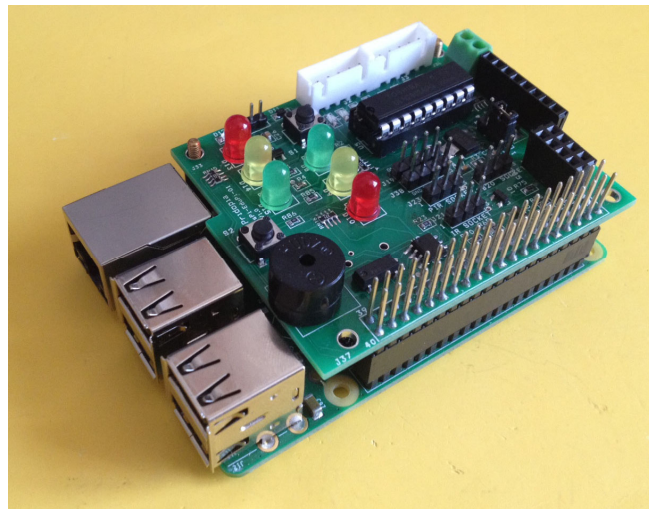
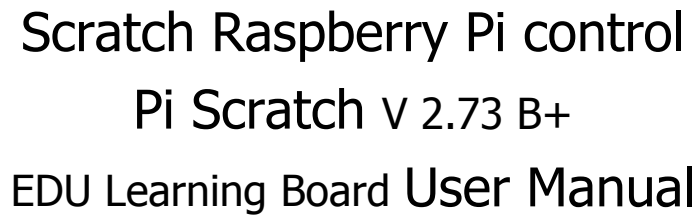




Tony Chang

18,Dec,2014

Raspberry Pi is a trademark of the Raspberry Pi Foundation

Contents

1. Install EDU Learning Board	2
2. Download & Install Pi_Scratch software	3
Setting Scratch for Auto loading Pi_Scratch	4
3. Enable i2c & Install i2c	5
4. Other software pack install	7
5. Basic GPIO Function	13
1. BASIC GPIO input & output demo	15
2. BASIC GPIO step motor demo	17
6. EDU Learning Board support	18
1. RTC setting	20
2. Stepper Motor	24
3. 1-Wire DS18B20 Temperature Sensor	27
4. DC 5V Mini Fan	29
5. IR PIR Motion Sensor Module	31
6. IR remote control	34
7. RFID reader	40
8. NRF24L01 Radio Frequency Reader module	45
7. Raspberry Pi B+ GPIO support	49
8. Python Test Code	50

Q & A

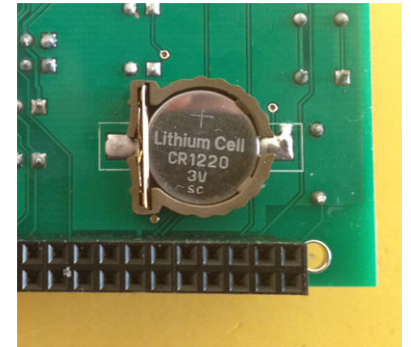
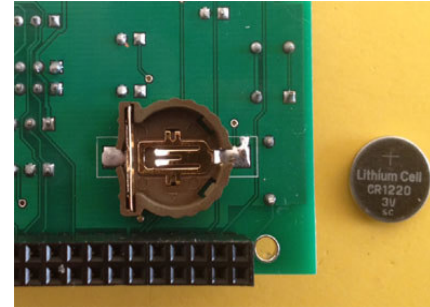
Pi_Scratch Install Q & A	52
How to Auto run Scratch & auto load Pi_Scratch example file	56
How to make Image from three ZIP file	58
First thing need do for Pi_Scratch ready Image	59
How to upgrade from old Pi_Scratch to new Ver Pi-Scratch	60
VNC (VIRTUAL NETWORK COMPUTING) setting	61
Setting Scratch to active Mesh function	64

EDU Learning Board GPIO List

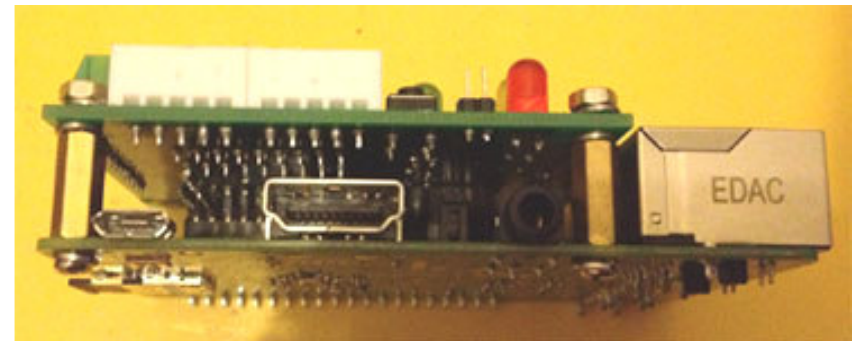
	StepMotor1	StepMotor2	DS18B20 Temp Sensor	IR remote Sensor	PIR motion sensor	Buzzer
GPIO17	v					
GPIO18	v					
GPIO27	v					
GPIO22	v					
GPIO23		v		v	v	
GPIO24		v				v
GPIO25		v				
GPIO4		v	v			
	Green LED 1	Yellow LED 1	Red LED 1	Green LED 2	Yellow LED 2	Red LED 2
GPIO12	v					
GPIO13		v				
GPIO19			v			
GPIO16				v		
GPIO21					v	
GPIO26						v
SPI signal	RFID	NRF24L01		Micro SW 1	Micro SW 2	Mini Fan
GPIO7(CE1)				GPIO20	v	
GPIO8(CE0)	v	v		GPIO6	v	
GPIO9(MISO)	v	v		GPIO5		v
GPIO10(MOSI)	v	v				
GPIO11(SCLK)	v	v		i2c signal	RTC	
GPIO25		v		GPIO2(SDA)	v	
				GPIO3(SCL)	v	

Install EDU Learning Board

(1) Install RTC Battery



(2) Install PCB spacer

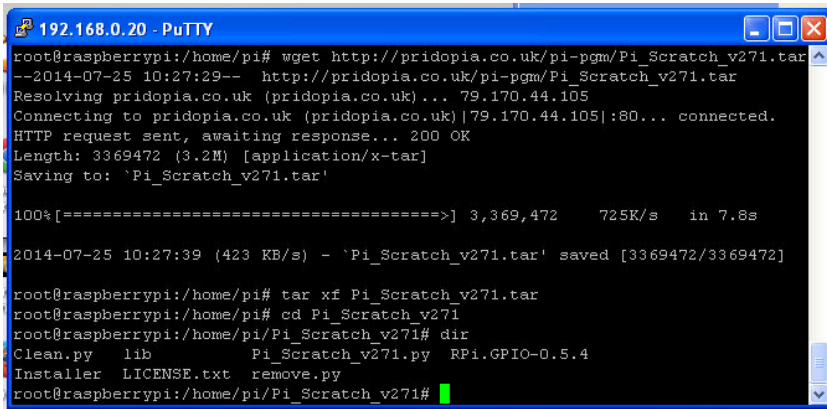


Download program pack from our web site

sudo wget http://pridopia.co.uk/pi-pgm/Pi_Scratch_v273.tar

tar xf Pi_Scratch_v273.tar

cd Pi_Scratch_v273

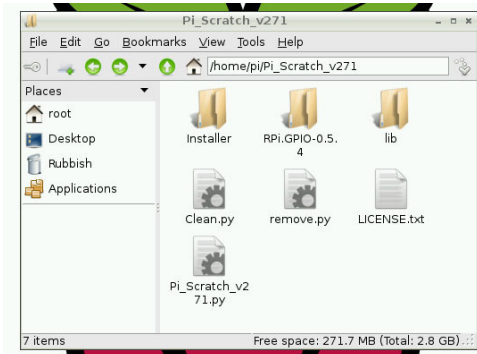


```
192.168.0.20 - PuTTY
root@raspberrypi:/home/pi# wget http://pridopia.co.uk/pi-pgm/Pi_Scratch_v271.tar
--2014-07-25 10:27:29-- http://pridopia.co.uk/pi-pgm/Pi_Scratch_v271.tar
Resolving pridopia.co.uk (pridopia.co.uk)... 79.170.44.105
Connecting to pridopia.co.uk (pridopia.co.uk)|79.170.44.105|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3369472 (3.2M) [application/x-tar]
Saving to: 'Pi_Scratch_v271.tar'

100%[=====>] 3,369,472    725K/s   in 7.8s

2014-07-25 10:27:39 (423 KB/s) - 'Pi_Scratch_v271.tar' saved [3369472/3369472]

root@raspberrypi:/home/pi# tar xf Pi_Scratch_v271.tar
root@raspberrypi:/home/pi# cd Pi_Scratch_v271
root@raspberrypi:/home/pi/Pi_Scratch_v271# dir
Clean.py  lib      Pi_Scratch_v271.py  RPi.GPIO-0.5.4
Installer LICENSE.txt  remove.py
root@raspberrypi:/home/pi/Pi_Scratch_v271#
```



You will find inside folder **Pi_Scratch_v273**

have 1 library folder **lib**

one WebIoPi-0.6.0 **support DAC function**

one installer folder include two python tools **Install-With-Scratch.py** & **Install-Without-Scratch.py** for setting Auto Login & Auto Load Pi_Scratch

Install-autorun-Scratch.py - for auto run example without press any key

one Tools install all necessary program **install.py**

RFID-Installer.py Install RFID tools pack

one License document **LICENSE.txt**

one main program **Pi_Scratch_v273.py**

* set up the auto login & auto start manually.

setting Auto login -> startx -> Scratch -> load file for Pi_Scratch

1. Copy the two [.desktop] files to `~/.config/autostart/` and to the desktop at `~/Desktop`

You can use "**sudo python Install-With-Scratch.py**" this will help you create a folder and copy two files to the folder & install Auto login -> startx -> Pi_Scratch

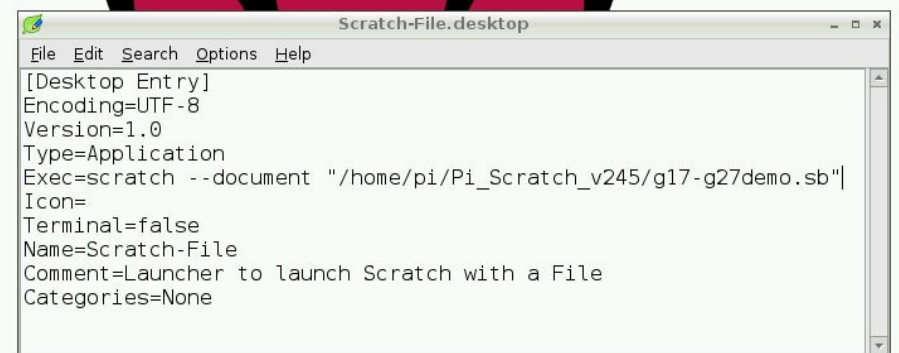
You can use "**sudo python Install-Without-Scratch.py**" this will help you create a folder and copy two files to the folder & install Auto login

Two desktop files you can modify

Scratch-Handler



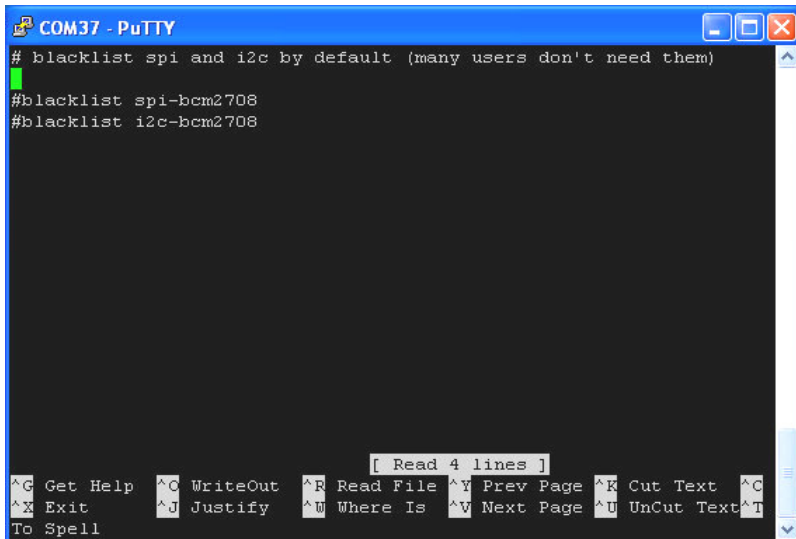
Scratch-File



1. Make sure your I2C driver are enable

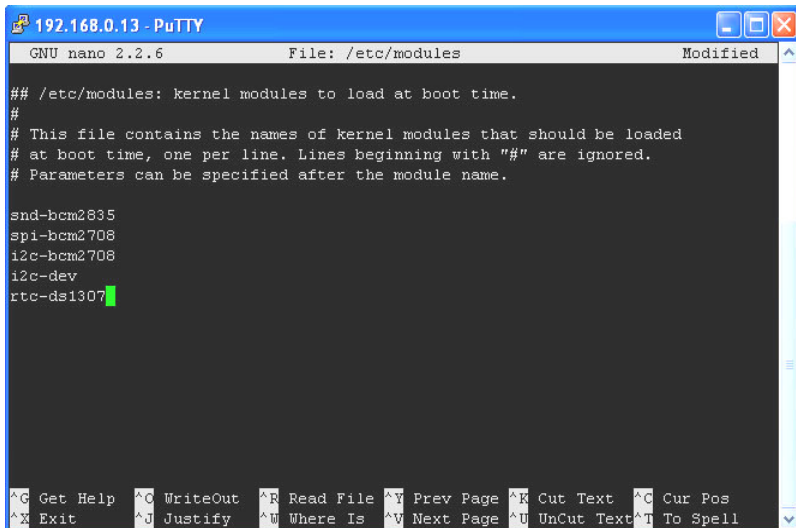
To enable it all you need to do is comment out a line by putting # in front

```
sudo nano /etc/modprobe.d/raspi-blacklist.conf
```



2. Add i2c-dev in /etc/modules by use

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



3. Next install the **python-smbus** python module:

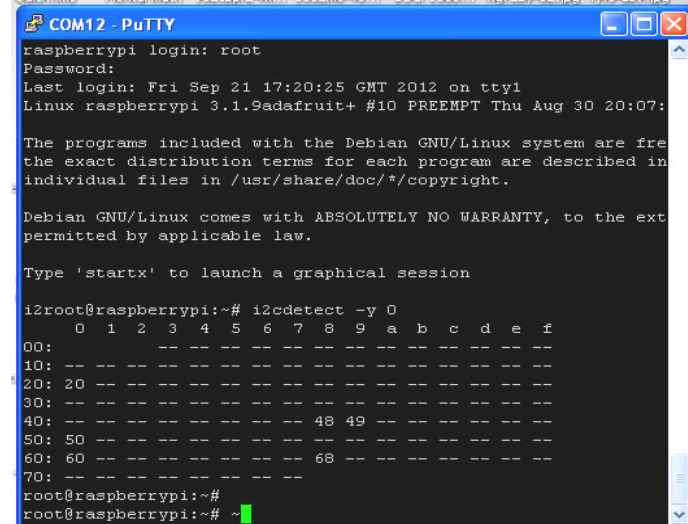
```
sudo apt-get install python-smbus
```

```
sudo apt-get install i2c-tools
```

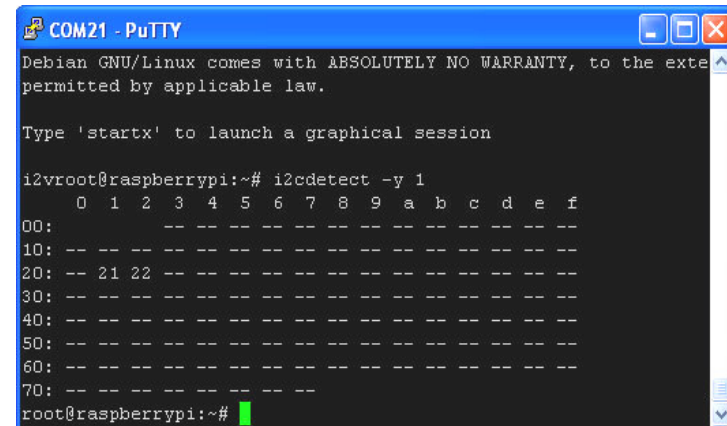
Now you are ready to use the i2c with python.

If you already install I2c driver ,

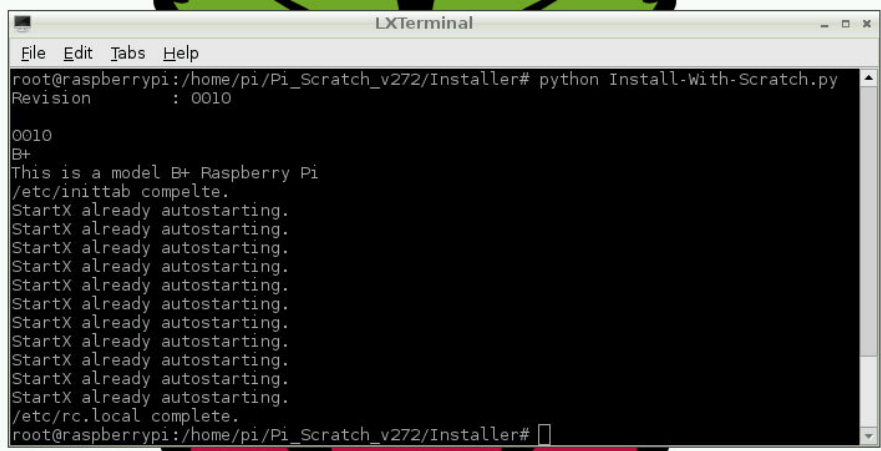
i2cdetect -y 0 - for Raspberry Pi V1 board



i2cdetect -y 1 - for Raspberry Pi V2 Board



B+ detect and install



```
root@raspberrypi:/home/pi/Pi_Scratch_v272/Installer# python Install-With-Scratch.py
Revision : 0010
0010
B+
This is a model B+ Raspberry Pi
/etc/inittab complete.
StartX already autostarting.
StartX already autostarting.
StartX already autostarting.
StartX already autostarting.
StartX already autostarting.
StartX already autostarting.
StartX already autostarting.
StartX already autostarting.
StartX already autostarting.
StartX already autostarting.
/etc/rc.local complete.
root@raspberrypi:/home/pi/Pi_Scratch_v272/Installer#
```

