AO, connect to PCF8591 analog input channel 0 AIN0

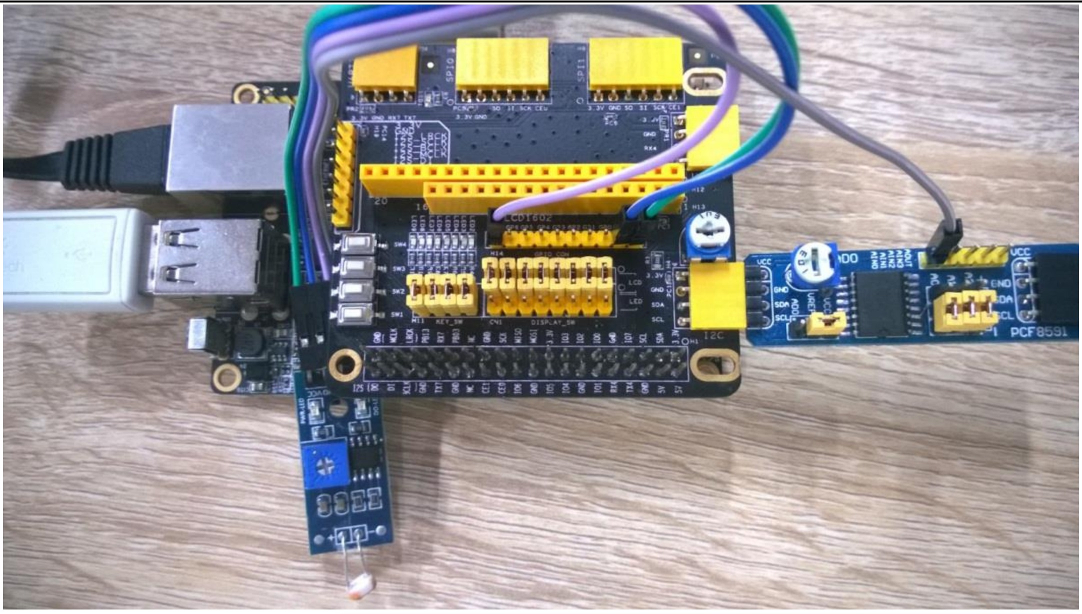
   (b) digital switching output DO (0 and 1), connect to GPIO such as GPIO4

2.10.2 Example: Photoresistor

Note that in this example, the LMK.BCM pin definition is applied (please refer to the Appendix 3.1 for details). Remember firstly to run LeScratch and then you start Scratch client; and, always remember to stop Scratch running blocks when you want to restart LeScratch handler.

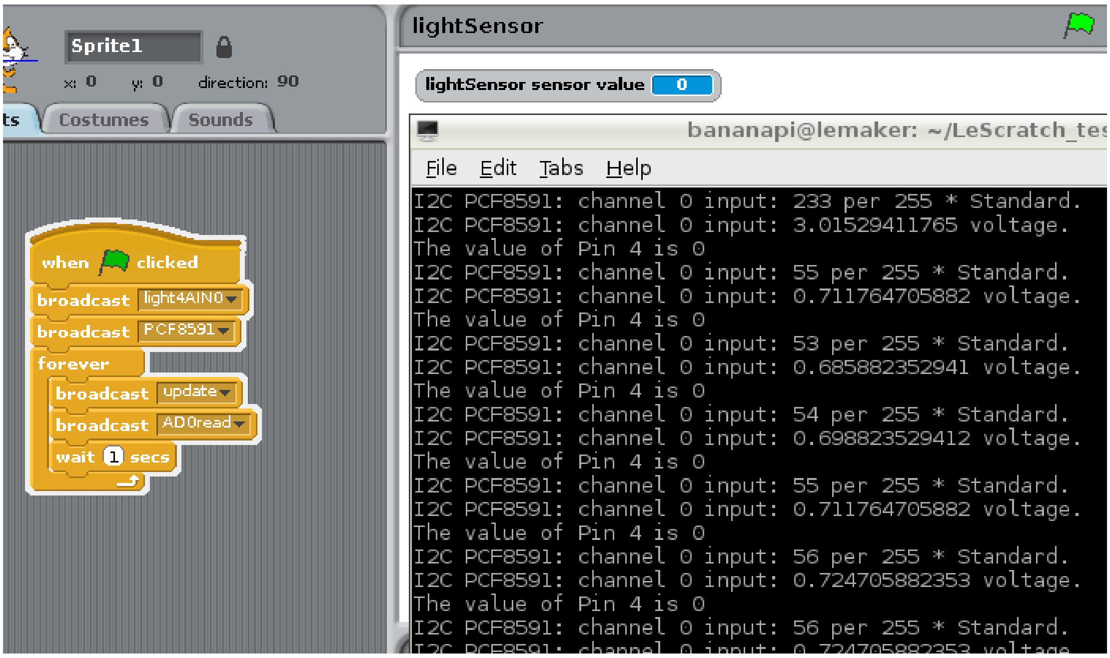
Here we use a 4Pins photoresistors while you can also use the others such as the 3Pins photoresistors. This module allows you to define your own DO pin; AO signal is connected to AIN0 AD/DA module for measurement, the example is as following:

