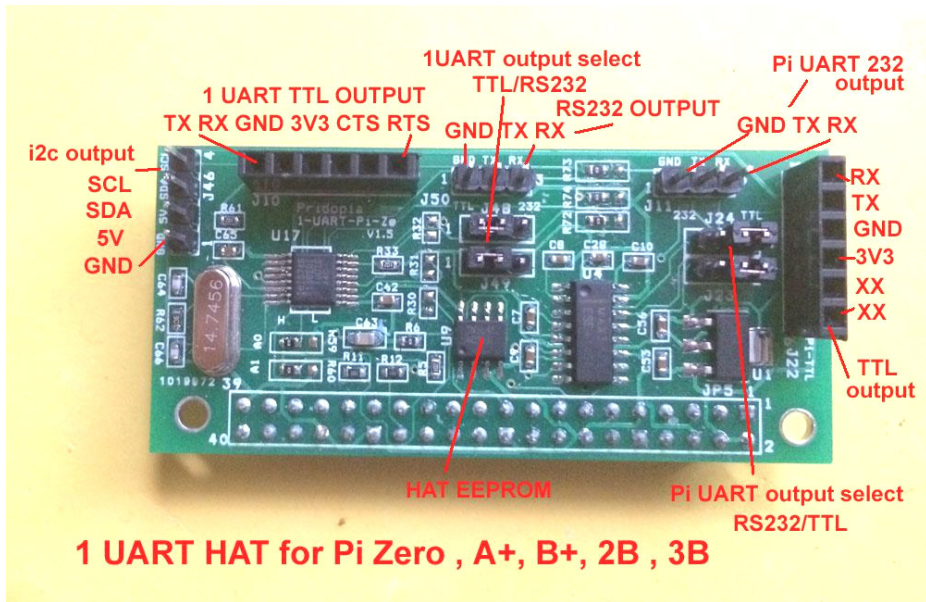


# Raspberry Pi - extra 1 UART User Manual



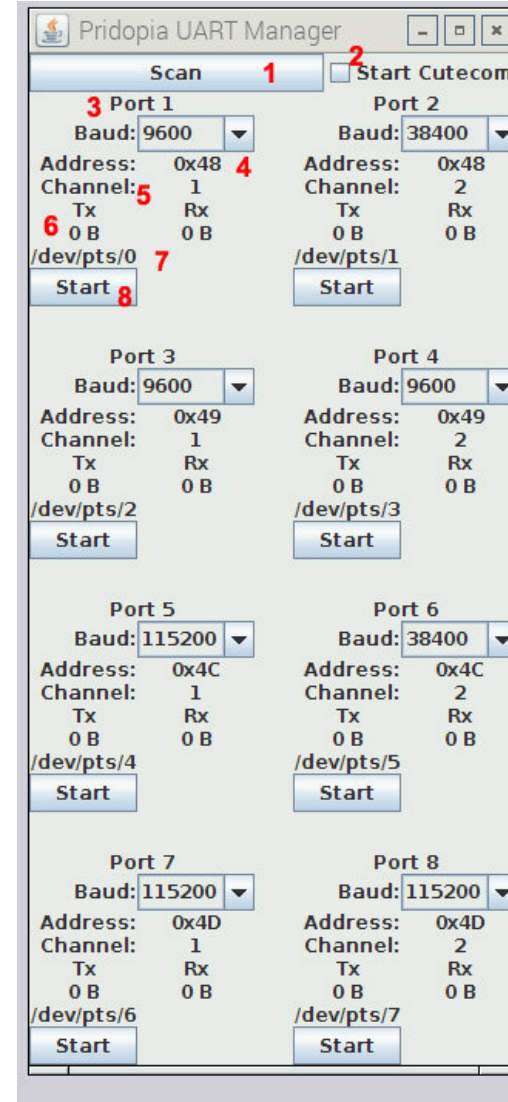
1. provide extra 1 UART TTL output
  - 1x UART TTL output TX,RX,GND,3V3,CTS,RTS
  - 1xUART 232 output GND,TX,RX
  - TTL/RS232 output select
2. Pi UART can select TTL/232 output
  - Pi UART RS232 output GND, TX, RX
  - Pi UART TTL output TX,RX,GND,3V3,XX,XX
3. i2c signal out GND,5V,SDA,SCL for extend extra UART HAT
4. It provides an individual 3.3V power for UART & output port
5. Provide HAT EEPROM for future driver program update.