In Installer folder , type "python Install-With-Scratch.py" System will auto detect model B+, and install Pi_Scratch for model B+
In "Pi_Scratch_v272-B" folder

install other necessary software package

use our **install.py** will help you install WebIOPi , spidev & RPIO

sudo python install.py

(1) install WebIOPi

goto folder "WebIOPi-0.6.0" → "Python" →

run the command:

sudo python setup.py install

(2) install spidev

sudo apt-get install git

Then install py-spidev:

git clone git://github.com/doceme/py-spidev

cd py-spidev/

sudo python setup.py install

(3) install GPIO Library

<https://pypi.python.org/pypi/RPi.GPIO> GPIO library

GPIO library - RPi.GPIO-0.5.6.tar.gz

Install python , library and run the test program

sudo apt-get install python-dev

wget http://www.pridopia.co.uk/pi-pgm/RPi.GPIO-0.5.6.tar.gz

gunzip RPi.GPIO-0.5.6.tar.gz

tar -xvf RPi.GPIO-0.5.6.tar

cd RPi.GPIO-0.5.6

sudo python setup.py install

(4) install **RPIO** 0.10.0

<https://pypi.python.org/pypi/RPIO#downloads>

sudo apt-get install python-setuptools

sudo easy_install -U RPIO

Install IR diver by manually

(5) LIRC module

sudo apt-get install lirc

(6) python LIRC module

<https://pypi.python.org/pypi/python-lirc/1.1>

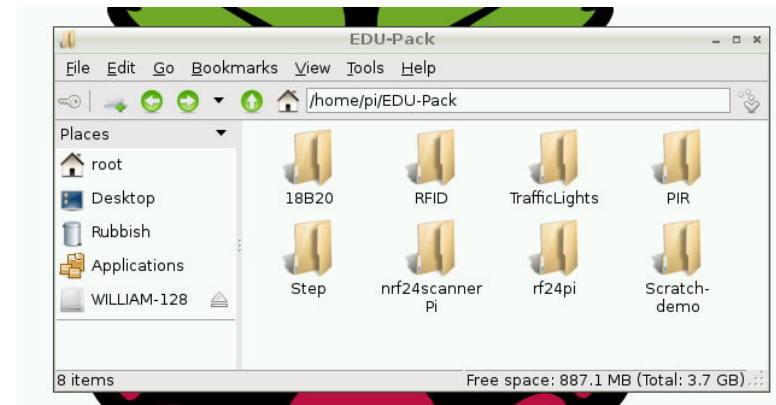
sudo apt-get install python-lirc

finished all necessary software install.

sudo Pi_Scratch_v273.py

sudo Pi_Scratch_v273.py -d in debug mode with more information display

sudo wget http://pridopia.co.uk/pi-pgm/EDU-Pack.tar



EDU Learning Board Software Pack

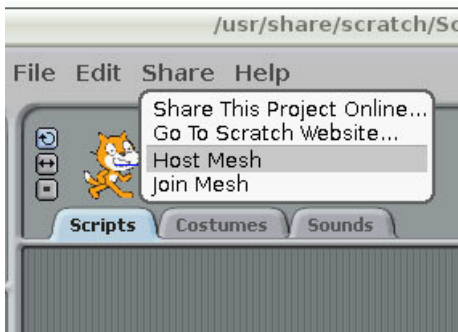
```

LXTerminal
File Edit Tabs Help

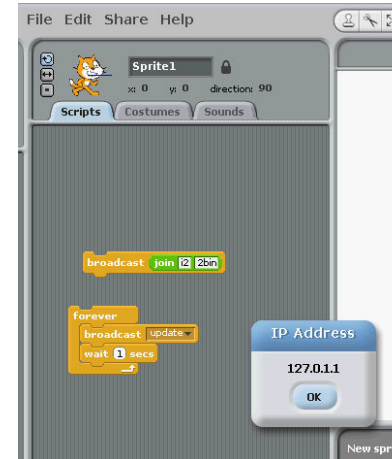
Reseting PCA9685
Setting PWM frequency to 60 Hz
Estimated pre-scale: 100
Final pre-scale: 101
Reseting PCA9685
Setting PWM frequency to 60 Hz
Estimated pre-scale: 100
Final pre-scale: 101
12510
Scratch not up. Sleeping for 5 and trying again.
0x20 not found.
LED x 32 Board loaded MCP23017.
Pridopia LED x 32 Board not loaded MCP23S17.
Reseting PCA9685
Setting PWM frequency to 60 Hz
Estimated pre-scale: 100
Final pre-scale: 101
Reseting PCA9685
Setting PWM frequency to 60 Hz
Estimated pre-scale: 100
Final pre-scale: 101
12510
Scratch not up. Sleeping for 5 and trying again.

```

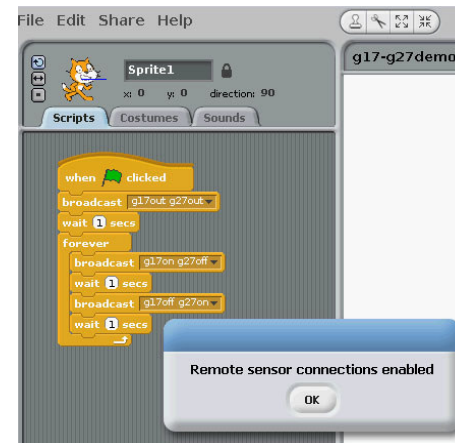
The Program will run and waiting for Scratch connect ...



1. Shift-click the Share in menu bar choose "Host Mesh".
2. choose Host Mesh --- will appear a IP address windows



The Program run in back ground detect all the device connect to Raspberry Pi. You can use Scratch with GPIO control function now.



*** For the first time use Pi_Scratch

Load "g17-g27demo.sb" demo program first, shift-click "Share" → Host Mesh --→ will appear a IP address windows "Ok"

Then save this project in your pi again.

Next time you can load "g17-g27demo" the scratch will automatic connect our "Pi_Scratch", with "Remote sensor connections enable" window, you don't need choose "Host Mesh" every time"

*** You can run Scratch in PC use "Join Mesh"

To control Raspberry Pi (need IP for your Raspberry Pi)

Raspberry Pi in " Host Mesh"

```
LXTerminal
File Edit Tabs Help
12510
Connected!
0x20 not found.
LED x 32 Board loaded MCP23017.
Pridopia LED x 32 Board not loaded MCP23S17.
Resetting PCA9685
Setting PWM frequency to 60 Hz
Estimated pre-scale: 100
Final pre-scale: 101
Resetting PCA9685
Setting PWM frequency to 60 Hz
Estimated pre-scale: 100
Final pre-scale: 101
12510

BHP085 : False
lcd2004g : False
SpiButton : False
lcd2004 : False
lcd1602g : False
8Switch5 : False
16x16Matrix : False
Servo : True
0x22 : True
```

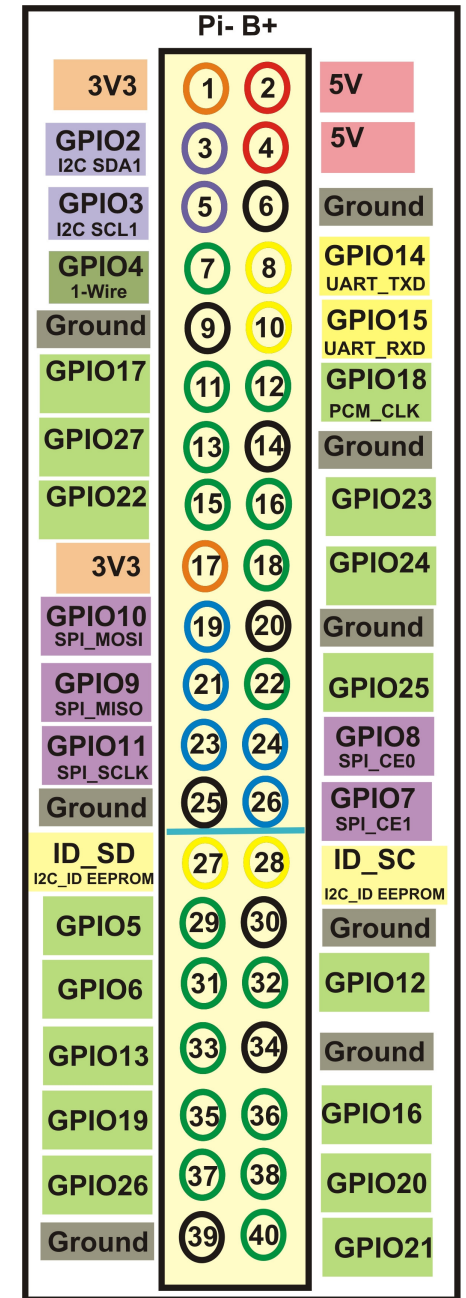
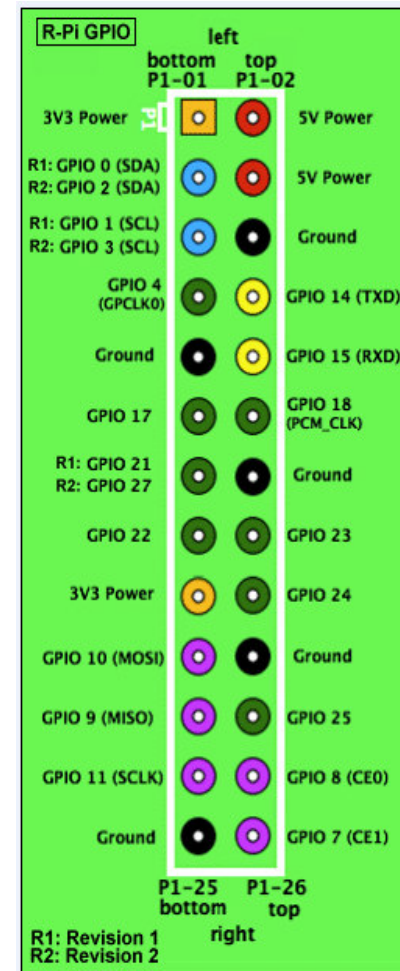
Minimum Requirements – a Raspberry Pi with Raspbian (wheezy) installed

	StepMotor1	StepMotor2	DS18B20 Temp Sensor	IR remote Sensor	PIR motion sensor	Buzzer
GPIO17	v					
GPIO18	v					
GPIO27	v					
GPIO22	v					
GPIO23		v		v	v	
GPIO24		v				v
GPIO25		v				
GPIO4		v	v			

	Green LED 1	Yellow LED 1	Red LED 1	Green LED 2	Yellow LED 2	Red LED 2
GPIO12	v					
GPIO13		v				
GPIO19			v			
GPIO16				v		
GPIO21					v	
GPIO26						v

SPI signal	RFID	NRF24L01		Micro SW 1	Micro SW 2	Mini Fan
GPIO7(CE1)				GPIO20	v	
GPIO8(CE0)	v	v		GPIO6	v	
GPIO9(MISO)	v	v		GPIO5		v
GPIO10(MOSI)	v	v				
GPIO11(SCLK)	v	v				
GPIO25		v				

i2c signal	RTC
GPIO2(SDA)	v
GPIO3(SCL)	v

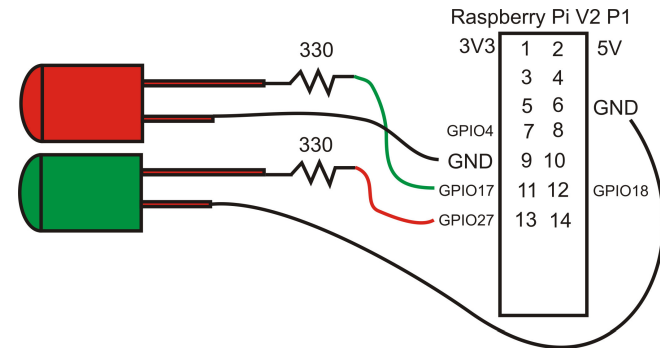




Scratch demo code

Our "Pi_Scratch" for scratch V1.4 driver

The main goal for this software is towards helping children / beginner programmers / hardware enthusiasts build up their skills by doing simple tasks with even easier commands / controls.



1. The Pi_Scratch use GPIO number (BCM) not raspberry pi pin number. so can support V2 P5 (GPIO 28,29,30,31) four extra GPIO pins.

2. example 1) define GPIO as input or output

g17out ---> GPIO17 output (P1 pin 11) g17in --> GPIO17 input

2) set GPIO g17on --> GPIO17 on(High) g17off ---> GPIO17 off(Low)

Example : 1. G17out g27out (g17,g27 LED on off) Delay 1 sec

2 G17on g27off delay 2 sec

3 G17off g27on delay 2sec

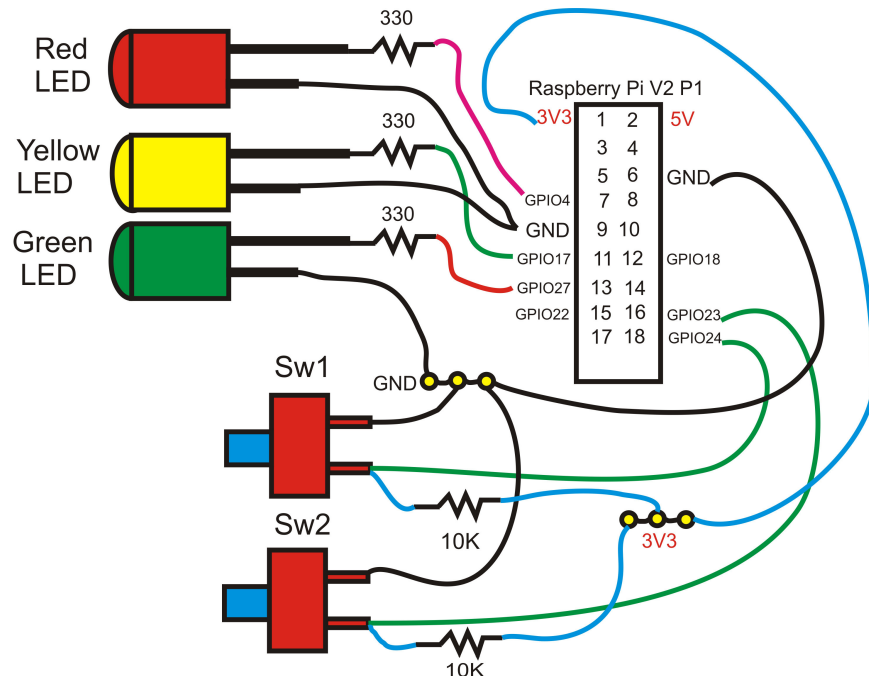
4 goto step 2

example Program g17-g27demo.sb

g17-g27demo.py

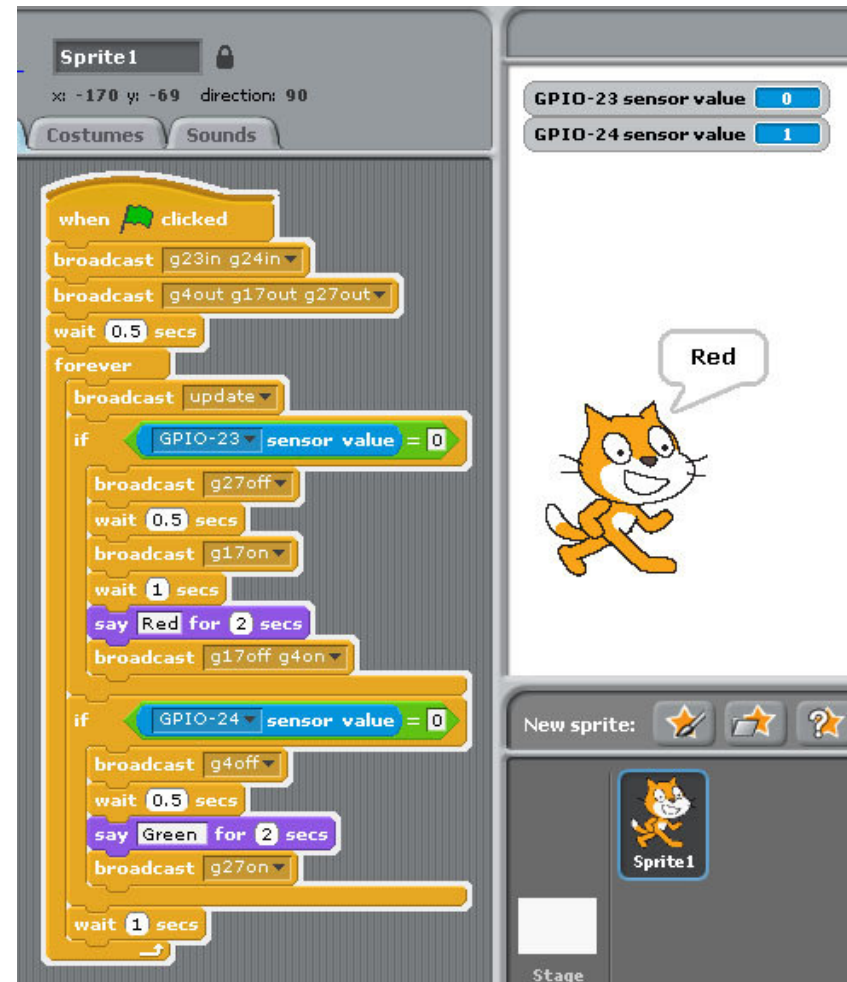


1. BASIC function 1 BASIC GPIO input & output demo

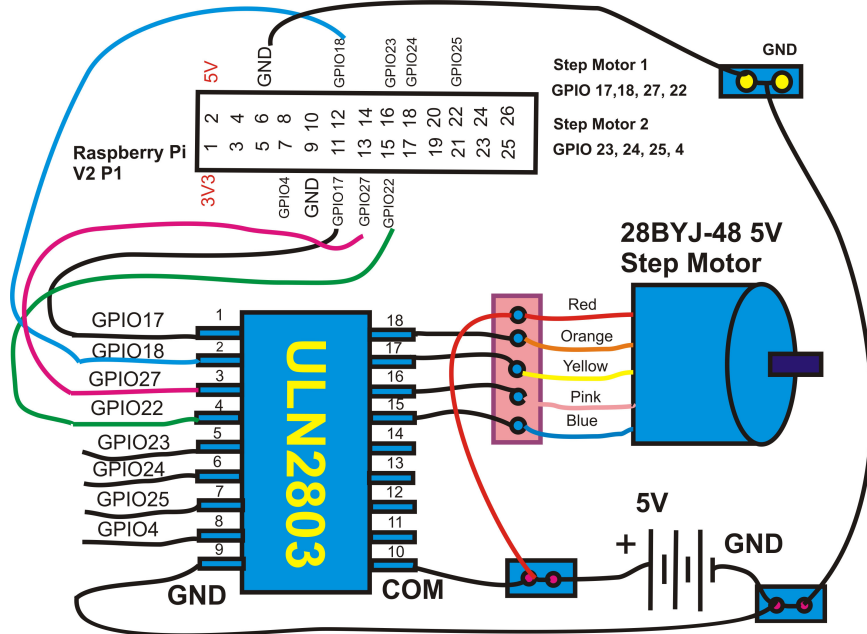


example Program RYG-LED-SW-demo.sb

1. define GPIO 23, 24 as input
2. define GPIO 4, 17, 27 as output
3. press sw2 yellow LED on 1 sec, then Red LED "ON" The Cat say "Red"
4. press sw1 Red LED off 1 sec, then Green LED "ON" The Cat say "Green"

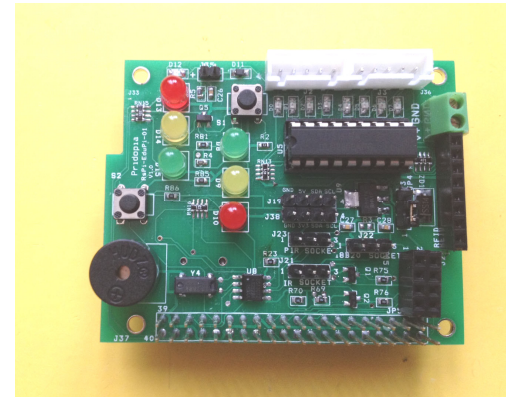


2 BASIC function 2 BASIC GPIO Step Motor demo



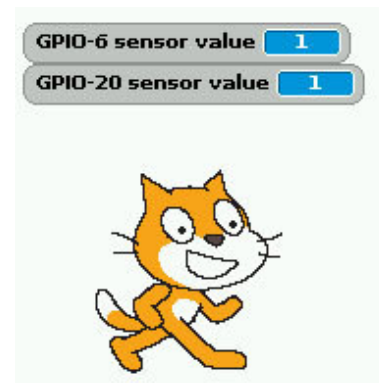
command "SMOTORainit" initial GPIO 17,18,27,22 as A Step Motor Port
 command "SMOTORbinit" initial GPIO 23,24,25,7 as B Step Motor Port
 command "SMOTOR" + speed (3-50) +(a or B) + output step
 100 --- clockwise 100 step , -100 --- anticlockwise 100 step
 a -- GPIO 17,18,27,22 , b -- GPIO 23,24,25,4