VCC	DO	AO	GND
PIN 1	PIN 7	AIN0	PIN 6



Click on the Green Flag to Run:
Declare Photoresistor module
Declare PCF8591 to read analog value (sensor is light intensity sensitive)
While True:
Constantly check for update per second:
AIN0 read light intensity from channel 0
DO gives digital indicator output

The result is shown as the following (when the light intensity is low or equally saying the environment is dark, the voltage it generate is bigger, totally dark will generate 3.3V or equally digital value 255; if the environment is bright, the voltage is smaller, totally brightness equals to 0):



2.11 Touch Sensor

2.11.1 Technical Specification

The touch sensor is based on a touch-sensing IC (TTP223B) low-power capacitive touch switch module. Under normal touch conditions, the signal SIS pin outputs 0/low, the led indicator lights off; when the corresponding sensing capacitor position is touched, the signal SIS outputs 1/high and the led indicator lights. If within 12 seconds no touch is detected, then touch sensor will switch to low-power consumption. The parameters are as follows:

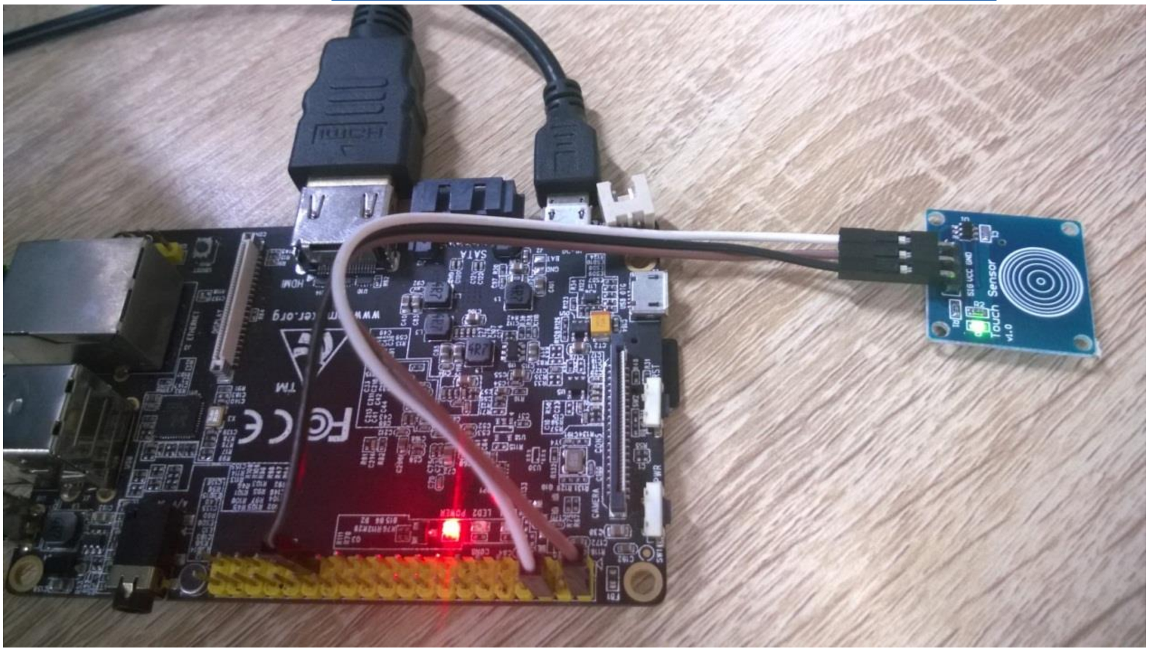
1. Connections: VCC (power supply range: 3-5V, current: 5mA),
   GND (ground), SIS (touch signals)
2. Response time: 220ms at low-power state, 60ms at touch state