## Configuration

There are two mini jumpers (0 ohm resistor) to configure each chip for one of the following 7-bit I2C addresses:

0x48	A0 - H	A1 - H	0x49	A0 - L	A1 - H
0x4C	A0 - H	A1 - L	0x4D	A0 - L	A1 - L

GUI Java Base Serial port control panel



- 1. Scan: Scan for available UART devices.
- 2. Launch cutecom together when starting Serial Port.
- 3. Port number in system
- 4. Baud Rate choose
- 5.6.7. UART Port Information: Port Address 0x48,0x49, 0x4c, 0x4d  
Chip Address Location, Channel Number ( 1 / 2 ), TX and RX receive counter, and Serial port Location /dev/pts/X
- 8. Start : Start UART port

Provided is a terminal command mode to activate serial ports through a terminal / ssh.

You can also activate a serial port on boot up with the provided software.

Commands for starting the serial ports via command line;

```
./Pi8s -a ADDR -c CHANNEL -b BAUD -s DEVPTS  
ADDR = Address of Serial Port e.g. 0x48, 0x49, 0x4C,  
and 0x4D
```

CHANNEL = Channel of serial Port ( 1 / 2 )

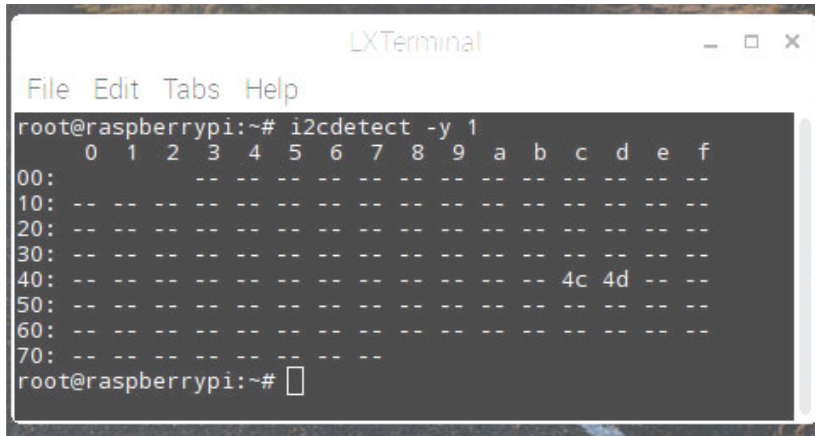
BAUD = Baud Rate for the Serial Port

DEVPTS = PICOM Service Number ( Will direct the filename to /dev/picomX where X = your number )

**For example, To start a serial port from 0x49, on channel 2 with 115200 Baud run the following;**

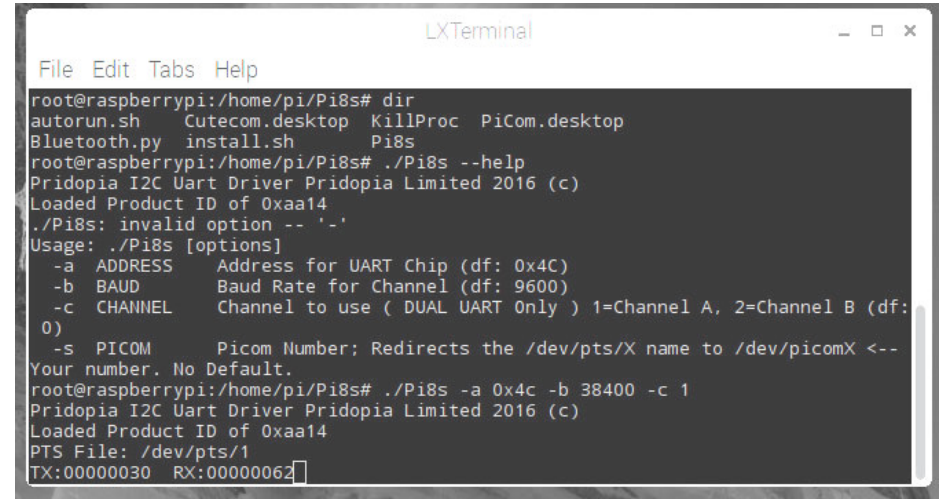
```
./Pi8s -a 0x49 -c 2 -b 115200 -s 1
```