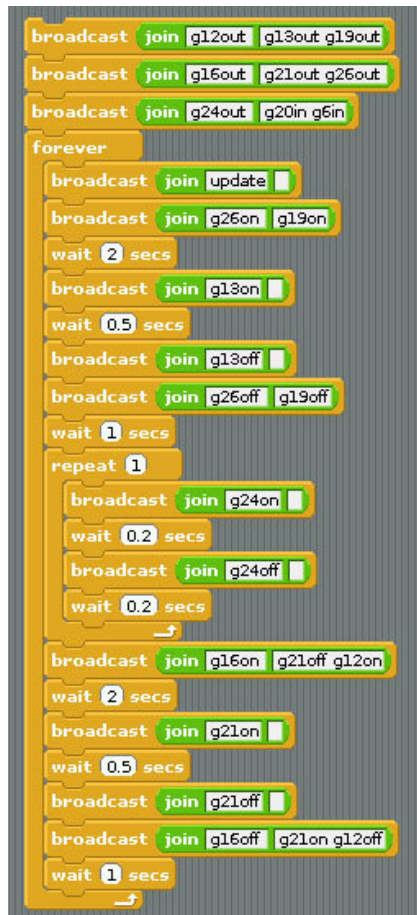
3. Raspberry Pi B+ (B Plus) EDU Learning Board support



first set traffic light Red (GPIO19), Yellow(GPIO13), Green (GPIO12)
 ,Switch 1(GPIO20)
 second set traffic light Red (GPIO16), Yellow(GPIO21), Green (GPIO26)
 ,Switch 2(GPIO6)

1. Setting GPIO 19,13,12 as output for R,Y,G LED
2. Setting GPIO 16,21,26 as output for R,Y,G LED
3. Setting GPIO 20,6 as input for 2 Switch





BASIC traffic light demo

1. EDU Learning Board RTC setting
I2C DS1307 RTC 68 -> RTC DS1307

```

192.168.0.9 - PuTTY
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Dec 8 17:08:10 2014
root@raspberrypi:~# i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  68  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
root@raspberrypi:~#

```

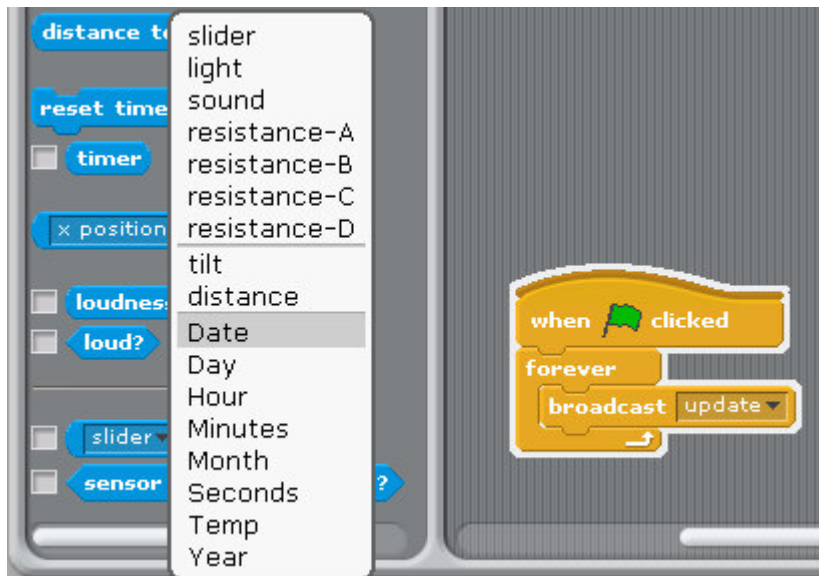
```

192.168.0.9 - PuTTY
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Dec 8 09:32:22 2014 from 192.168.0.13
root@raspberrypi:~# i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  UU  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
root@raspberrypi:~#

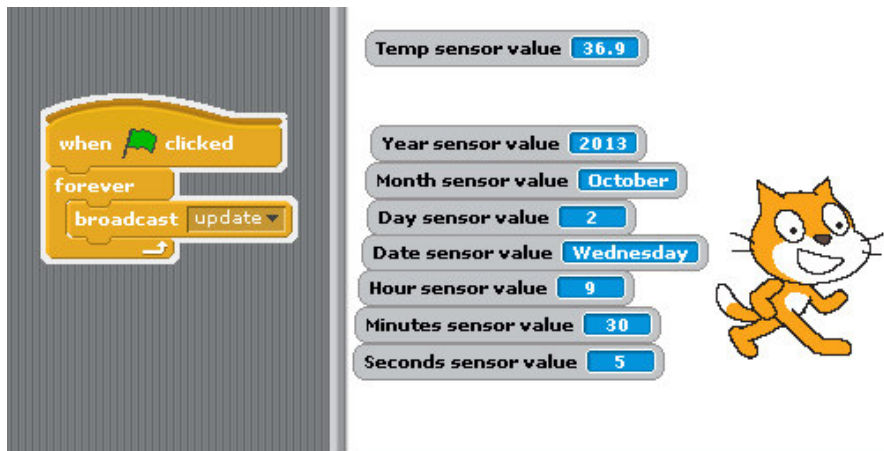
```

i2c address 68 DS1307 RTC detect & demo



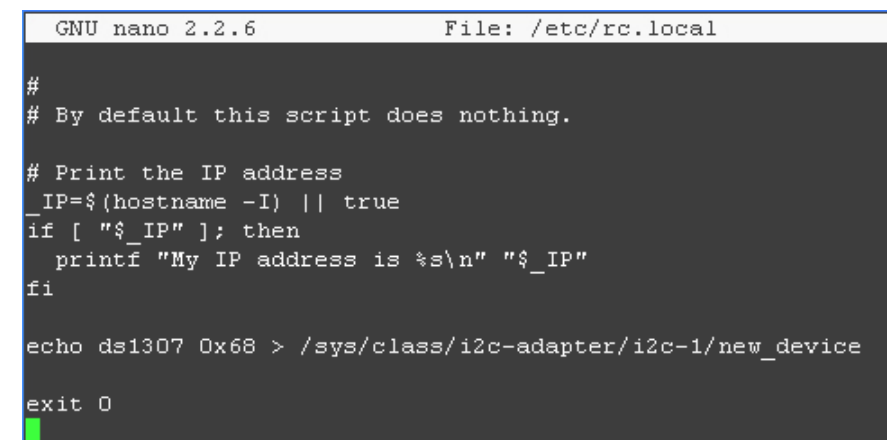
after broadcast "Update" in Sensing --> Slider, you will see the RTC in the list

"Year, Month, Day, Date, Hour, Minutes, seconds"

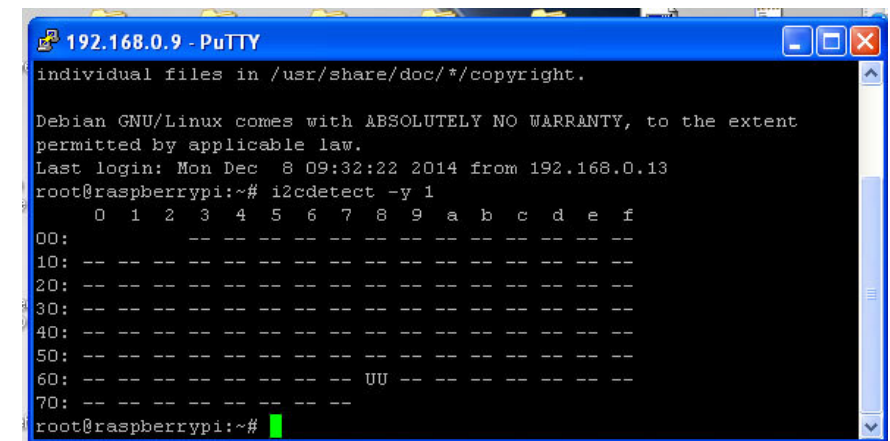


RTC information from Raspberry Pi i2c RTC module

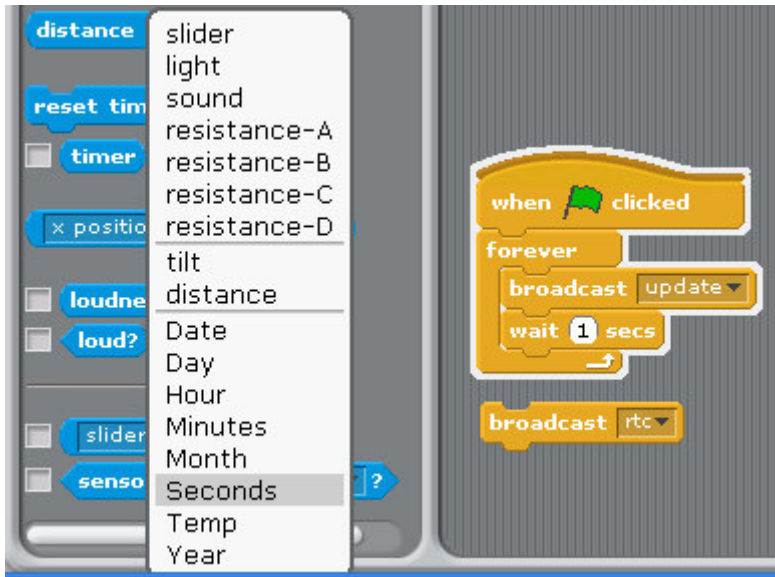
if you can't detect RTC function in Scratch



you can add ds1307 in /etc/rc.local



in 0x68 address display "UU"

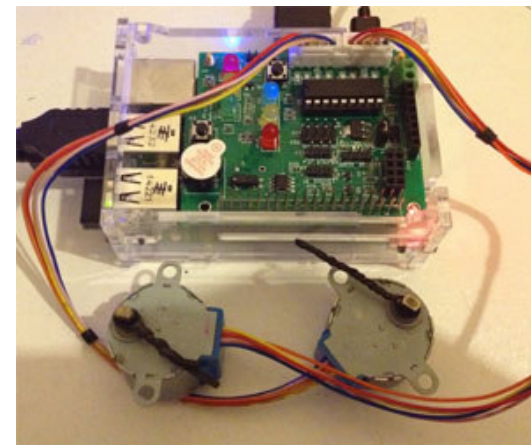
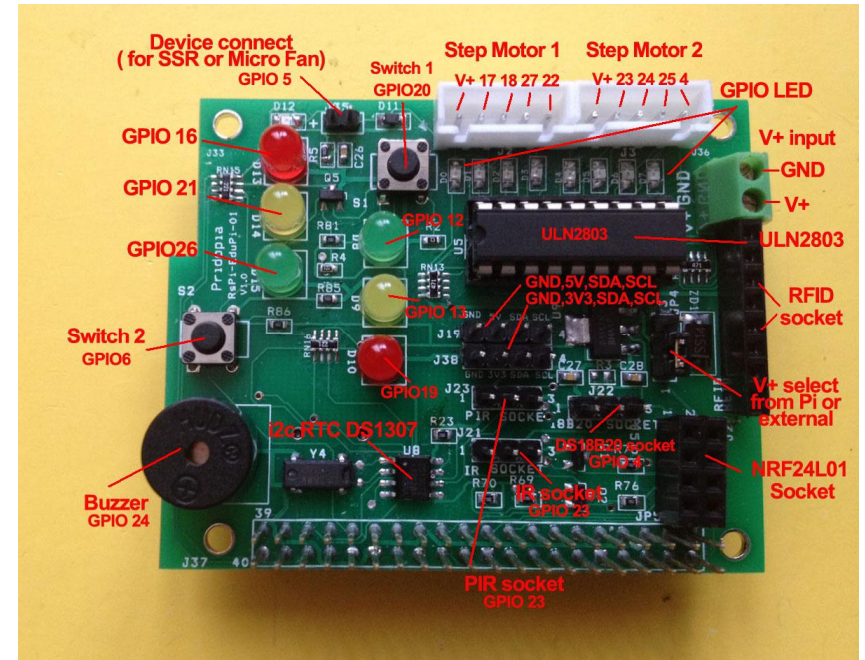


send broadcast "rtc" then broadcast "update"

after broadcast "Update" in Sensing --> Slider, you will see the RTC in the list

"Year, Month, Day, Date, Hour, Minutes, seconds"

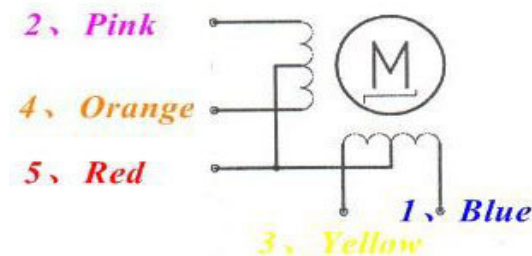
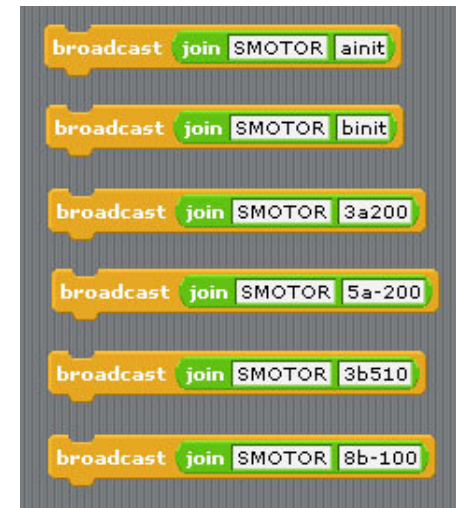
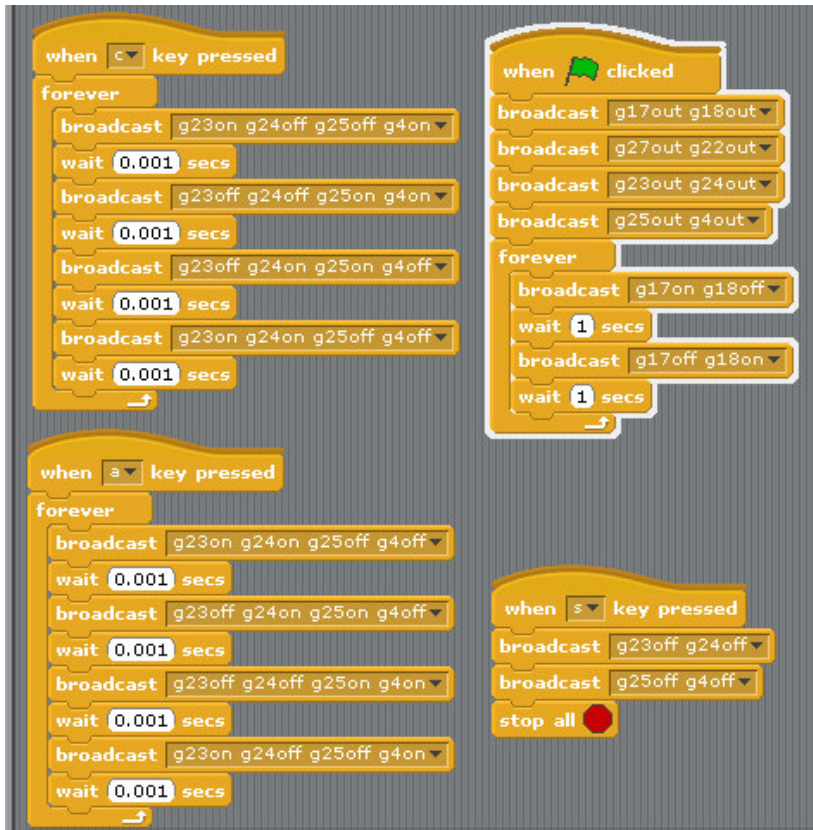
2. Stepper Motor



main program turn relay 1, on 2 off -> 1 off, 2 on on cycle

press C Stepper Motor clockwise press A Stepper Motor anticlockwise

Press S Stop



(2) Step Motor smotor command easy way

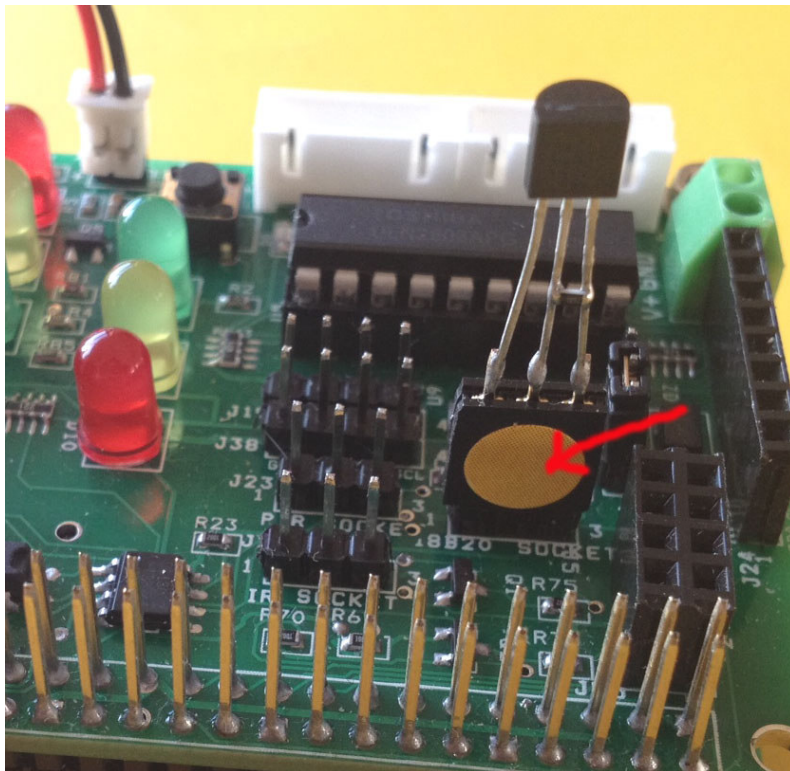
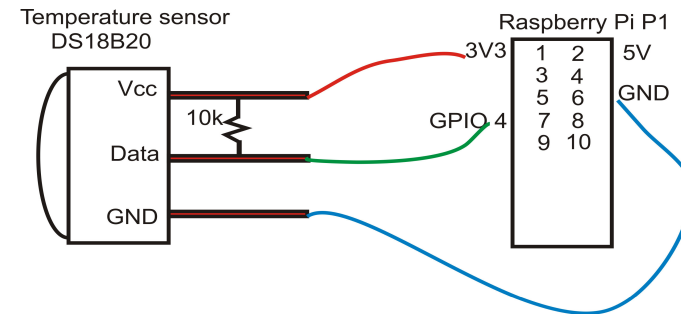
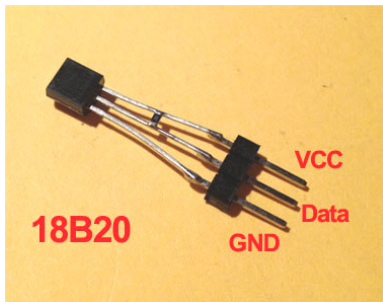
command "SMOTORainit" initial GPIO 17,18,27,22 as A Step Motor Port
 command "SMOTORbinit" initial GPIO 23,24,25,7 as B Step Motor Port
 command "SMOTOR" + speed (3-50) +(a or B) + output step
 100 --- clockwise 100 step , -100 --- anticlockwise 100 step
 a -- GPIO 17,18,27,22 , b -- GPIO 23,24,25,4