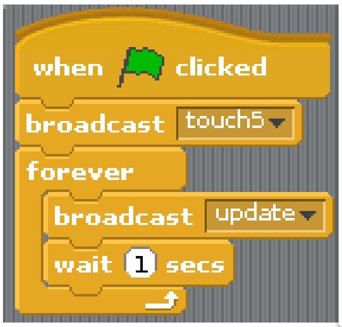
2.11.2 Example: Touch Sensor

Note that in this example, the LMK.BCM pin definition is applied (please refer to the Appendix 3.1 for details). Remember firstly to run LeScratch and then you start Scratch client; and, always remember to stop Scratch running blocks when you want to restart LeScratch handler.

This module allows you to connect any digital pins to SIS pin, the example is as following:

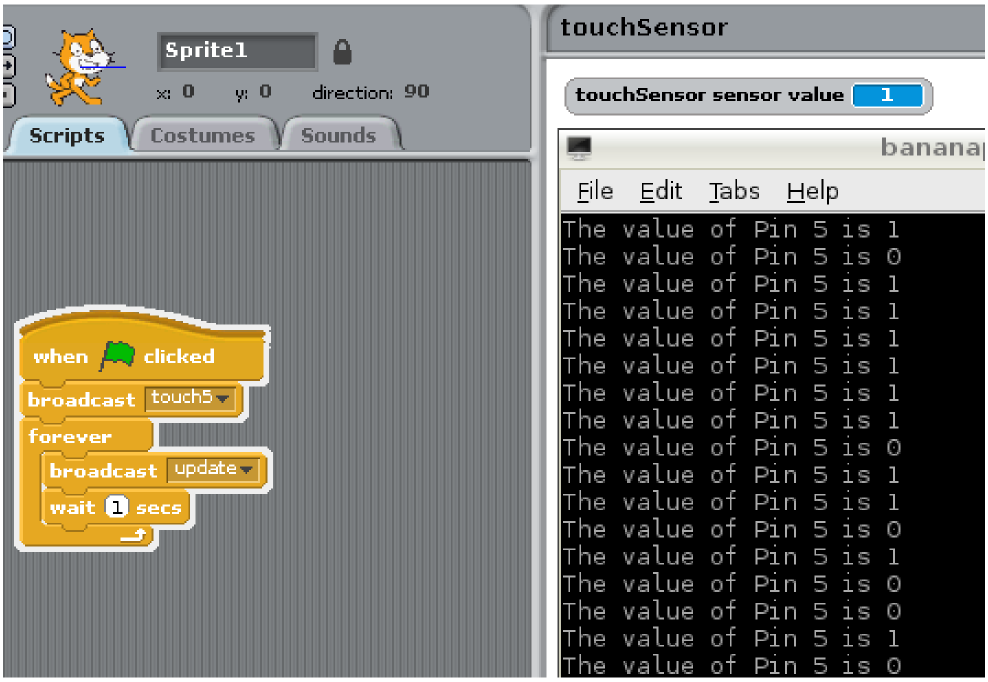
VCC	SIS	GND
PIN 1	PIN 29	PIN 6





Click on the Green Flag to Run:
Declare touch sensor, SIS connect to GPIO5
While True:
Constantly check for update per second:
SIS digital signal output