This will activate the serial port and then set /dev/picom1 to your serial port location /dev/pts/X

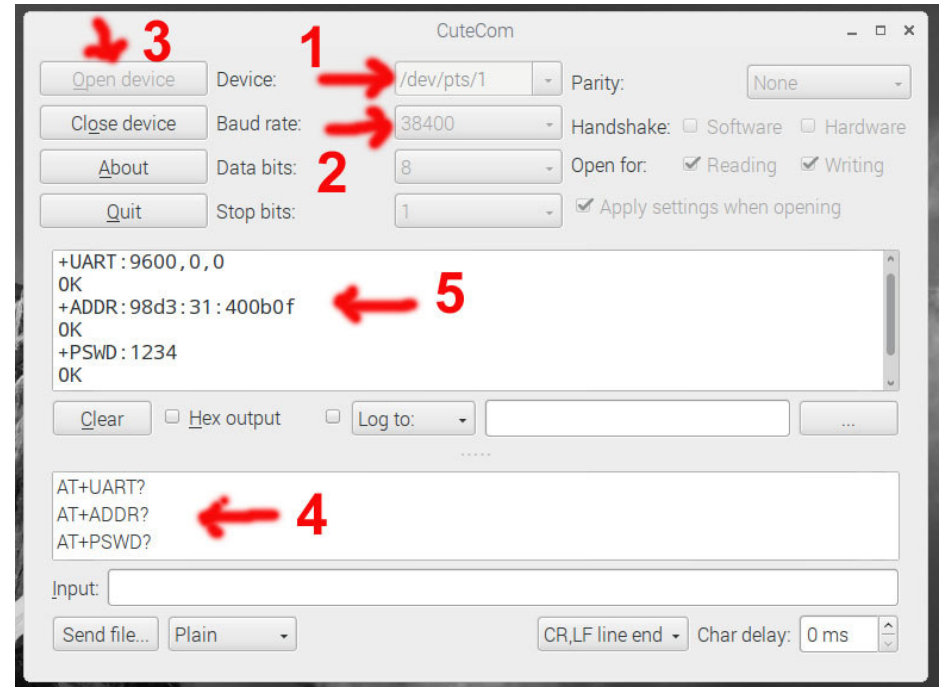


```
LXTerminal  
File Edit Tabs Help  
root@raspberrypi:~# i2cdetect -y 1  
 0 1 2 3 4 5 6 7 8 9 a b c d e f  
00: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
40: -- -- -- -- -- -- -- -- -- -- 4c 4d -- -- --  
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
70: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
root@raspberrypi:~#
```

Use i2cdetect check the 4xUART address



```
LXTerminal  
File Edit Tabs Help  
root@raspberrypi:/home/pi/Pi8s# dir  
autorun.sh      Cutecom.desktop KillProc PiCom.desktop  
Bluetooth.py    install.sh      Pi8s  
root@raspberrypi:/home/pi/Pi8s# ./Pi8s --help  
Pridopia I2C Uart Driver Pridopia Limited 2016 (c)  
Loaded Product ID of 0xaa14  
./Pi8s: invalid option -- '-'  
Usage: ./Pi8s [options]  
-a ADDRESS      Address for UART Chip (df: 0x4C)  
-b BAUD         Baud Rate for Channel (df: 9600)  
-c CHANNEL      Channel to use ( DUAL UART Only ) 1=Channel A, 2=Channel B (df:  
0)  
-s PICOM        Picom Number; Redirects the /dev/pts/X name to /dev/picomX <--  
Your number. No Default.  
root@raspberrypi:/home/pi/Pi8s# ./Pi8s -a 0x4c -b 38400 -c 1  
Pridopia I2C Uart Driver Pridopia Limited 2016 (c)  
Loaded Product ID of 0xaa14  
PTS File: /dev/pts/1  
TX:00000030 RX:00000062
```



1. device -> /dev/pts/1
2. Baud rate -> 38400
3. click Open device
4. input AT command (HC-05 bluetooth module)
5. screen will show Bluetooth module response

To begin using the external UART ports on the 1xUART, 2xUART, and 4xUART. A few details are to be noted;

\* All UART Locations are marked in /dev/picomX where X is 1, 2, 3, or 4 depending on the amount of UARTs available.