5V Step Motor

Lead Wire Color	---> CW Direction (1-2 Phase)							
	1	2	3	4	5	6	7	8
4 ORG	-	-						-
3 YEL		-	-	-				
2 PIK				-	-	-		
1 BLU						-	-	-

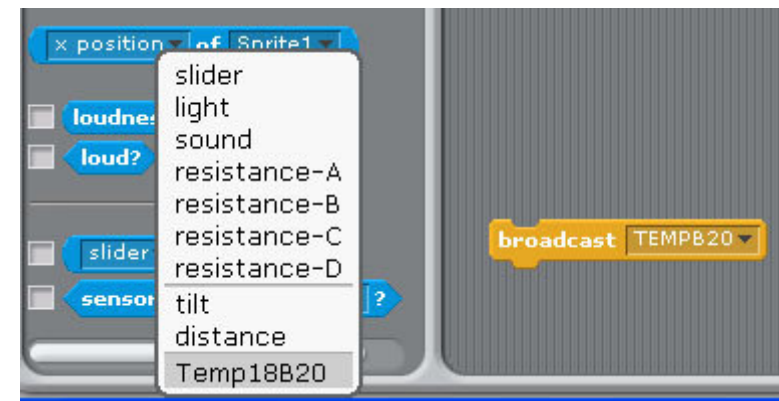
(2) Stepper Motor Test

3. 1-Wire 18B20 temp Sensor



VCC to 3.3V P1 pin1, Data to P1 pin7(GPIO4) , GND to P1 pin6(GND)

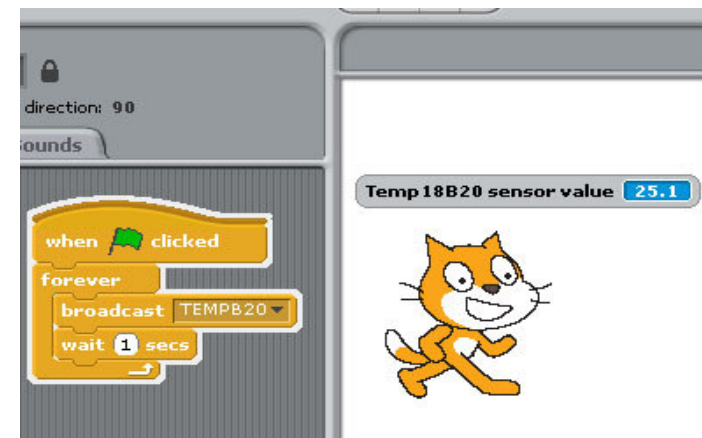
Data & VCC we have one 10K SMD Resister



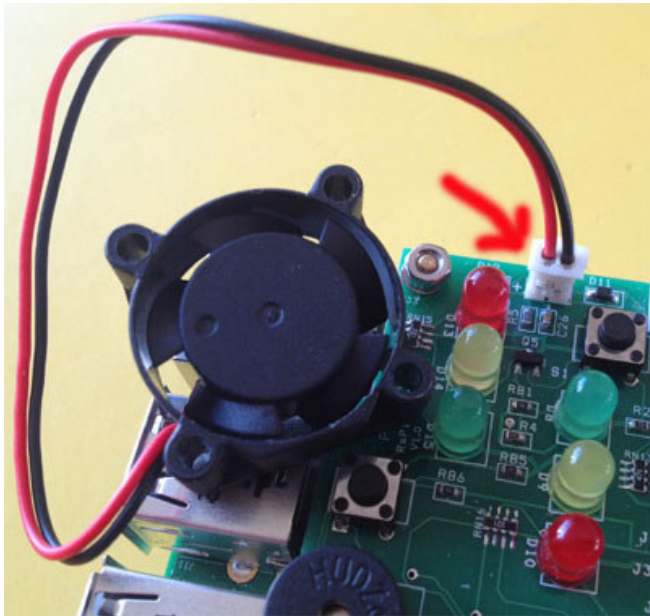
Command " TEMPB20" to active 1-wire in pin7(GPIO4)

1.broadcast ""TEMPB20"

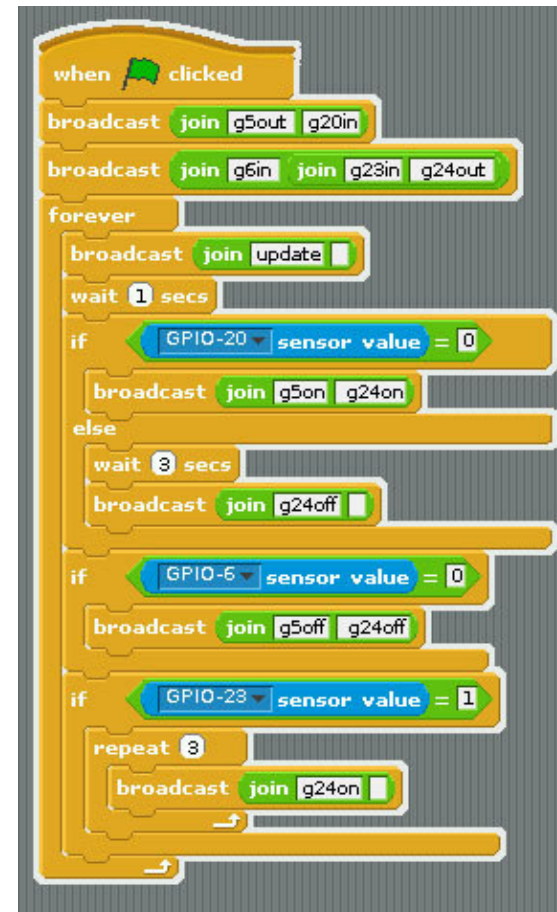
2. after broadcast in Sensing --> Slider ,
you will see the " Temp18B20" in the list



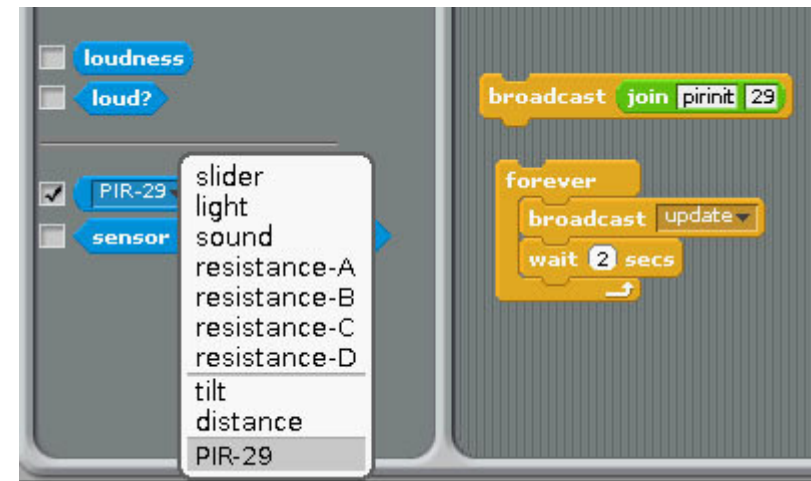
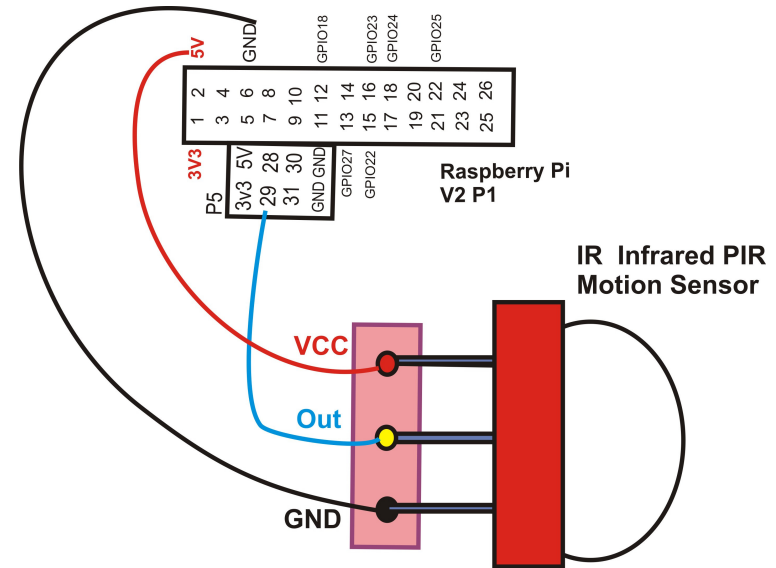
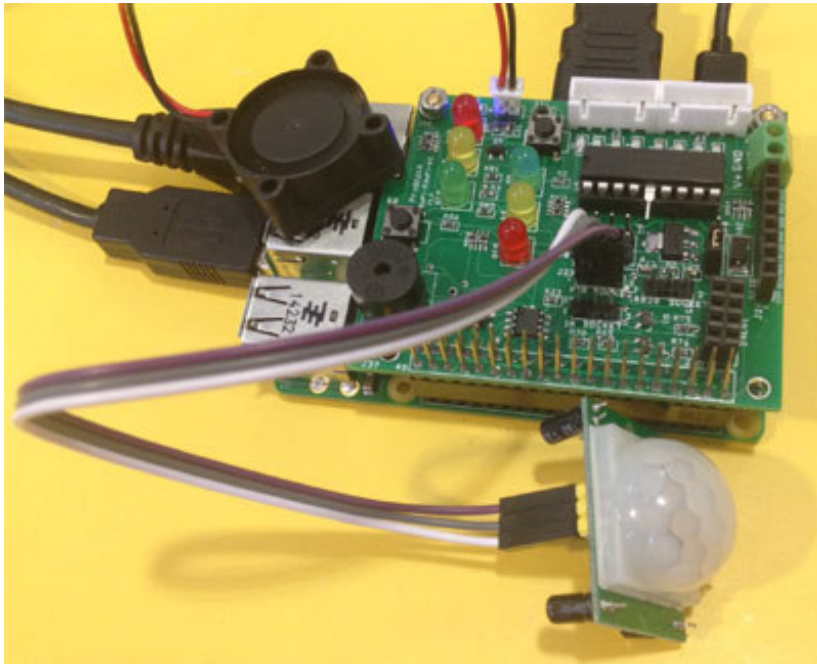
4. DC 5V MINI FAN control (GPIO5)



1. broadcast "g5" + "out" setting GPIO 5 as output for Mini Fan
broadcast "g20" + "in" setting GPIO 205 as Micro Switch input
2. broadcast "update"
3. in Sensing --> Slider, you will see the "GPIO-20" in the list



5. IR PIR Motion Sensor Module (GPIO23)

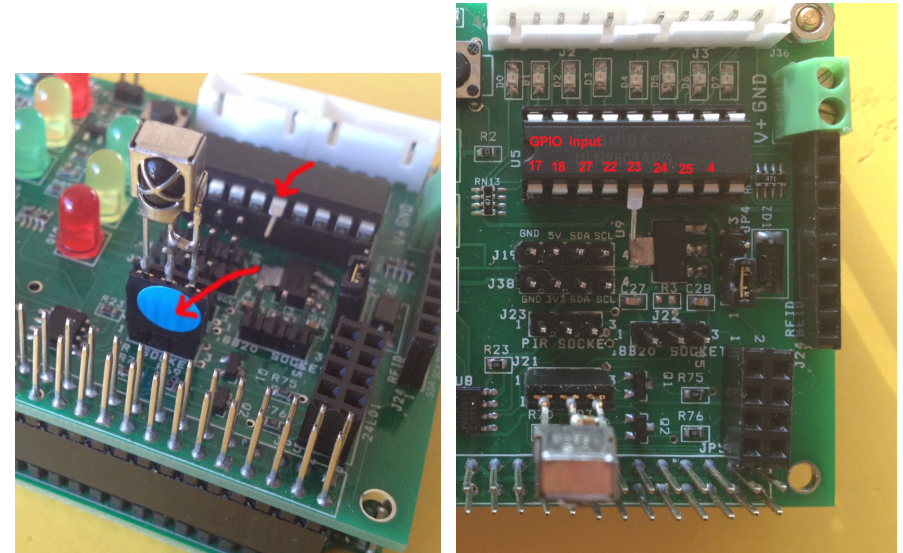


- 1.broadcast "pir" + "init" + "29" GPIO 29 as input
- 2.in Sensing --> Slider , you will see the "PIR-29" in the list

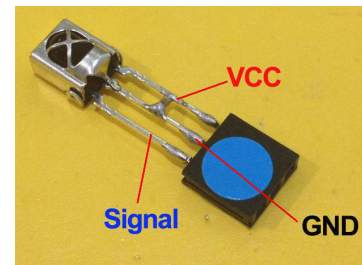


The PIR -29 value change between "0" & "1"

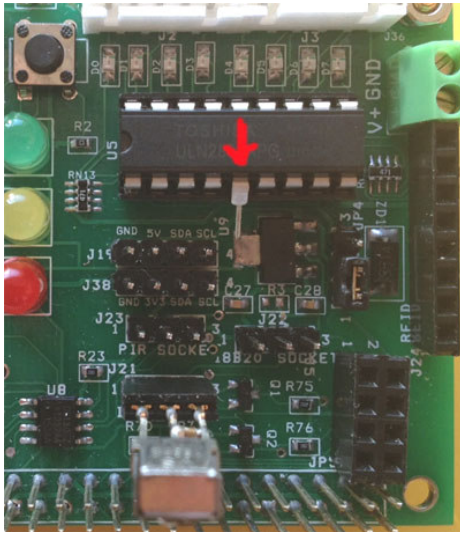
6. IR remote control (GPIO23)



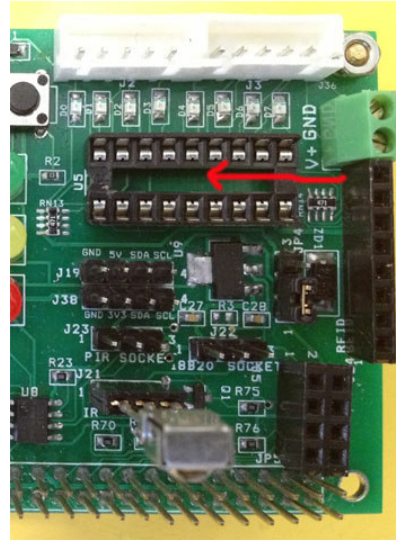
ULN2803 pin 5 input from GPIO23



Provide 20cm male to female cable, you can use this cable connect IR sensor to Pi without EDU board



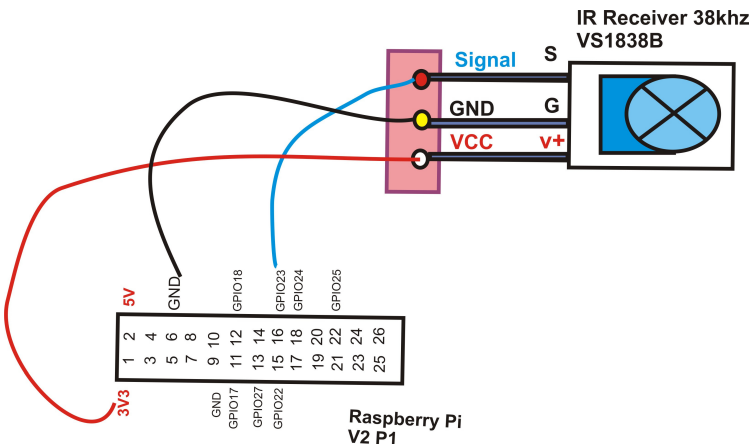
1. pin5 move



2. ULN2803 remove

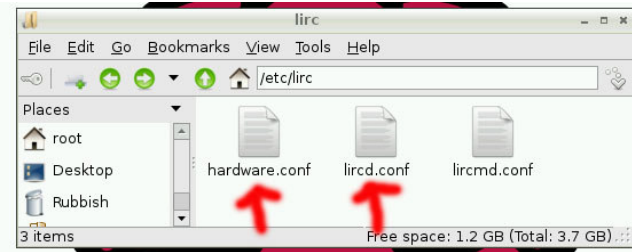
1. remove unln2803 pin 5 GPIO23 out of socket, use this way , so you can use one Stepper Motor and 1-Wire DAS18B20 sensor & IR remote sensor
or you can use way 2

2. remove uln2803 IC out of socket for IR remote sensor working correctly. Use this way you can have stepper Motor function.