The result is shown as the following:



2.12 Tilt Sensor

2.12.1 Technical Specification

Tilt sensor can sense the change of angles of objects: level the module on the desktop and rotates this slowly in one direction, you will see the led indicator lights up; then rotates it in the opposite direction, (back to the original state), the led indicator lights off. Changing the object angle will trigger a digital switching signal (0/1), so that the led indicator will light on or off. By monitoring the tilt sensor digital signal DO (low/high), the tilt status of objects can be informed. The parameters are as follows:

a) Working voltage of 5V,

b) Power led (red) and digital switching output led indicator (green)

c) Lifetime: at room temperature and under normal use, switching life of up to 100,000 times (/1sec).

d) Function: sense the change of angle of objects (precision up to 15-45 degree)

e) Adjustable sensitivity (Digital potentiometer adjustment: blue cube on the board)

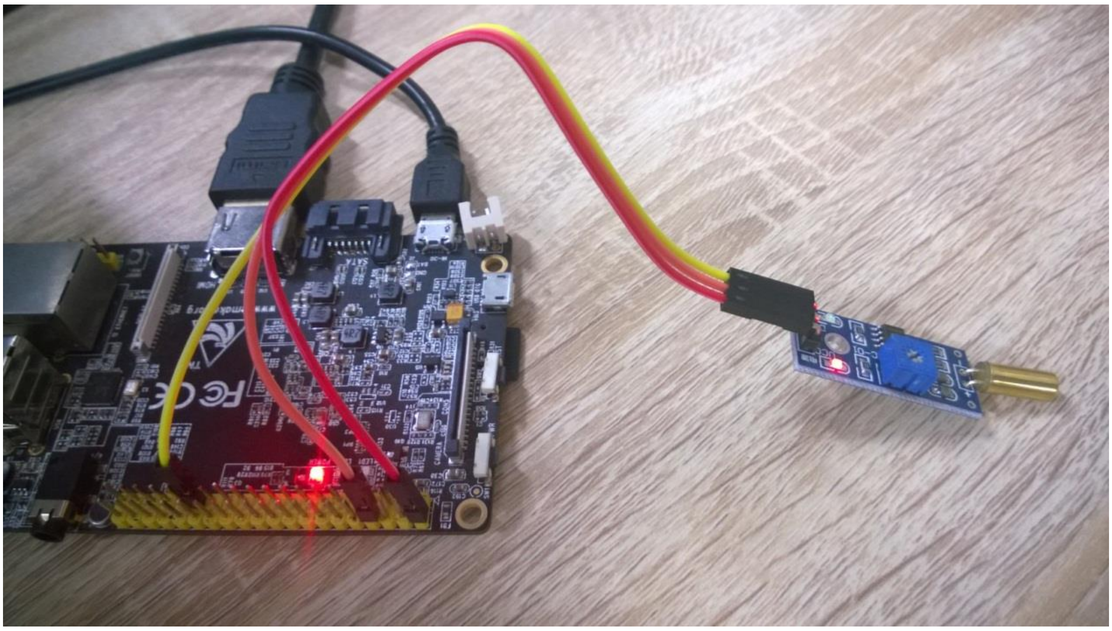
f) Output: digital switching outputs DO (0 and 1)

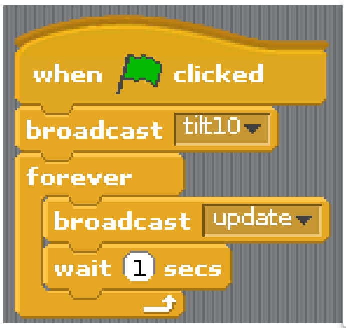
2.12.2 Example: Tilt Sensor

Note that in this example, the LMK.BCM pin definition is applied (please refer to the Appendix 3.1 for details). Remember firstly to run LeScratch and then you start Scratch client; and, always remember to stop Scratch running blocks when you want to restart LeScratch handler.