\* /dev/picom1 for example is for the 1st UART, if you have 2xUART you can find your ports at /dev/picom1, and /dev/picom2 and so on so forth for 4xUART

\* /dev/picom is only a file to store the location of the actual Port NOT the UART port location. /dev/picom1 will output the location e.g. /dev/pts/9

\* When a UART port is closed or has an error which causes it to restart / reboot the port the location MIGHT change

\* Poll your /dev/picom numbers when you come across an error, or are opening up a new UART Port.

\* The pts numbers are usually /dev/pts1, /dev/pts2, /dev/pts/3 etc.

### Starting your UART Ports: ( SKIP if using GUI )

To start up the UART port, You need to know which board you have ( 1, 2, or 4xUART ). Once you know this you can check the table below for information about the ports.

You should have a program called "Pi8s" somewhere in the folder provided. You need to install this as this communicates with the board to perform serial functions.

To install the software, you need to run the command script ./install.sh to install the necessary software. After this is complete you can use the software straight away

but you might want to run the autorun.sh script as this will automatically start the Pi8s software on boot. It will ask a few questions about what board ( to get the addresses

right ), and once that is complete it will auto-launch the Pi8s software for all your ports. After a reboot you will be able to find the UART port locations at;

Port 1: /dev/picom1

Port 2: /dev/picom2

Port 3: /dev/picom3

Port 4: /dev/picom4

1xUART = Port 1.

2xUART = Port 1 & Port 2.

4xUART = Port 1, 2, 3, and 4.

### Starting the GUI:

There are multiple ways to start the GUI for the Pi8s software. Below are the different ways you can achieve this;

1) Go to the folder JavaUART and launch Launch.sh by running "./Launch.sh"

( Without quotes )

2) This will start up the GUI for you, Done.

A) Go to the folder JavaUART, enter the folder "jar"

B) Run command "java PiCom"

C) Done, GUI will run

I) Run the Desktop shortcut called JavaUART.

II) This will also run the GUI very simply.

----- UART Board Specifications -----

UART Board Selection : Addresses : Channels :

1xUART : 0x48 : 1 : 2xUART : 0x49 : 2 :

4xUART : 0x4C, 0x4D : 2 :

\*\*\*\* increase i2c bus speed to prevent data lost

**sudo nano /boot/config.txt**

add a line in end of file

**dtoverlay=i2c1\_baudrate=1000000**

( you can test the value from 400000 to 1000000 in your system.

## Communicating with the UART ports:

Once you've started up the Pi8s Software, ran the GUI, or auto-started the software using the `./autorun.sh` command, You should be ready to go. You can poll the file `/dev/picom1` to check if you have a `/dev/ptsx` address. If not this means that your UART has failed to start, the software isn't running or the HAT hasn't been recognised. We recommend making sure the board is plugged in, check "ps aux" to see if the command "Pi8s" is running. Once you have confirmed you have a `/dev/ptsx` location then you can open cutecom or any other serial read/write software and open that `/dev/ptsx` location.

For instance if your `/dev/picom1` file has the location `"/dev/pts9"` inside the file you will use that to open the Serial port on cutecom.

### F.A.Q:

Q: I Keep getting the error [bash: ./Command.sh: Permission Denied]:

A: Run `chmod` on the file in question, if the file is called "Command.sh" run the following

```
chmod +x Command.sh
```

You should be able to run the command now.

Q: The output from the UART port is jibberish / data isn't ASCII / English:

A: You may have the incorrect Baud Rate set up, the default is 9600, but you may get a baud rate of 115200 / 19200 / 57600 for most devices.