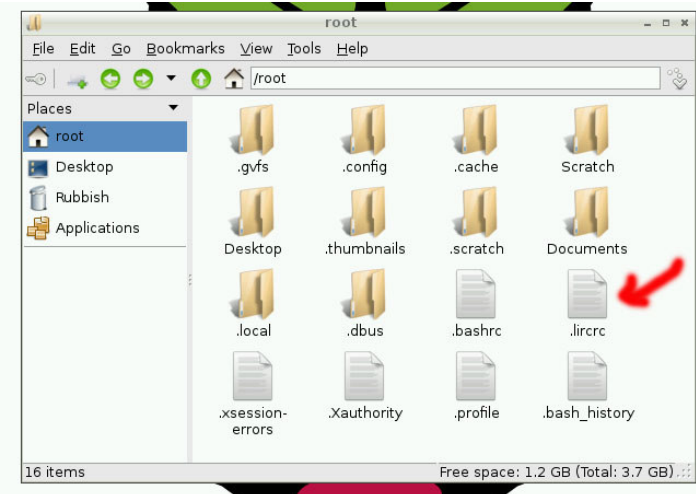


Download 3 Remote control setting files

[lircd.conf](#) [.lircrc](#) [hardware.conf](#)



Copy hardware.conf , lircd.conf & lircrc to /etc/lirc



Copy .lircrc to /root



Remote control example

```
192.168.0.26 - PuTTY
IRR_v2.py  Pi_Scratch_v253  Pi_Scratch_v256.tar  ServoArm
Logger    Pi_Scratch_v253.tar  python_games
root@raspberrypi:/home/pi# sudo python IRR_v2.py
IRR_v2.py:7: RuntimeWarning: This channel is already in use, continuing anyway.
Use GPIO.setwarnings(False) to disable warnings.
  GPIO.setup(17, 0)
CHAN_DOWN
CHAN_SEL
CHAN_UP
PREV
NEXT
PLAY
VOL_DOWN
VOL_UP
EQ
BUT_0
BUT_100+
BUT_200+
BUT_1
BUT_2
BUT_3
BUT_4
BUT_5
BUT_6
BUT_7
BUT_8
BUT_9
```

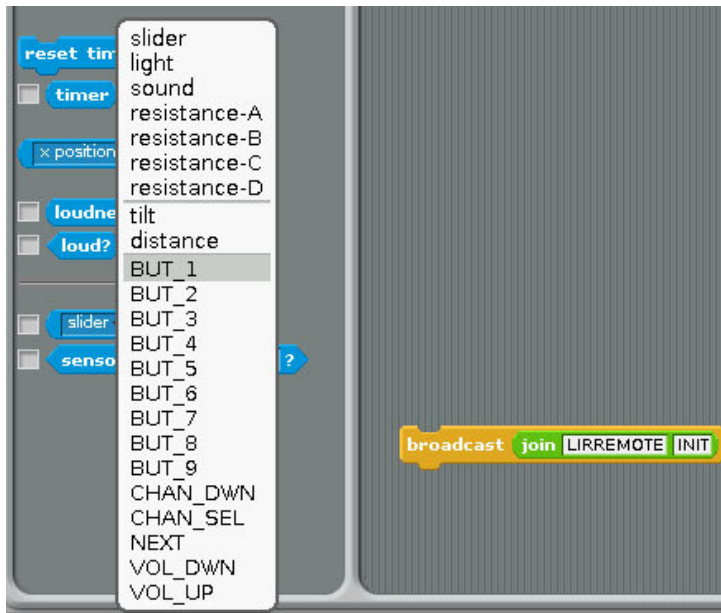
IRR_v2.py demo

```
192.168.0.26 - PuTTY
permitted by applicable law.
Last login: Tue Feb 25 14:23:24 2014
root@raspberrypi:~# irw
00000000000000000001 00 BUT_CHANNEL_DOWN /etc/lirc/lircd.conf
00000000000000000002 00 BUT_CHANNEL_SELECT /etc/lirc/lircd.conf
00000000000000000003 00 BUT_CHANNEL_UP /etc/lirc/lircd.conf
00000000000000000004 00 BUT_PREV /etc/lirc/lircd.conf
00000000000000000005 00 BUT_NEXT /etc/lirc/lircd.conf
00000000000000000006 00 BUT_PLAY /etc/lirc/lircd.conf
00000000000000000007 00 BUT_VOL_DOWN /etc/lirc/lircd.conf
00000000000000000008 00 BUT_VOL_UP /etc/lirc/lircd.conf
0000000000000000000a 00 BUT_0 /etc/lirc/lircd.conf
0000000000000000000b 00 BUT_100 /etc/lirc/lircd.conf
0000000000000000000c 00 BUT_200 /etc/lirc/lircd.conf
0000000000000000000d 00 BUT_1 /etc/lirc/lircd.conf
0000000000000000000e 00 BUT_2 /etc/lirc/lircd.conf
0000000000000000000f 00 BUT_3 /etc/lirc/lircd.conf
00000000000000000010 00 BUT_4 /etc/lirc/lircd.conf
00000000000000000011 00 BUT_5 /etc/lirc/lircd.conf
00000000000000000012 00 BUT_6 /etc/lirc/lircd.conf
00000000000000000013 00 BUT_7 /etc/lirc/lircd.conf
00000000000000000014 00 BUT_8 /etc/lirc/lircd.conf
00000000000000000015 00 BUT_9 /etc/lirc/lircd.conf
```

irw read information from remote control

```
192.168.0.26 - PuTTY
root@raspberrypi:~# irexec
Channel Down pressed
Channel SELECT pressed
Channel UP pressed
PREV pressed
NEXT pressed
PLAY pressed
VOL DOWN pressed
VOL UP pressed
EQ pressed
0 pressed
100+ pressed
200+ pressed
1 pressed
2 pressed
3 pressed
4 pressed
5 pressed
6 pressed
7 pressed
8 pressed
9 pressed
```

irexec read information from remote control



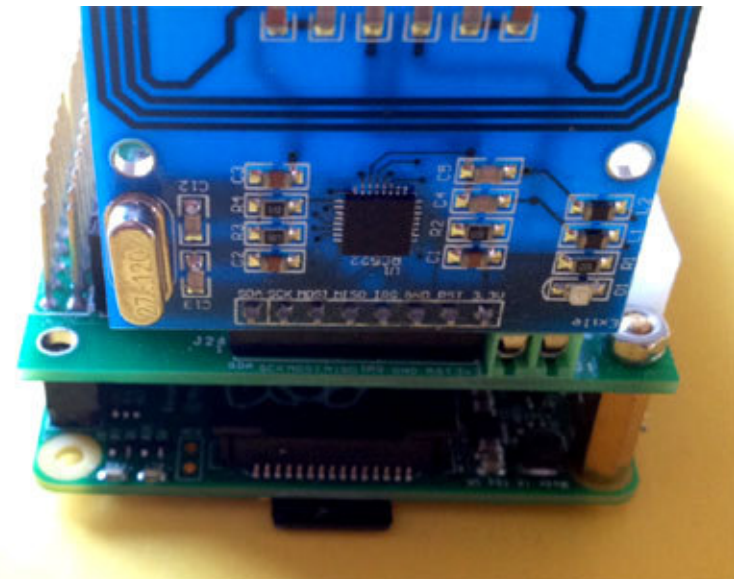
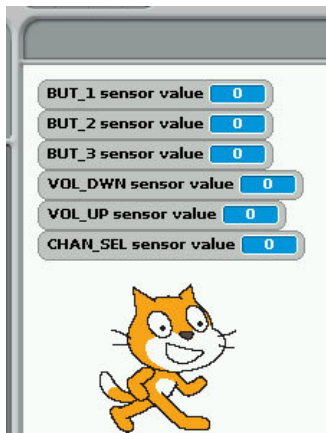
Scratch setting demo

7. RFID reader (SPI signal)



Command " LIRREMOTE" + "INIT" to active IR REMOTE CONTROL

1. broadcast "LIRREMOTEINIT"
2. after broadcast Press the key in remote control you want use in Scratch in Sensing --> Slider , you will see the key you press in the list



Install tools for RFID kit
In folder" Installer"

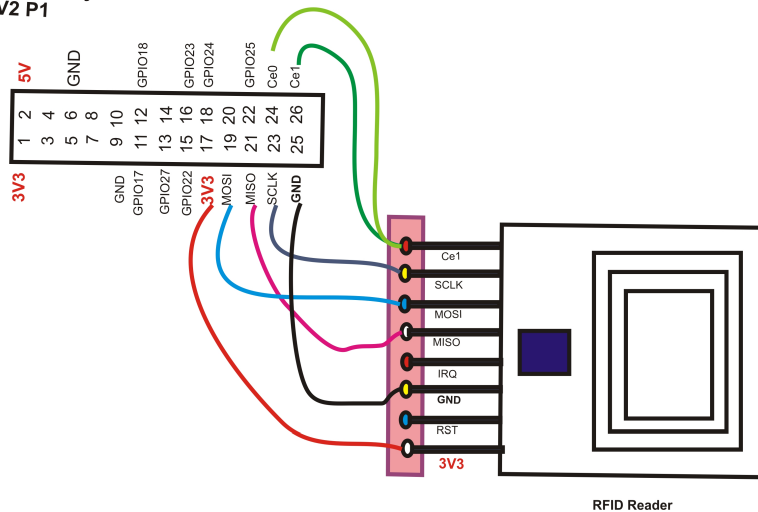


RFID socket

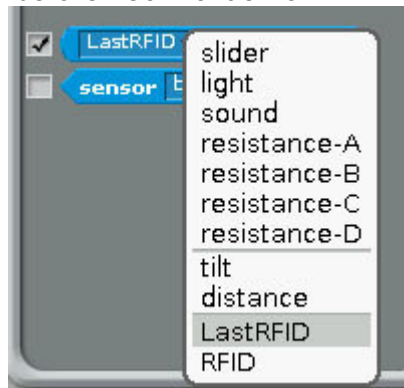
From pin1 (SDA,SCK,MOSI,MISO,IRQ,GND,RST,3V3)

(1) GPIO (SPi) RFID Reader & GPIO Relay output

Raspberry Pi
V2 P1



Scratch control demo

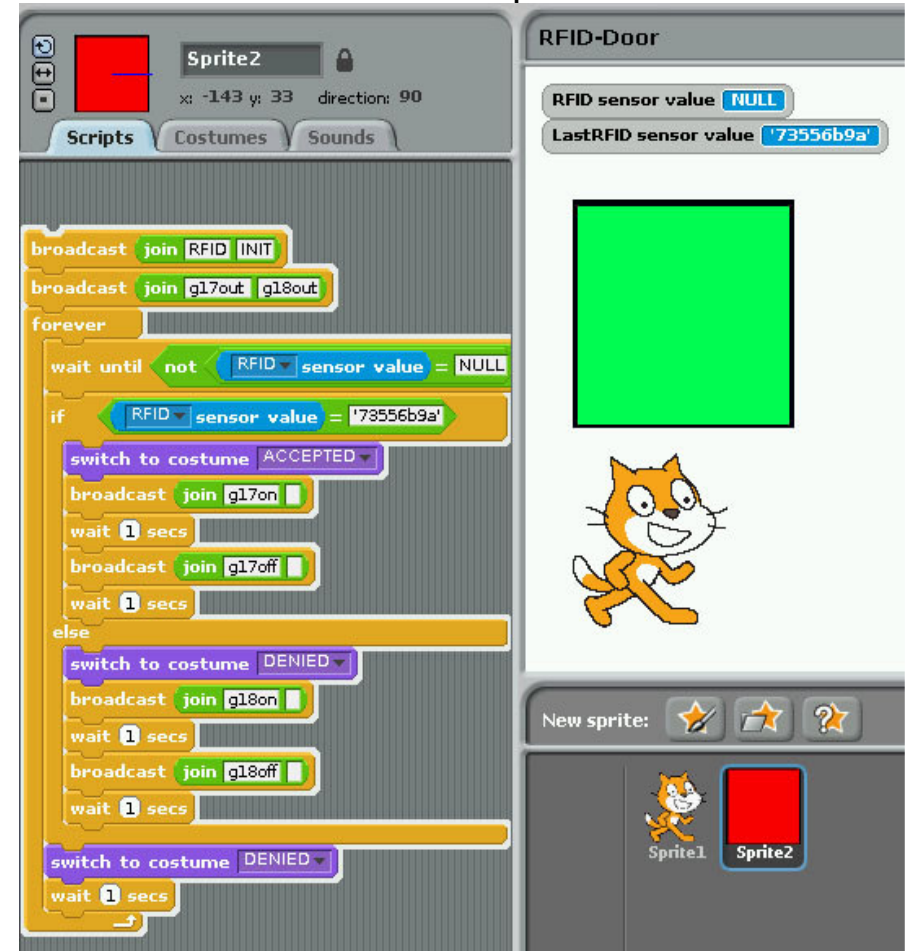


- 1) Command "RFID"+"INIT"+"0" for CE0
or Command "RFID"+"INIT"+"1" for CE1

will initial SPI signal to active RFID Reader

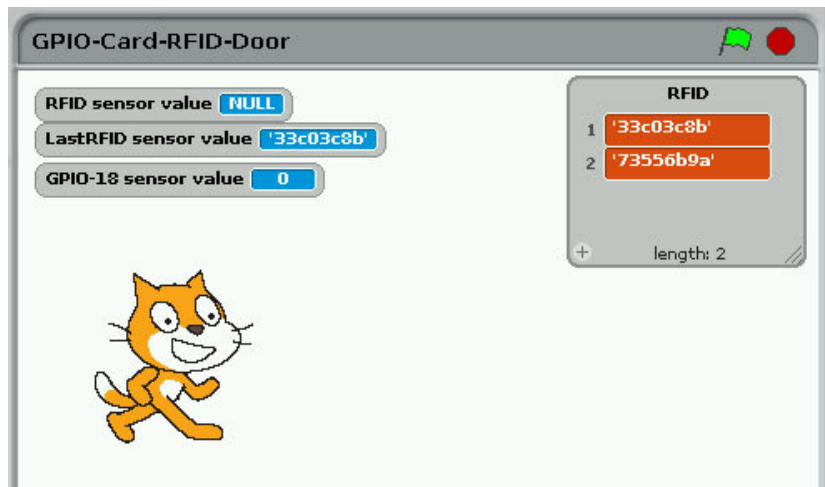
- 2) you will see "LastRFID" & "RFID" in Sensors

Scratch demo read RFID and GPIO output



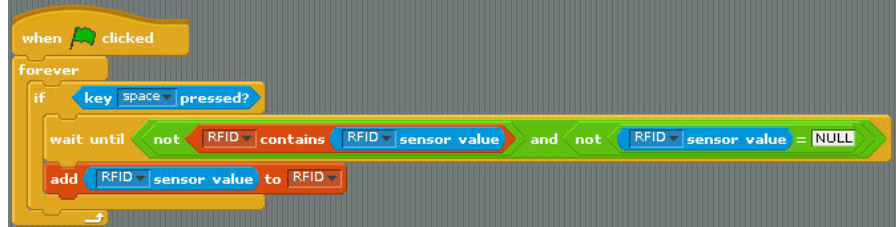
RFID reader & Relay & Step Motor control

scratch demo code

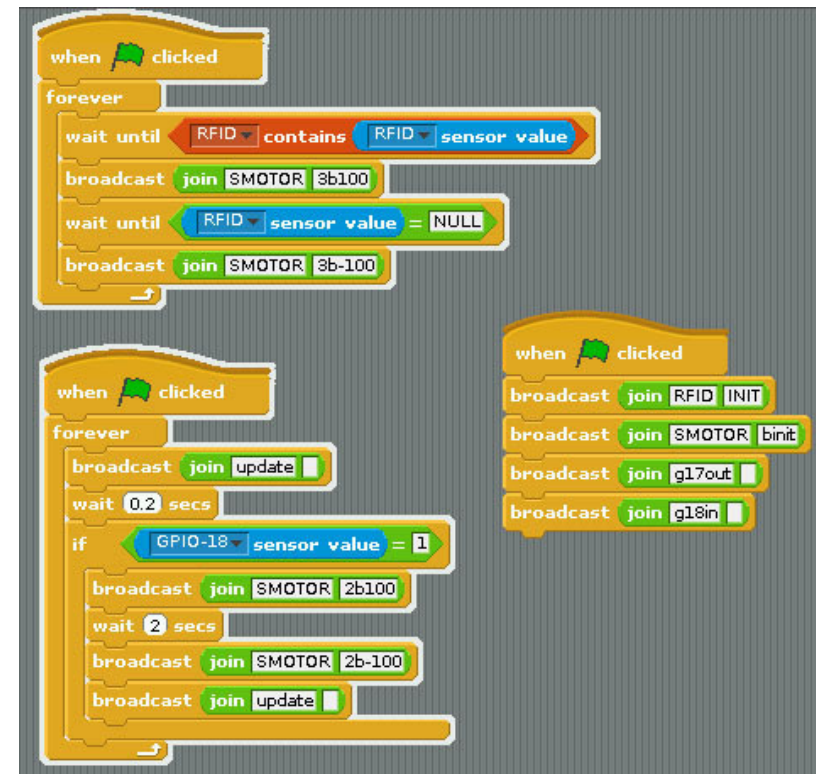


ADD RFID Card into data base

Press "space" key then scan your RFID card



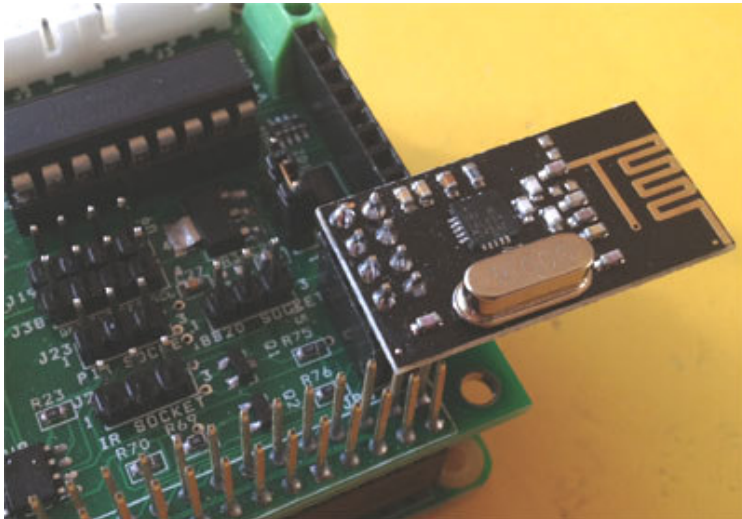
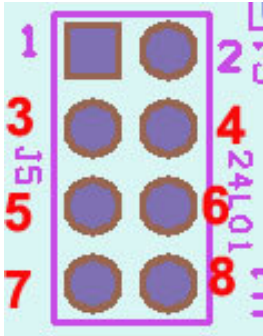
Active with Step Motor and GPIO Switch input



8. NRF24L01 Radio Frequency Reader module (SPI signal)

NRF24L01 socket

From pin1 (GND,3V3,GPIO25,CE0,SCLK,MOSI,MISO,X)



For test NRF24L01 module you need

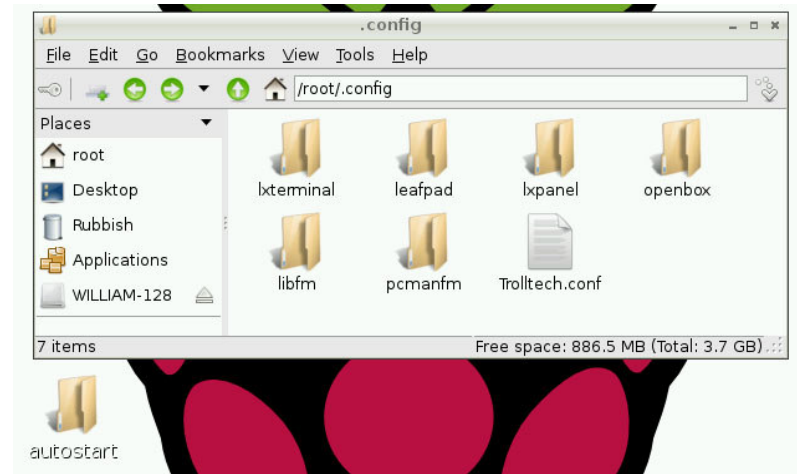
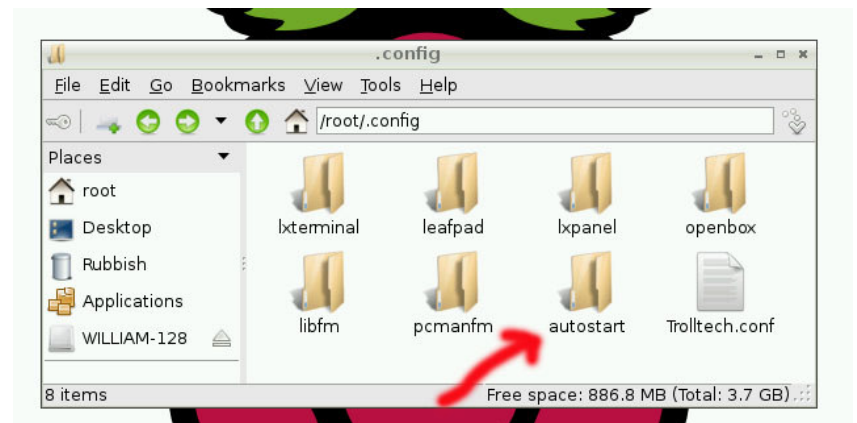
`Sudo nano /etc/rc.local`

Put # in front of `su -l root -c startx`

Or Or move "autostart" folder from /root/.config to desktop

```
192.168.0.9 - PuTTY
GNU nano 2.2.6 File: /etc/rc.local Modified
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi
echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device
# su -l root -c startx
exit 0
Get Help WriteOut Read File Prev Page Cut Text Cur Pos
Exit Justify Where Is Next Page UnCut Text To Spell
```