Note that tilt sensor can easily go wrong (such as the metal ball inside is stuck, thus it can't reach the right point, both DO signal and the LED indicator won't work), This module allows to connect DO pin to one of which in this list [4,14,15,17,18,27,22,23,24,10,9,25,11,8,7] , the example is as below:

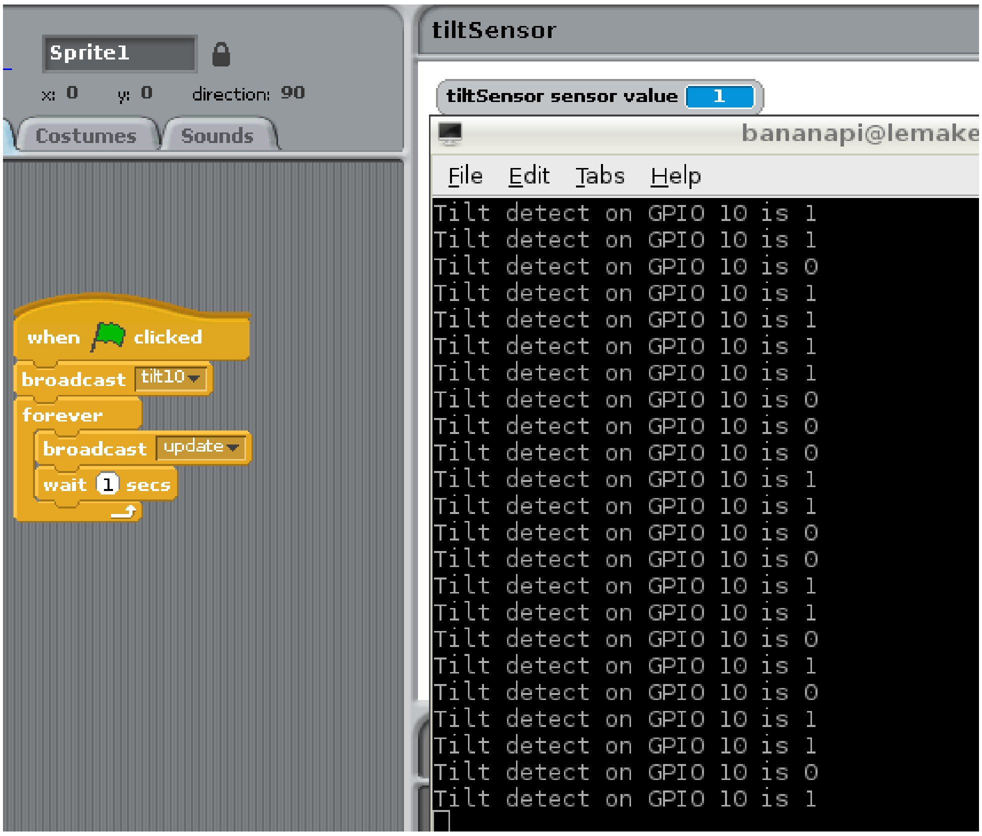
VCC	DO	GND
PIN 1	PIN 19	PIN 6





Click on the Green Flag to Run:
Declare tilt sensor module
While True:
Constantly check for update per second:
DO digital signal output

The result is shown as the following:



2.13 LCD1602 Display

2.13.1 Technical Specification

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. LCDs are economical; easily programmable; have no limitation of displaying special and even custom characters.

The LCD has two registers, namely, Command and Data. The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.