If your baud rate is correct, make sure your serial device connected is properly grounded and is transmitting normal UART 8bit bytes at no parity, 1 stop bit ( Default 8n1 setting )

Q: My Serial software keeps telling me `/dev/picom1` isn't a serial port:

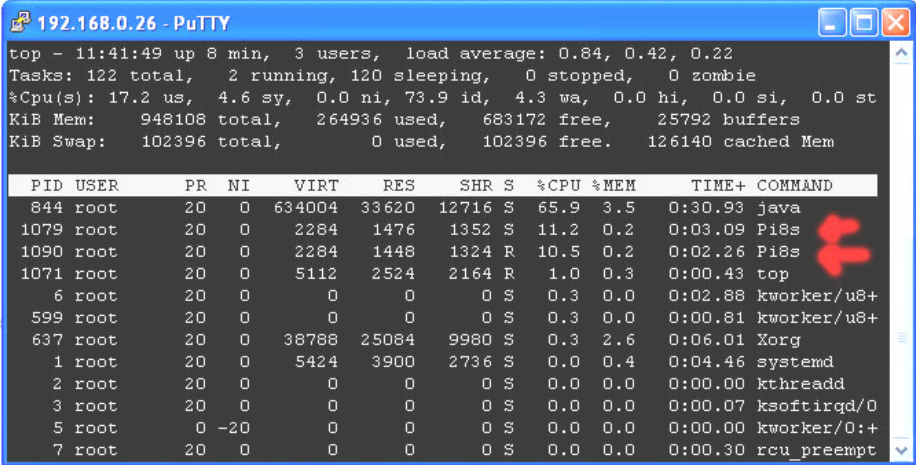
A: `/dev/picomX` isn't a serial port, its a file that stores the location of the serial port as a link. such as `/dev/pts/5`

Q: Can I have the source code for my own project?

A: No.

Q: How can I check Pi8s enable already ?

A: in Terminal windows , type **top** then press enter. If you enable a port you will find Pi8s run in process. for example enable 2port you will found 2 Pi8s run in processes.



```
top - 11:41:49 up 8 min, 3 users, load average: 0.84, 0.42, 0.22
Tasks: 122 total, 2 running, 120 sleeping, 0 stopped, 0 zombie
%Cpu(s): 17.2 us, 4.6 sy, 0.0 ni, 73.9 id, 4.3 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem: 948108 total, 264936 used, 683172 free, 25792 buffers
KiB Swap: 102396 total, 0 used, 102396 free. 126140 cached Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+  COMMAND
  844 root        20   0 634004 33620 12716 S   65.9   3.5   0:30.93  java
 1079 root        20   0  2284   1476  1352 S   11.2   0.2   0:03.09  Pi8s
 1090 root        20   0  2284   1448  1324 R   10.5   0.2   0:02.26  Pi8s
 1071 root        20   0   512   2524  2164 R    1.0   0.3   0:00.43  top
    6 root        20   0     0     0     0 S    0.3   0.0   0:02.88  kworker/u8+
   599 root        20   0     0     0     0 S    0.3   0.0   0:00.81  kworker/u8+
   637 root        20   0 38788 25084  9980 S    0.3   2.6   0:06.01  Xorg
    1 root        20   0   5424  3900  2736 S    0.0   0.4   0:04.46  systemd
    2 root        20   0     0     0     0 S    0.0   0.0   0:00.00  kthreadd
    3 root        20   0     0     0     0 S    0.0   0.0   0:00.07  ksoftirqd/0
    5 root         0 -20     0     0     0 S    0.0   0.0   0:00.00  kworker/0:+
    7 root        20   0     0     0     0 S    0.0   0.0   0:00.30  rcu_preempt
```

### Package Content

1x 1xUART HAT Board 1x Manual

### UART Driver download

<http://www.pridopia.co.uk/UartDownload.html>

### Product information

<http://www.pridopia.co.uk/rs-pi-i2c-luart.html>

<http://www.pridopia.co.uk/rs-pi-i2c-uart-autoload.html>

### Auto Load Pi8s when Pi Boot

Example : 2UART 0x49

```
sudo crontab -e
```

add 2 line in the end of file

```
@reboot /usr/bin/./Pi8s -a 0x49 -c 1 -b 38400 -s 1
```

```
@reboot /usr/bin/./Pi8s -a 0x49 -c 2 -b 38400 -s 2
```

when load minicom also open port1

```
minicom -D $(cat /dev/picom1)
```