Or move "autostart" folder from /root/.config to desktop



```

192.168.0.16 - PuTTY
[3]+ Stopped ./scanner
root@raspberrypi:/home/pi/nrf24scannerPi# dir
bcm2835.c  bcm2835.o  README.md  scanner.cpp  scan_pi_slower
bcm2835.h  Makefile   scanner    scanner.o
root@raspberrypi:/home/pi/nrf24scannerPi# ./scanner
Starting Poor Man's Wireless 2.4GHz Scanner.. [raspberry pi]

Channel Layout
> 1 2 3 4 5 6 7 8 9 10 11 12 13 14 <
|                                     :W+a ++ |4
|                                     : =WR:   |5
|                                     W-  -    |3
|               +W                      W      |2
|                                     +W+++    |0
|               W W    W W              |1
|                                     *W        = |0
|               :  :  :  +++:: W+         |5
|               W W    W                |4
|               +W                      |1
|               WR :                     |2
|                                     |5
|                                     |0
> 1 2 3 4 5 6 7 8 9 10 11 12 13 14 <

```

Use `chmod +x scanner` , `chmod +x pingtest` , `chmod +x pongtest`

[illegible]

```

192.168.0.10 - PuTTY
root@raspberrypi:/home/pi/EDU-Pack/rf24pi/examples# dir
Makefile pingtest.cpp pongtest.cpp rpi-hub.cpp scanner.cpp sendto_hub.cpp
pingtest pongtest rpi-hub scanner sendto_hub
root@raspberrypi:/home/pi/EDU-Pack/rf24pi/examples# ./pingtest

RF24/examples/pingpair/
ROLE: Ping out
SPI device      = /dev/spidev0.0
SPI speed       = 8000000
CE GPIO  = 25
STATUS
RX_ADDR_P0-1    = 0x0e RX_DR=0 TX_DS=0 MAX_RT=0 RX_P_NO=7 TX_FULL=0
RX_ADDR_P2-5    = 0xf0f0f0f0e1 0xf0f0f0f0d2
RX_ADDR_P6-7    = 0xc3 0xc4 0xc5 0xc6
TX_ADDR         = 0xf0f0f0f0e1
RX_PW_P0-6      = 0x20 0x20 0x00 0x00 0x00 0x00
EN_AA           = 0x00
EN_RXADDR       = 0x02
RF_CH           = 0x4c
RF_SETUP        = 0x07
CONFIG          = 0x0f
DYNPD/FEATURE   = 0x00 0x00
Data Rate       = 1MBPS
Model           = nRF24L01+
CRC Length      = 16 bits
PA Power        = PA_MAX

Now sending 612411052...ok...Got response 612411052, round-trip delay: 41
Now sending 612412096...ok...Got response 612412096, round-trip delay: 40
Now sending 612413139...ok...Got response 612413139, round-trip delay: 40
Now sending 612414182...ok...Got response 612414182, round-trip delay: 41

```

```

192.168.0.10 - PuTTY
root@raspberrypi:/home/pi/EDU-Pack/rf24pi/examples# dir
Makefile pingtest.cpp pongtest.cpp rpi-hub.cpp scanner.cpp sendto_hub.cpp
pingtest pongtest rpi-hub scanner sendto_hub
root@raspberrypi:/home/pi/EDU-Pack/rf24pi/examples# ./pongtest

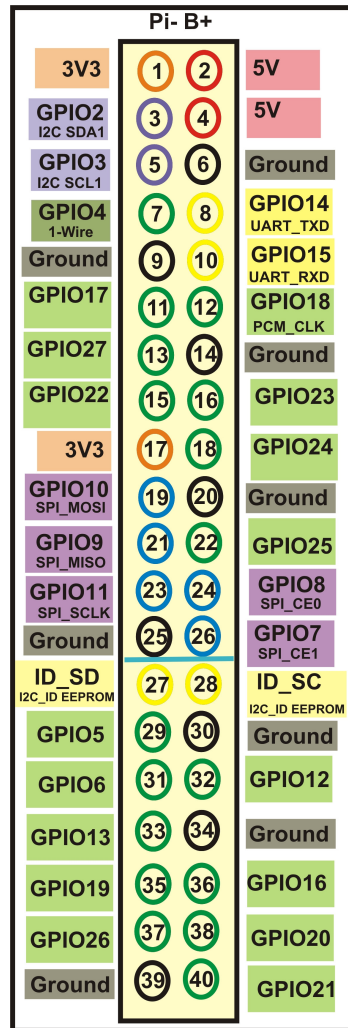
RF24/examples/pingpair/
ROLE: Pong back
SPI device      = /dev/spidev0.0
SPI speed       = 8000000
CE GPIO  = 25
STATUS          = 0x0e RX_DR=0 TX_DS=0 MAX_RT=0 RX_P_NO=7 TX_FULL=0
RX_ADDR_P0-1    = 0xf0f0f0f0d2 0xf0f0f0f0e1
RX_ADDR_P2-5    = 0xc3 0xc4 0xc5 0xc6
TX_ADDR         = 0xf0f0f0f0d2
RX_PW_P0-6      = 0x20 0x20 0x00 0x00 0x00 0x00
EN_AA           = 0x00
EN_RXADDR       = 0x02
RF_CH           = 0x4c
RF_SETUP        = 0x03
CONFIG          = 0x0f
DYNPD/FEATURE   = 0x00 0x00
Data Rate       = 1Mbps
Model           = nRF24L01+
CRC Length      = 16 bits
PA Power        = PA_LOW

Got payload 852646...Sent response.
Got payload 853694...Sent response.
Got payload 854742...Sent response.
Got payload 855789...Sent response.

```

7. Example 01- Setting GPIO as output

1. define GPIO 5,6,12,13,16,19,20,21,26 as output
2. send GPIO 5,6,12,13,16,19,20,21,26 LED "ON"
3. send GPIO 6,13,16,20,26 LED "OFF" & 5,12,19,21 LED "ON"
4. send GPIO 6,13,16,20 LED "ON" & 5,12,19,21 LED "OFF"



Python Test Code

In EDU-Pack "18B20" folder Type `python 18b20.py` or `python 18b20-1.py`

```

192.168.0.19 - PuTTY
root@raspberrypi:/home/pi/EDU-Pack# dir
18B20 nrf24scannerPi PIR rf24pi RFID Scratch-demo Step TrafficLights
root@raspberrypi:/home/pi/EDU-Pack# cd 18B20/
root@raspberrypi:/home/pi/EDU-Pack/18B20# python 18b20.py
(36.0, 96.8)
(35.687, 96.2366)
(35.375, 95.675)
(35.062, 95.1116)
^Z
[14]+ Stopped python 18b20.py
root@raspberrypi:/home/pi/EDU-Pack/18B20# python 18b20-1.py
(33.875, 92.975)
(33.687, 92.6366)
(33.437, 92.1866)
^Z
[15]+ Stopped python 18b20-1.py
root@raspberrypi:/home/pi/EDU-Pack/18B20#
  
```

In PIR folder type "python PIR.py"

```

192.168.0.19 - PuTTY
root@raspberrypi:/home/pi/EDU-Pack/PIR# dir
PIR.py
root@raspberrypi:/home/pi/EDU-Pack/PIR#python PIR.py
100.0
100.0
100.0
100.0
32.0
0.0
0.0
0.0
0.0
0.0
0.0
  
```

You can change GPIO23 to any GPIO you want use

```

GNU nano 2.2.6 File: PIR.py
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setup(23, 1)

Prev = []

while True:
    Sig = GPIO.input(23)
    if len(Prev) > 99:
        Avg = 0
        for A in Prev:
            Avg = Avg + A
        Avg = Avg / 100.00
        Prev = []
        print Avg * 100
    else:
        Prev.append(Sig)
        time.sleep(0.0003)
  
```

```
192.168.0.19 - PuTTY
root@raspberrypi:/home/pi/EDU-Pack/TrafficLights# cd ..
root@raspberrypi:/home/pi/EDU-Pack# dir
18B20  nrf24scannerPi  PIR  rf24pi  RFID  Scratch-demo  Step  TrafficLights
root@raspberrypi:/home/pi/EDU-Pack# cd Step
root@raspberrypi:/home/pi/EDU-Pack/Step# dir
2step-v2.py  step-v2.py
root@raspberrypi:/home/pi/EDU-Pack/Step# python step-v2.py
Delay between steps (milliseconds)?( 3 to 20 )10
How many steps forward? 200
How many steps backwards? 300
```

In "Step" folder python step-v2.py for 1 step motor
Type "python step-v2.py"

```
192.168.0.19 - PuTTY
root@raspberrypi:/home/pi/EDU-Pack/TrafficLights# cd ..
root@raspberrypi:/home/pi/EDU-Pack# dir
18B20  nrf24scannerPi  PIR  rf24pi  RFID  Scratch-demo  Step  TrafficLights
root@raspberrypi:/home/pi/EDU-Pack# cd TrafficLights/
root@raspberrypi:/home/pi/EDU-Pack/TrafficLights# python main.py
^2
```

In "TrafficLight" folder mina.py
Type "python main.py"

```
192.168.0.19 - PuTTY
^2root@raspberrypi:/home/pi/EDU-Pack/RFID# cd ..
root@raspberrypi:/home/pi/EDU-Pack# dir
18B20  nrf24scannerPi  PIR  rf24pi  RFID  Scratch-demo  Step  TrafficLights
root@raspberrypi:/home/pi/EDU-Pack# cd RFID/
root@raspberrypi:/home/pi/EDU-Pack/RFID# python RFID.py 0
Paths match, Continue.
Serial: ['324285b5']
Serial: ['324285b5']
Serial: ['53c652b6']
Serial: ['53c652b6']
```

In "RFID" folder RFID.py
Type "python RFID.py 0" for CEO

Pi_Scratch Install Q & A

1. When you type "python Pi_Scratch_v268.py -d"
You see following message

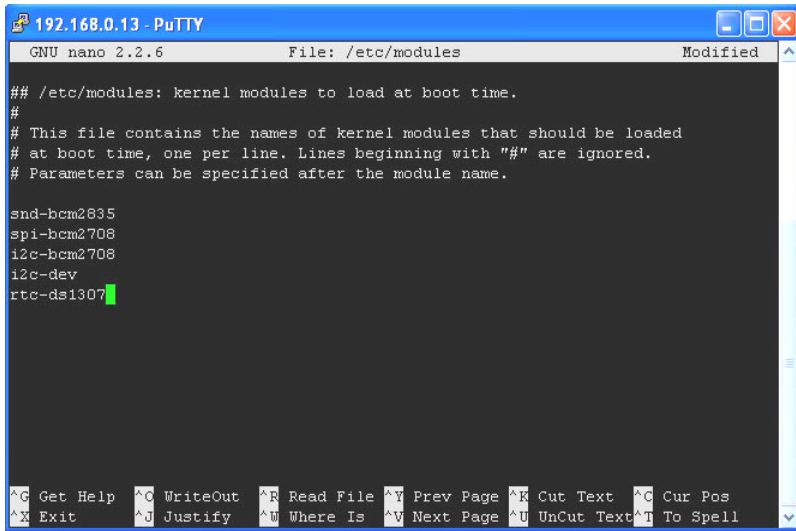
```
LXTerminal
File Edit Tabs Help
root@raspberrypi:/home/pi/Pi_Scratch_v268# python Pi_Scratch_v268.py -d
Traceback (most recent call last):
  File "Pi_Scratch_v268.py", line 45, in <module>
    import spidev
ImportError: No module named spidev
root@raspberrypi:/home/pi/Pi_Scratch_v268#
```

Type "python Install.py" install again

2. When you type "python Pi_Scratch_v268.py -d"
You see following message

```
LXTerminal
File Edit Tabs Help
root@raspberrypi:/home/pi/Pi_Scratch_v268# python Pi_Scratch_v268.py -d
Traceback (most recent call last):
  File "Pi_Scratch_v268.py", line 71, in <module>
    Bus = smbus.SMBus(1)
IOError: [Errno 2] No such file or directory
root@raspberrypi:/home/pi/Pi_Scratch_v268#
```

Check your setting in /etc/modules by use
sudo nano /etc/modules

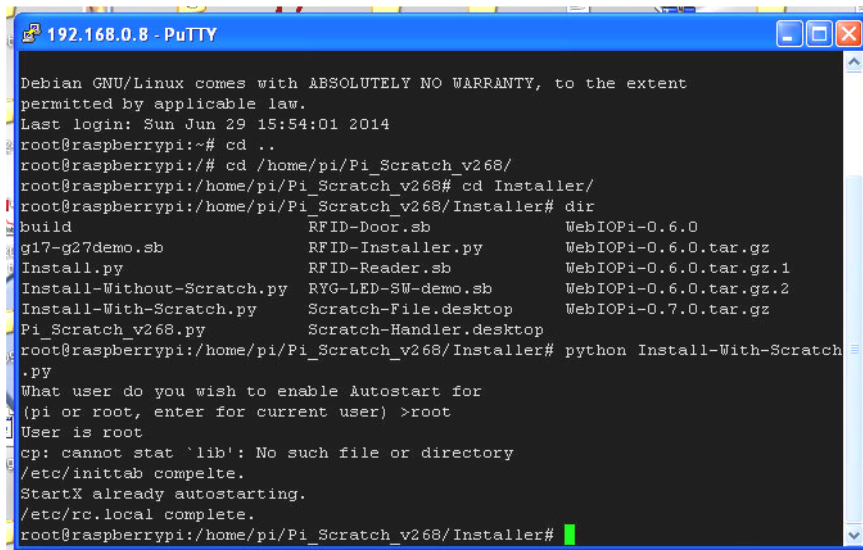


```
192.168.0.13 - PuTTY
GNU nano 2.2.6 File: /etc/modules Modified

## /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.

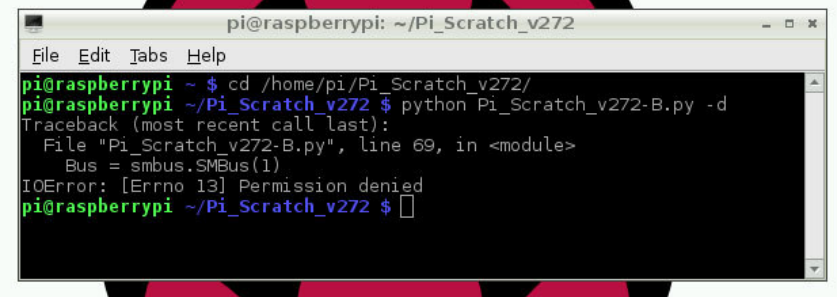
snd-bcm2835
spi-bcm2708
i2c-bcm2708
i2c-dev
rtc-ds1307
```

3. Use manually start Pi_Scratch first, after everything working fine, then setting Auto login and auto start , use root first



```
192.168.0.8 - PuTTY
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Jun 29 15:54:01 2014
root@raspberrypi:~# cd ..
root@raspberrypi:~/# cd /home/pi/Pi_Scratch_v268/
root@raspberrypi:/home/pi/Pi_Scratch_v268# cd Installer/
root@raspberrypi:/home/pi/Pi_Scratch_v268/Installer# dir
build                RFID-Door.sb         WebIOPi-0.6.0
gl7-g27demo.sb       RFID-Installer.py    WebIOPi-0.6.0.tar.gz
Install.py           RFID-Reader.sb       WebIOPi-0.6.0.tar.gz.1
Install-Without-Scratch.py RYG-LED-SW-demo.sb  WebIOPi-0.6.0.tar.gz.2
Install-With-Scratch.py Scratch-File.desktop WebIOPi-0.7.0.tar.gz
Pi_Scratch_v268.py   Scratch-Handler.desktop
root@raspberrypi:/home/pi/Pi_Scratch_v268/Installer# python Install-With-Scratch
.py
What user do you wish to enable Autostart for
(pi or root, enter for current user) >root
User is root
cp: cannot stat `lib': No such file or directory
/etc/inittab complete.
StartX already autostarting.
/etc/rc.local complete.
root@raspberrypi:/home/pi/Pi_Scratch_v268/Installer#
```

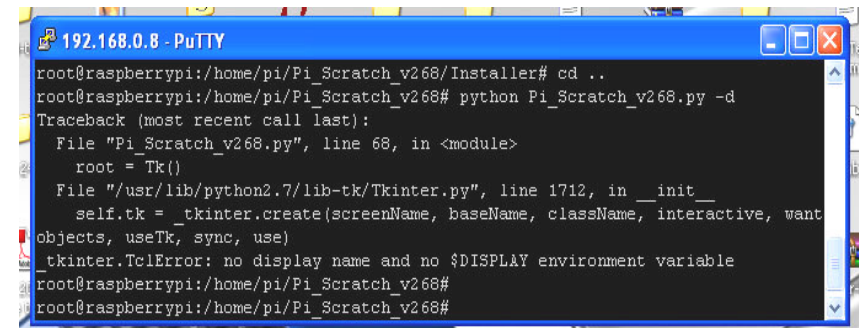
4. When you type “python Pi_Scratch_v272-B.py -d”
You see following message



```
pi@raspberrypi: ~/Pi_Scratch_v272
File Edit Tabs Help
pi@raspberrypi ~$ cd /home/pi/Pi_Scratch_v272/
pi@raspberrypi ~/Pi_Scratch_v272$ python Pi_Scratch_v272-B.py -d
Traceback (most recent call last):
  File "Pi_Scratch_v272-B.py", line 69, in <module>
    Bus = smbus.SMBus(1)
IOError: [Errno 13] Permission denied
pi@raspberrypi ~/Pi_Scratch_v272$
```

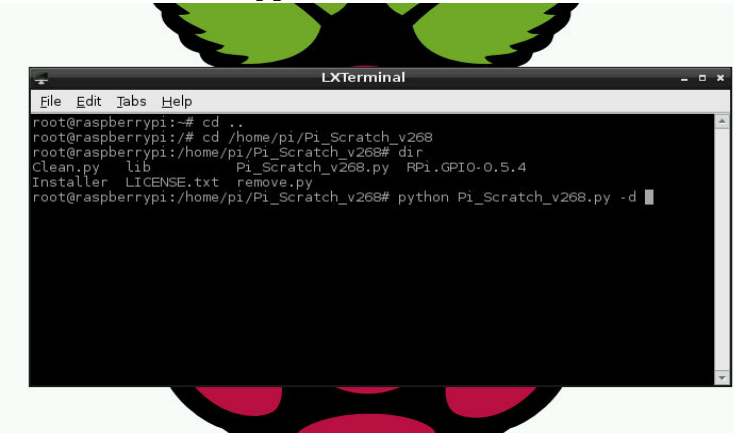
Use “root” to login your Pi, because some of the software function need access your Pi