It's featured as following:

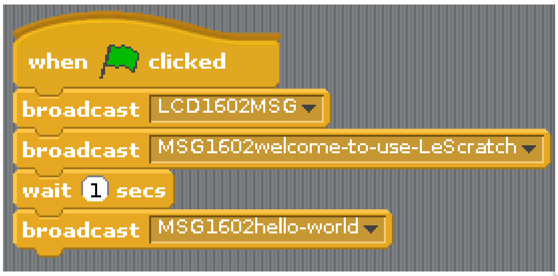
1. Display Format: 16 Character x 2 Line
2. Input Data: 4-Bits or 8-Bits interface available
3. Display Font: 5 x 8 Dots
4. Power Supply: Single Power Supply (5V±10%)

No.	Symbol	Level	Function	
1	Vss	--	0V	Power Supply
2	Vdd	--	+5V	
3	V0	--	for LCD	
4	RS	H/L	Register Select: H:Data Input L:Instruction Input	
5	R/W	H/L	H--Read L--Write	
6	E	H.H-L	Enable Signal	
7	DB0	H/L	Data bus used in 8 bit transfer	
8	DB1	H/L		
9	DB2	H/L		
10	DB3	H/L		
11	DB4	H/L	Data bus for both 4 and 8 bit transfer	
12	DB5	H/L		
13	DB6	H/L		
14	DB7	H/L		
15	BLA	--	BLACKLIGHT +5V	
16	BLK	--	BLACKLIGHT 0V-	

2.13.2 Example: LCD1602 Display

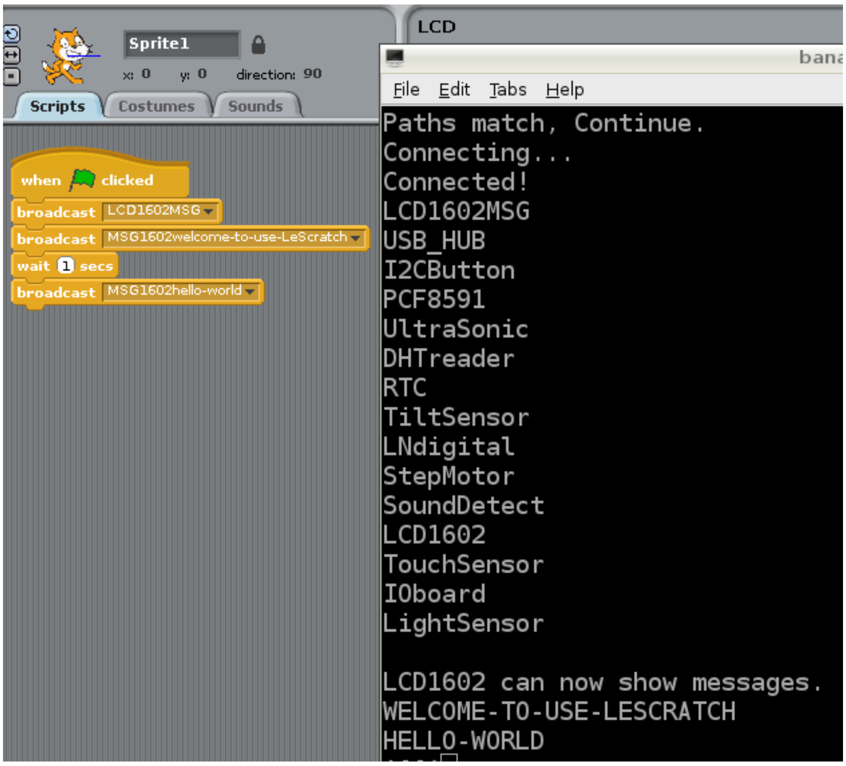
Note that in this example, the LMK.BCM pin definition is applied (please refer to the Appendix 3.1 for details). Remember you have to firstly run LeScratch and then you start Scratch client; and, always remember to stop Scratch running blocks when you want to restart LeScratch handler.

This module simply uses IO board to connect using the LCD1602 pins, the example is as following:



Click on the Green Flag to Run:
Declare LCD1602 to display message
Broadcast message: welcome-to-use-LeScratch
Wait for 1 second
Broadcast message: hello-world

The result is shown as the following, note that the last message will replace the previous one on the LCD screen:



3 Appendix

3.1 PIN MAPPING Comparison

Banana Pro/LeMaker Guitar			
Pin on Board	LMK.BCM	wiringPiSetup( )	Functions
CON6-01			3.3V
CON6-02			5V
CON6-03	GPIO2	PIN8	**TWI2_SDA**
CON6-04			5V
CON6-05	GPIO3	PIN9	**TWI2_SCK**
CON6-06			GND
CON6-07	GPIO4	PIN7	LCD1_D02/UART3_RTS/EINT2
CON6-08	GPIO14	PIN15	LCD1_D04/**UART4_TX**/EINT4
CON6-09			GND
CON6-10	GPIO15	PIN16	LCD1_D05/**UART4_RX**/EINT5
CON6-11	GPIO17	PIN0	SPI1_MISO/**UART2_RX**/EINT31
CON6-12	GPIO18	PIN1	**PWM1**/TWI4_SDA
CON6-13	GPIO27	PIN2	SPI1_MOSI/**UART2_TX**/EINT30
CON6-14			GND
CON6-15	GPIO22	PIN3	SPI1_CLK/**UART2_CTS**/EINT29
CON6-16	GPIO23	PIN4	LCD1_D20/ERXDV/**CAN_TX**/EINT20/CSI1_D20
CON6-17			3.3V
CON6-18	GPIO24	PIN5	LCD1_D21/EMDC/**CAN_RX**/EINT21/CSI1_D21
CON6-19	GPIO10	PIN12	**SPI0_MOSI**/UART6_TX/CLK_OUT_A/EINT24
CON6-20			GND
CON6-21	GPIO9	PIN13	**SPI0_MISO**/UART6_RX/CLK_OUT_B/EINT25
CON6-22	GPIO25	PIN6	SPI1_CS0/**UART2_RTS**/EINT28
CON6-23	GPIO11	PIN14	**SPI0_CLK**/UART5_RX/EINT23

CON6-24	GPIO8	PIN10	**SPI0_CS0**/UART5_TX/EINT22
CON6-25			GND
CON6-26	GPIO7	PIN11	**SPI0_CS1**/PS2_SCK1/TCLKIN0/EINT26/TCLKIN0
CON6-27	GPIO0	PIN30	**TWI3_SDA**
CON6-28	GPIO1	PIN31	**TWI3_SCK**
CON6-29	GPIO5	PIN21	IR0_TX/SPDIF_MCLK/STANBYWFI
CON6-30			GND
CON6-31	GPIO6	PIN22	PS2_SDA0/**UART7_RX**/HSDA
CON6-32	GPIO12	PIN26	PS2_SCK0/**UART7_TX**/HSCL
CON6-33	GPIO13	PIN23	SPI2_CS1/SPDIF_DO
CON6-34			GND
CON6-35	GPIO19	PIN24	**I2S0_LRCK**/AC97_SYNC
CON6-36	GPIO16	PIN27	**I2S0_BCLK**/AC97_BCLK
CON6-37	GPIO26	PIN25	**I2S0_MCLK**/AC97_MCLK
CON6-38	GPIO20	PIN28	**I2S_DI**/AC97_DI/SPDIF_DI
CON6-39			GND
CON6-40	GPIO21	PIN29	**I2S0_DO0**/AC97_DO

3.2 Components List

章节	器件名称	购买链接
2.4 Step Motor	ULN2803 Driver	Shopping Link
2.4 Step Motor	5V Step Motor	Shopping Link
2.5 RTC (real time clock)	RTC	Shopping Link
2.6 Ultrasonic Sensor	Ultrasonic Sensor	Shopping Link
2.7 Humidity&Temperature Sensor	DHT Sensor	Shopping Link
2.8 Sound Detect Sensor	Sound Detect Sensor	Shopping Link
2.9 AD/DA Converter	AD/DA Converter	Shopping Link
2.10 Photoresistor	Photoresistor	Shopping Link
2.11 Touch Sensor	Touch Sensor	Shopping Link
2.12 Tilt Sensor	Tilt Sensor	Shopping Link
2.13 LCD1602 Display	LCD1602	Shopping Link

Disclaimer: In order to avoid beginners buying sensors with wrong parameters, this manual provides a list of required sensors for reference only; LeMaker official site does not sell these parts.