5. When you type “python Pi_Scratch_v268.py -d”
You see following message



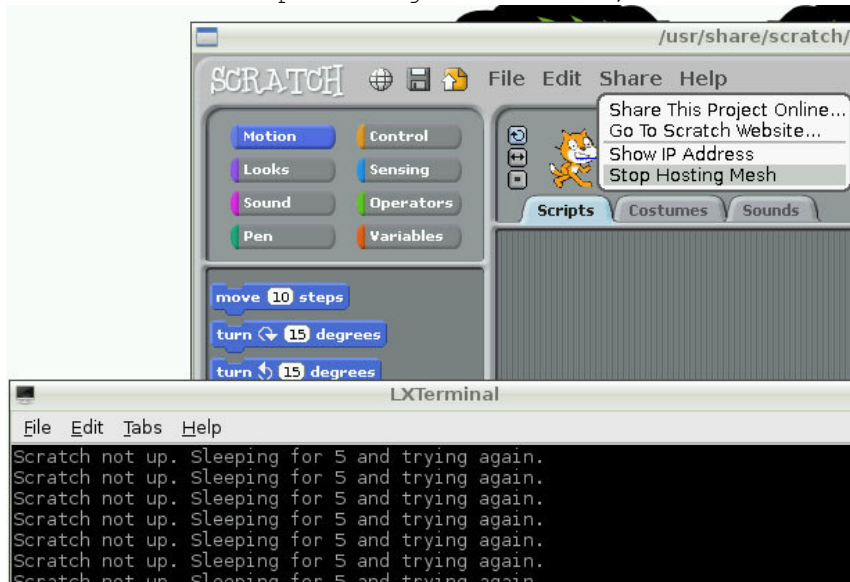
```
192.168.0.8 - PuTTY
root@raspberrypi:/home/pi/Pi_Scratch_v268/Installer# cd ..
root@raspberrypi:/home/pi/Pi_Scratch_v268# python Pi_Scratch_v268.py -d
Traceback (most recent call last):
  File "Pi_Scratch_v268.py", line 68, in <module>
    root = Tk()
  File "/usr/lib/python2.7/lib-tk/Tkinter.py", line 1712, in __init__
    self.tk = _tkinter.create(screenName, baseName, className, interactive, want
objects, useTk, sync, use)
_tkinter.TclError: no display name and no $DISPLAY environment variable
root@raspberrypi:/home/pi/Pi_Scratch_v268#
root@raspberrypi:/home/pi/Pi_Scratch_v268#
```

Type “startx” first, the open “LXTerminal” type “python Pi_Scratch_v268.py -d”

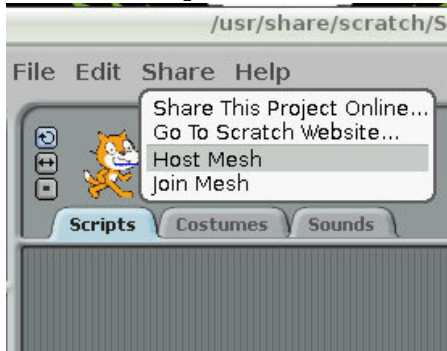


```
LXTerminal
File Edit Tabs Help
root@raspberrypi:~# cd ..
root@raspberrypi:~/# cd /home/pi/Pi_Scratch_v268
root@raspberrypi:/home/pi/Pi_Scratch_v268# dir
Clean.py  lib          Pi_Scratch_v268.py  RPi.GPIO-0.5.4
Installer LICENSE.txt remove.py
root@raspberrypi:/home/pi/Pi_Scratch_v268# python Pi_Scratch_v268.py -d
```

6. When Manually start Pi_Scratch if you can't see "Host Mesh" Choose "Stop Hosting Mesh" first,



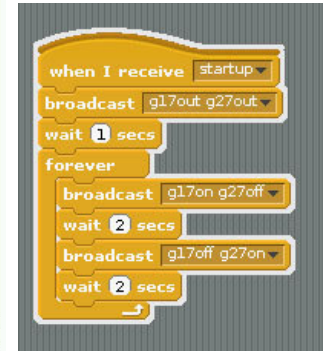
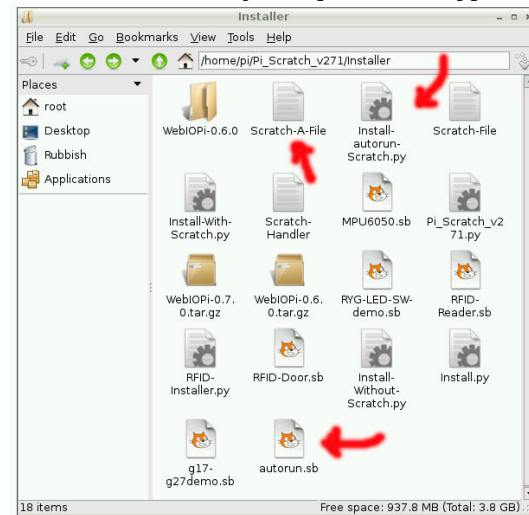
"shift S" you will see "Host Mesh"



How to auto run Pi_Scratch without press any key

Type "python Install-autorun-Scratch.py"

Will do the setting for you also copy the autorun.sb example file



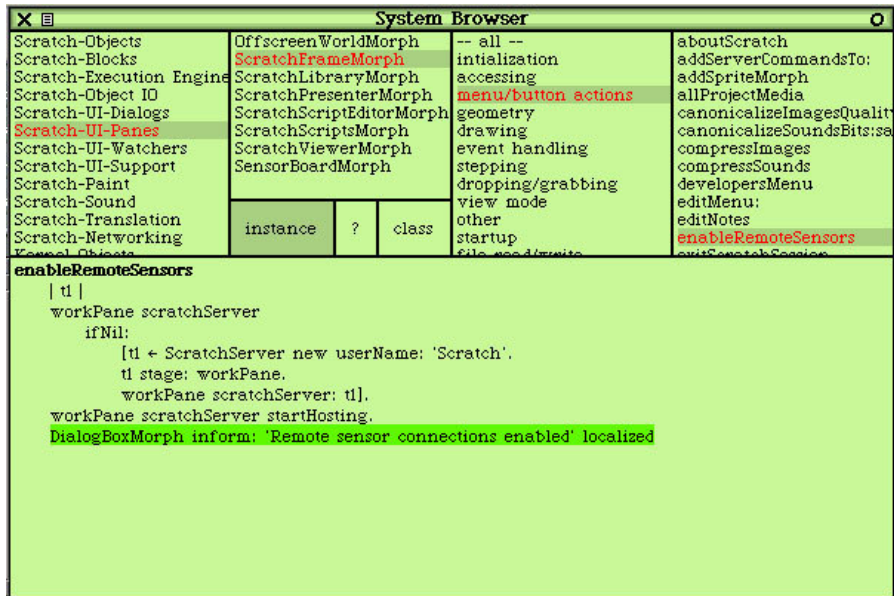
Use Pi_Scratch_V271

When receive "startup" to active auto run example file

We now have the program auto run in presentation mode, but as mentioned earlier there is a pop-up message **"Remote sensor connections enabled"** that stop update information for sensor " it will update until you have clicked OK. This is something that needs to be turned off within the Scratch application, which is one of the hidden settings within Scratch.

After launching Scratch shift left-click on the R in the Scratch logo (top left) and choose "turn fill screen off". Click on the white area to the right or bottom of the screen and choose "Open" and "Browser". Navigate to the section Scratch-UI-Panes -> ScratchFrameMorph -> menu/button actions -> enableRemoteSensors and remove the line:

Dialogboxmorph Inform: 'Remote Sensor Connections Enabled' Localized

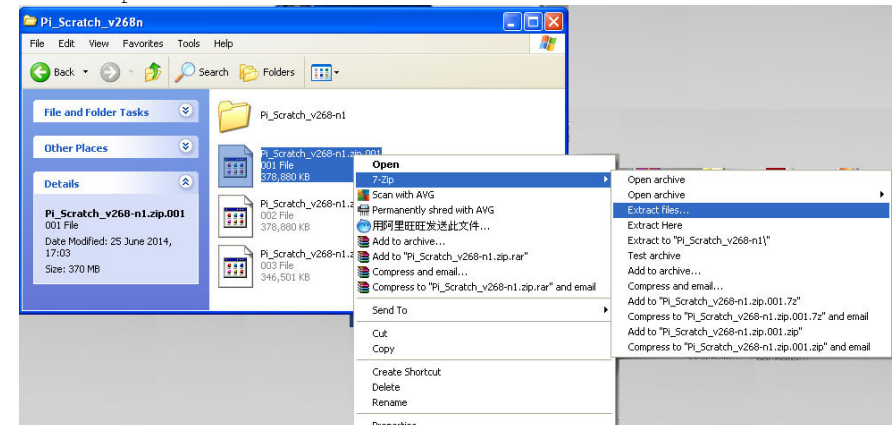


Then click with the centre mouse button (or Ctrl and left button) and choose Accept (s).

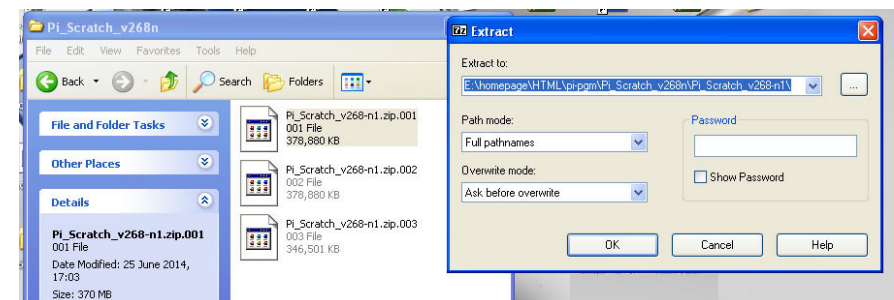
You can now remove the white space in Scratch by shift left click on the R in Scratch logo and choose Turn Fill screen on. Finally shift left click on the logo again and choose **"save image for end-user"**

How to use three zip file change to Image file

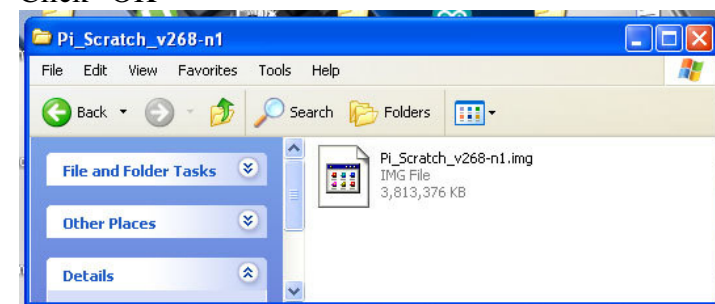
Use 7-Zip software



Copy three zip file into one new folder, use 7-zip, click right mouse key choose "Extract files"



Click "OK"



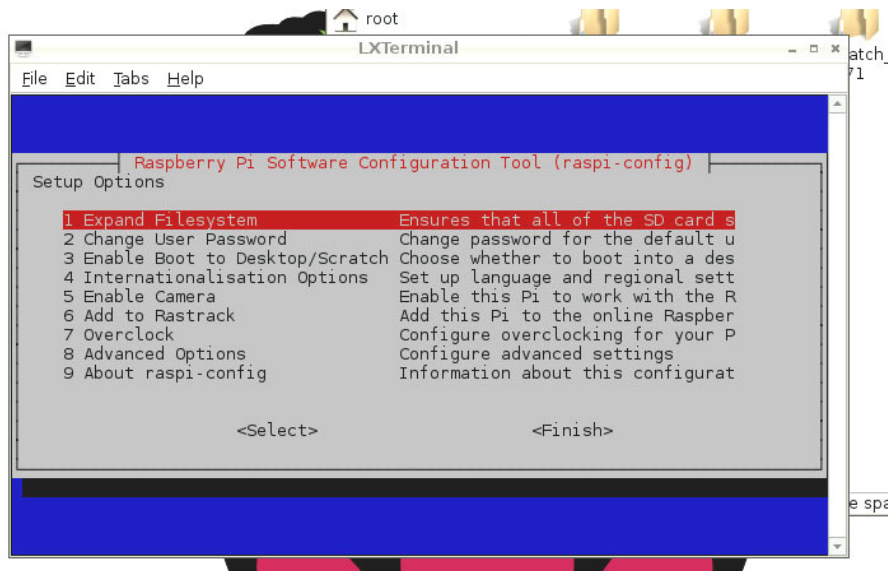
After finished, you will have Pi_Scratch-V268-n1.img

Upzip 3 zip file for Linux

- 1) Download all the files to the same directory such as /home/pi/Scratch
- 2) If they are named in the same format as Pi_Scratch.... then you can run "cat Pi_Scratch* >> merged.zip" or the name of the files minus the zip file extension & 001, 002 etc.
- 3) run "unzip -FF merged.zip" to unzip the image.
- 4) run the DD command to copy the image to another storage unit (SD card, USB drive etc.)

If either one of the archives has a missing part / sector then the unzip will fail.

First thing need do for Pi_Scratch ready Image



Use the Pi_Scratch ready Image first thing you need do

sudo raspi-config

Choose 1 "expand Filesystem to Ensures that all of the SD card s

How to upgrade from old Pi_Scratch to new Ver Pi_Scratch

In old ver Pi_Scratch_268 folder --- remove auto-login auto run first
sudo python remove.py

then reboot

download new Ver Pi_Scratch

sudo wget http://pridopia.co.uk/pi-pgm/Pi_Scratch_v272.tar

tar xf Pi_Scratch_v272.tar

cd Pi_Scratch_v272

cd Installer

sudo python Install-autorun-Scratch.py --- auto run example

without press any key - choose root

or

sudo python Install-with-Scratch.py --- auto load example but

need click "ok" and "greeng flag" in scratch - choose root

Pi_Scratch-v272 ready Image information

Use Raspbian Wheezy 2014-06-20

ready to use Pi_Scratch_v272 with Auto Login & Auto Load Pi_Scratch
& SSH enable --

SSH enable -- user - root password : pi for 4GB SD card

TightVNCserver install password : pipipi --

Use **sudo vncpasswd** to change password

VNC (VIRTUAL NETWORK COMPUTING)

VNC is a graphical desktop sharing system that allows you to remotely control the desktop interface of one computer from another. It transmits the keyboard and mouse events from the controller, and receives updates to the screen over the network from the remote host. You will see the desktop of the Raspberry Pi inside a window on your computer. You'll be able to control it as though you were working on the Raspberry Pi itself. Type **tightvncserver** to start VNC control from PC Mac or iPad

```
192.168.0.8 - PuTTY
login as: root
root@192.168.0.8's password:
Linux raspberrypi 3.12.22+ #691 PREEMPT Wed Jun 18 18:29:58 BST 2014 armv6l

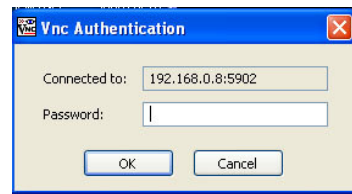
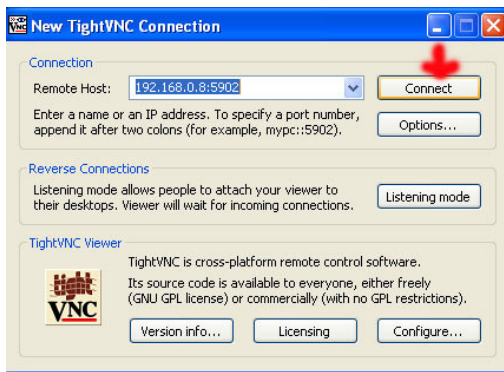
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Aug 1 09:09:35 2014 from 192.168.0.2
root@raspberrypi:~# tightvncserver

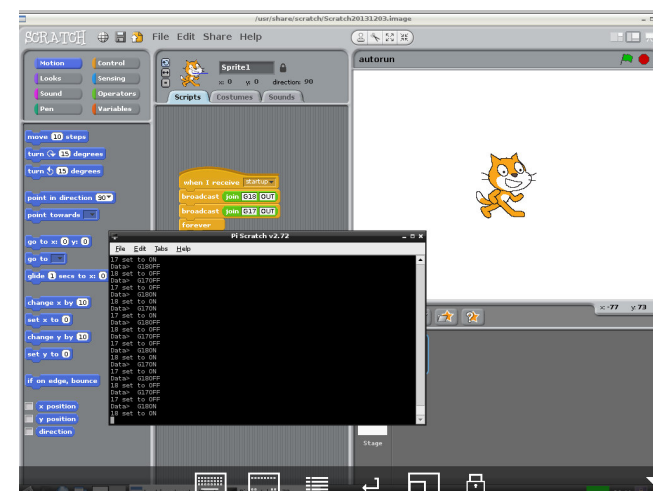
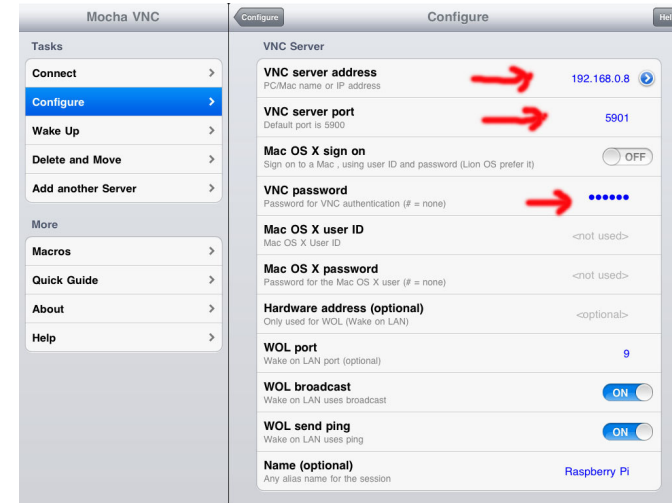
New 'X' desktop is raspberrypi:1

Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/raspberrypi:1.log

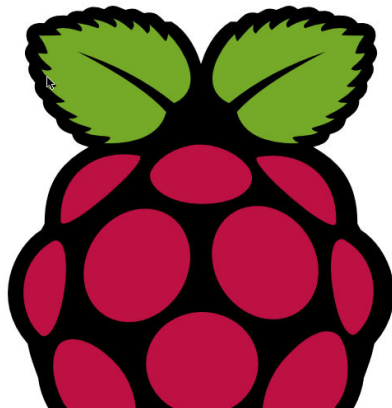
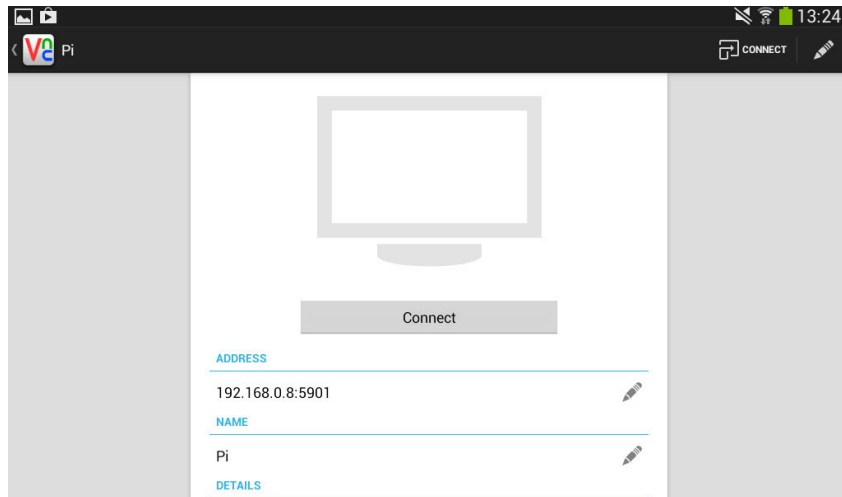
root@raspberrypi:~# sudo vncpasswd
Using password file /root/.vnc/passwd
Password:
Verify:
Would you like to enter a view-only password (y/n)? n
root@raspberrypi:~#
```



VNC connect from iPad & idevice

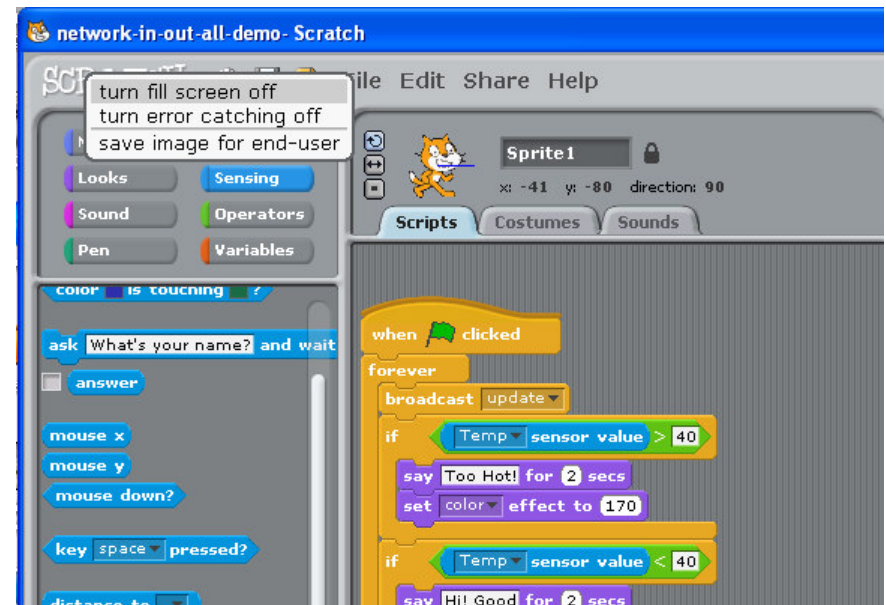


VNC connect from Android Smart Phone / Tablet



1. Scratch V1.4 Setting to Active Mesh function

1. Shift-click the upper part of the R in the Scratch logo.

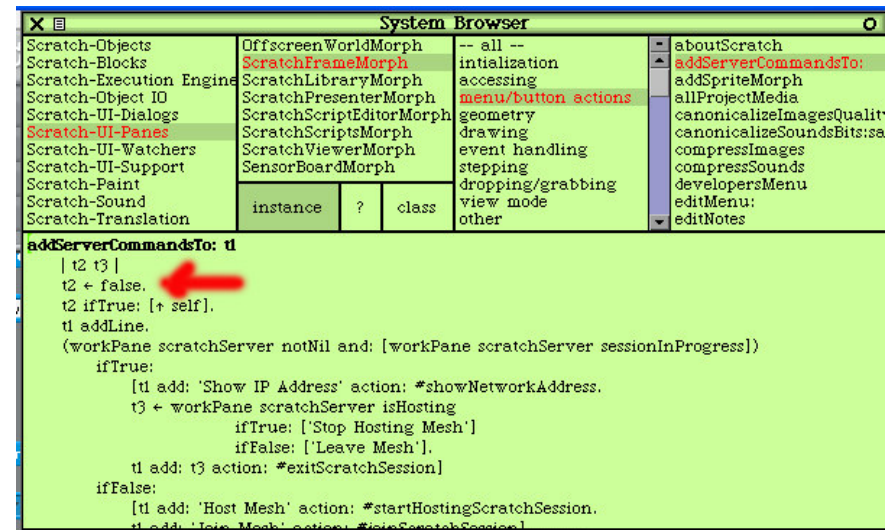
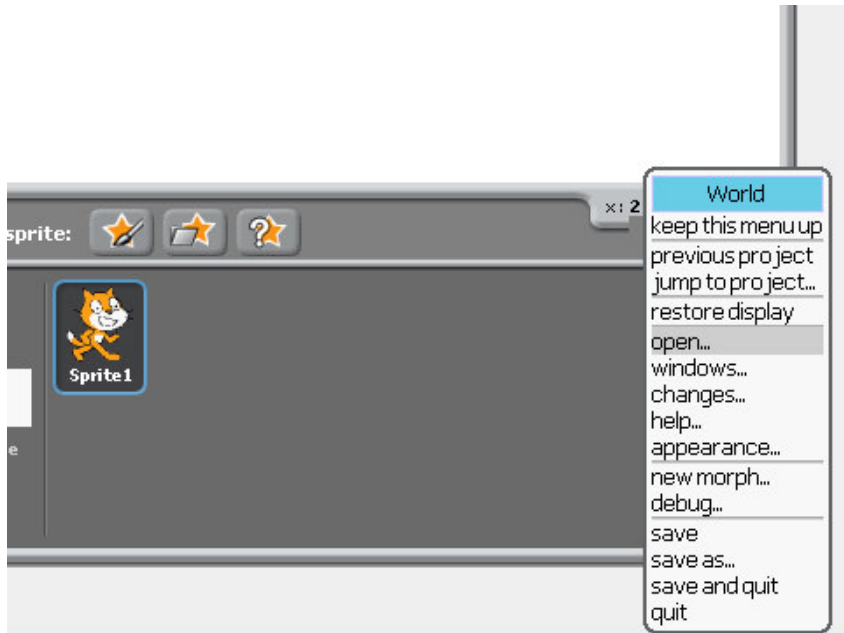


2. Select "turn fill screen off".

3. You will now have a white area at the bottom and right side of the screen. Click on the white area and select "open...".

4. Now select "browser".

5. Now you've opened the System Browser! In this place you can edit Scratch.



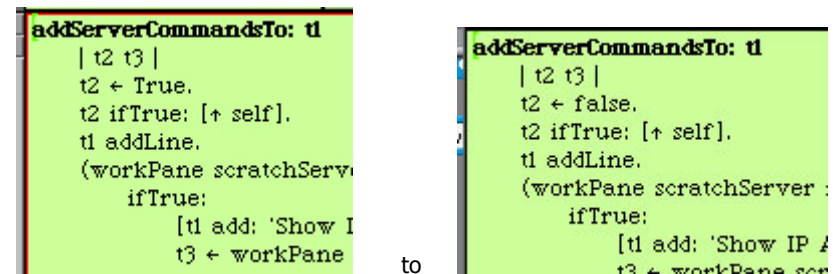
5. Now you've opened the System Browser! In this place you can edit Scratch.

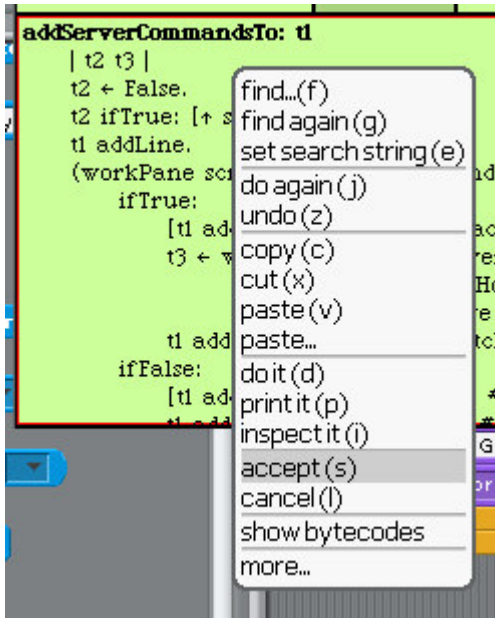
Go to Scratch-UI-Panes - ScratchFrameMorph - menu/button actions - addServerCommandsTo: .

6. Look down, and edit the "t2 <- true" to "t2 <- false".

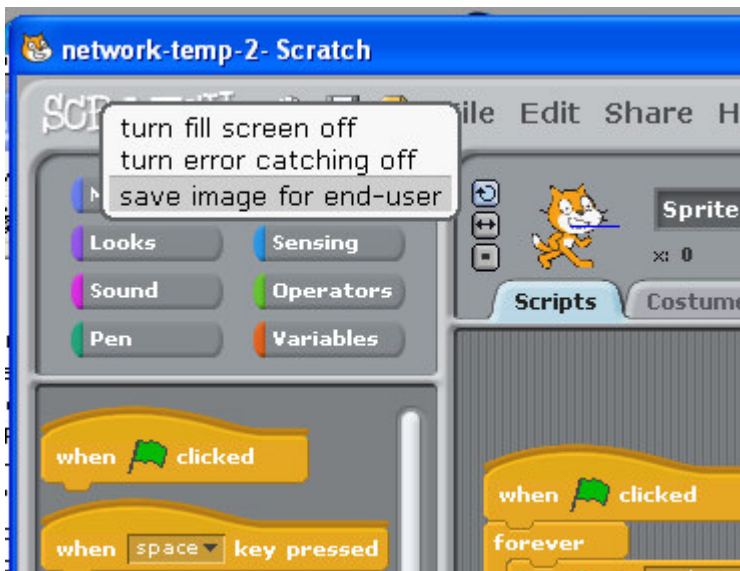
7. Right-click and select "accept".

(or press middle key or wheel)





8. Enter in your initials (you can fake it, it doesn't matter) and click "accept".
9. Exit the System Browser.
10. Shift-click the R and select "turn fill screen on".



If you don't want to go through the process every time you want to use Mesh, you can save the System Browser so Mesh will stay available - Shift-click the R again and select "save image for end-user".

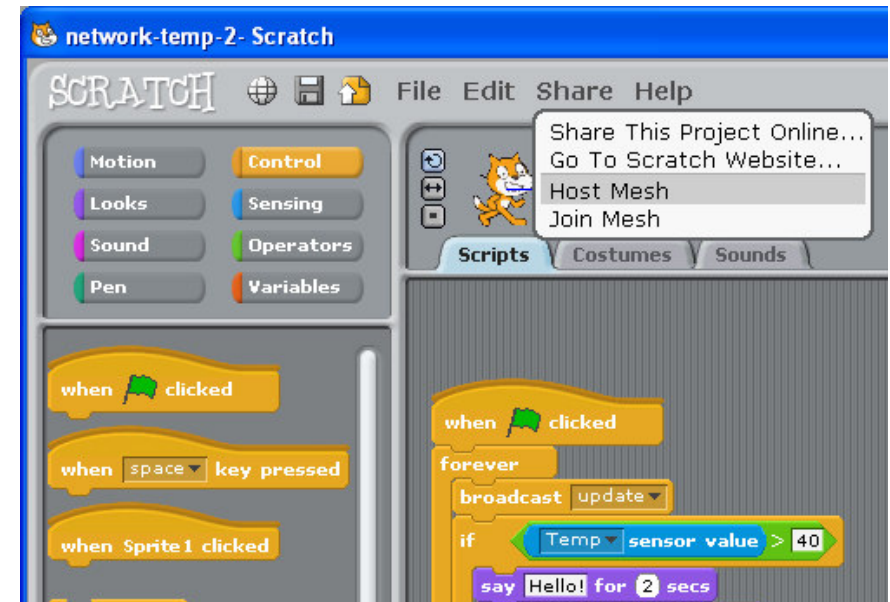
If you don't want Mesh anymore, just change the 'false' back to 'true'.

Active Mesh

- Host Mesh lets you host a Mesh session.
- Join Mesh lets you join a Mesh session.

To host a Mesh session:

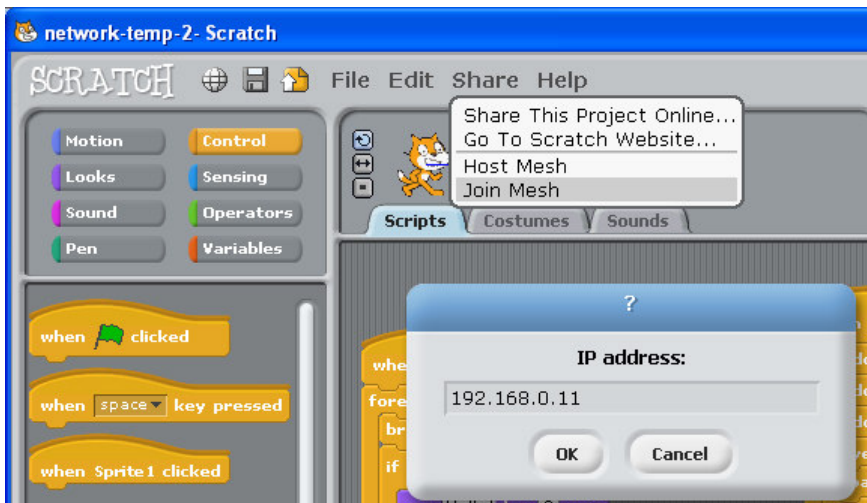
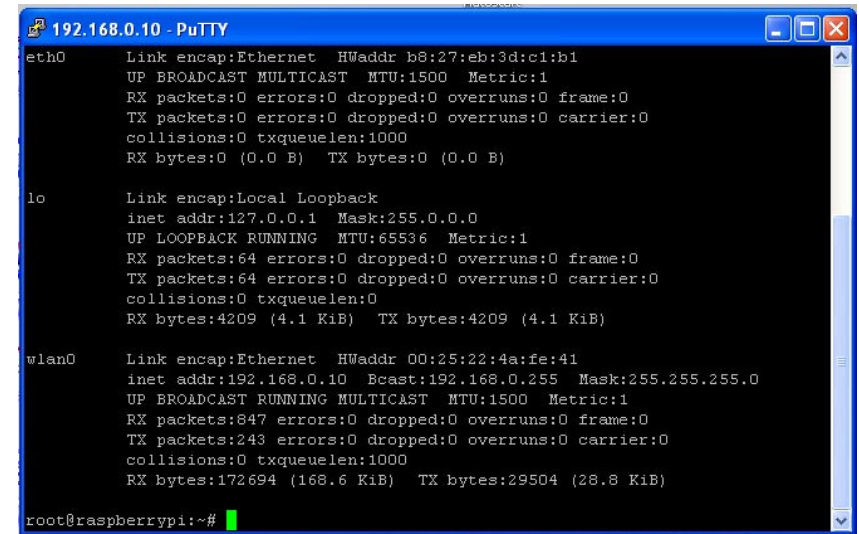
1. Click Host Mesh.
2. Done! Your IP address will show up



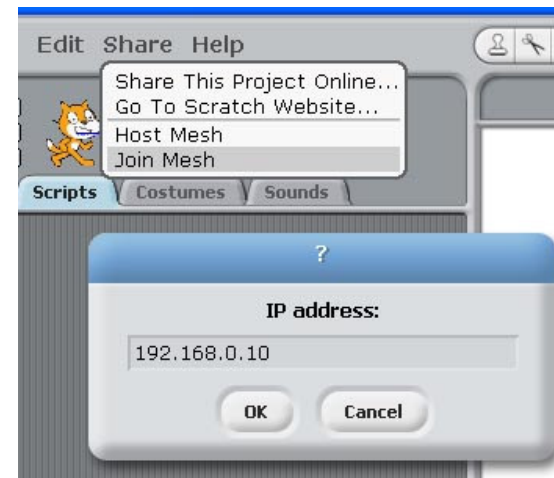
What if you want to join a Mesh session, but the person hosting the Mesh session forgot to tell you the IP address, and he/she forgot what the IP address is? He/she Shift-clicks the Share button and selects "Show IP address". The IP address will pop up like before.



Use "ifconfig" check the IP of Raspberry Pi



Join Mesh "input Raspberry Pi IP" login from Desktop or Laptop



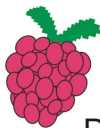
To join a Mesh session:

1. Click Join Mesh.
2. Enter in the IP address of the Raspberry Pi that is hosting the Mesh session. (use "ifconfig" in terminal command line)

A Mesh session can have more than two Scratch programs connected at once - there's no limit! All Scratch programs have to connect to the same IP though.



<http://www.pridopia.co.uk/ixx-rspi.html>



Find all your needs for Raspberry Pi in one place - Pridopia

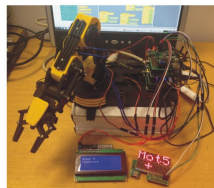
DC Motor Series



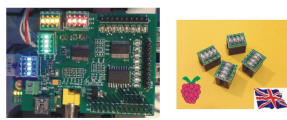
Rs-Pi-L293D-3
6 Motor Board



Rs-Pi-L293D-1
2 Motor Board



Rs-Pi-L293D-3 6 Motor-
i2c 20x4 LCD &
24 x16 LED Matrix Demo



PWM LED set demo
(Red, Green, Yellow, Blue)

RFID Reader



Rs-Pi-RFID Reader
& Relay + Step Motor
control



Rs-Pi-4 Relay
& Step Motor Board

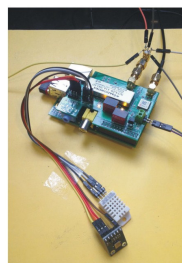


Rs-Pi- LED 24x16
Matrix Board

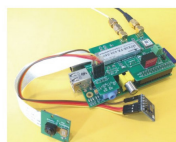


Rs-Pi- LED 16x16
Matrix Board

LED Matrix



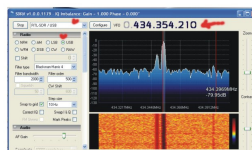
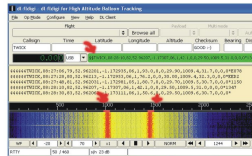
Rs-Pi-P-HAB 5V Board
(High-Altitude Balloon)



Rs-Pi-P-HAB 3.3V Board
(High-Altitude Balloon)



Rs-Pi-P-HAB & GPS Antenna
FM Antenna &
USB SDR Encoder



Pridopia Limited
New Brook House
385 Alfreton Road
Nottingham NG7 5LR
United Kingdom
TEL: 0844-2492899



Software for Scratch

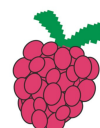
<http://www.pridopia.co.uk/rs-pi-set-scratch.html>

Make Raspberry Pi More Easy !

Raspberry Pi is a trademark of the Raspberry Pi Foundation



<http://www.pridopia.co.uk/ixx-rspi.html>



Find all your needs for Raspberry Pi in one place - Pridopia

4 USB Hub Series



Rs-Pi-4 USB Hub +
Multi-function i2c Board



Rs-Pi-4 USB Hub +
32 GPIO(i2c/SPI) Board

7 USB Hub Series



Rs-Pi-7 USB Hub +
Multi-function i2c Board



Rs-Pi-7 USB Hub +
16 GPIO (i2c/SPI) Board

PWM / Servo Series



Rs-Pi-32 channel
PWM/Servo Board



Rs-Pi-16 channel PWM
& 16 GPIO (i2c/SPI) Board



Rs-Pi-16 channel PWM
& 4 Motor - PSU Board



Rs-Pi- uln2803
SPI AD/DA -1-Wire
Multi-function Board



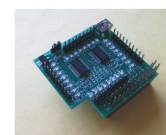
Rs-Pi- 2 Relay +
Step Motor Board

Relay & Motor Series

i2c & SPI GPIO Board



Rs-Pi-Multi-function
32 GPIO (i2c/SPI) Board



Rs-Pi-32 GPIO
(i2c/SPI) Board



Rs-Pi-64 GPIO
(i2c/SPI) Board



Rs-Pi-128 GPIO
(i2c/SPI) Board



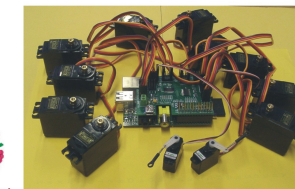
Rs-Pi- LED 24x16
Matrix Board



Rs-Pi-LCD-Master
Board (84x48, 16x2, 20x4)



Rs-Pi- RS232
console cable



Pridopia Limited
New Brook House
385 Alfreton Road
Nottingham NG7 5LR
United Kingdom
TEL: 0844-2492899



Software for Scratch

<http://www.pridopia.co.uk/rs-pi-set-scratch.html>

Make Raspberry Pi More Easy !

Raspberry Pi is a trademark of the Raspberry Pi Foundation