
WizFi360

Application – UART Throughput

Version 1.0

WIZnet Co.,Ltd

Copyright© 2019

1.	Test environment.....	3
2.	Using Serial command	4
3.	The result of UART Throughput	6
Appendix 1.....		6

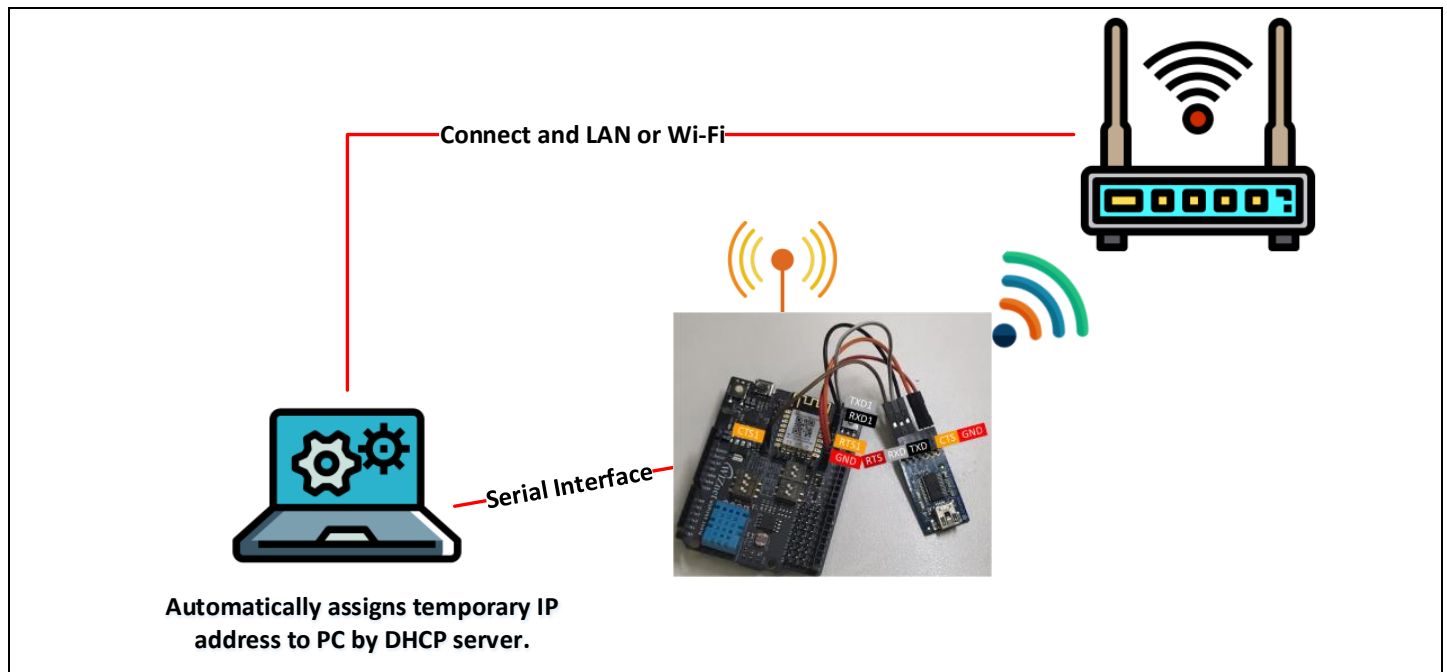
1. Test environment

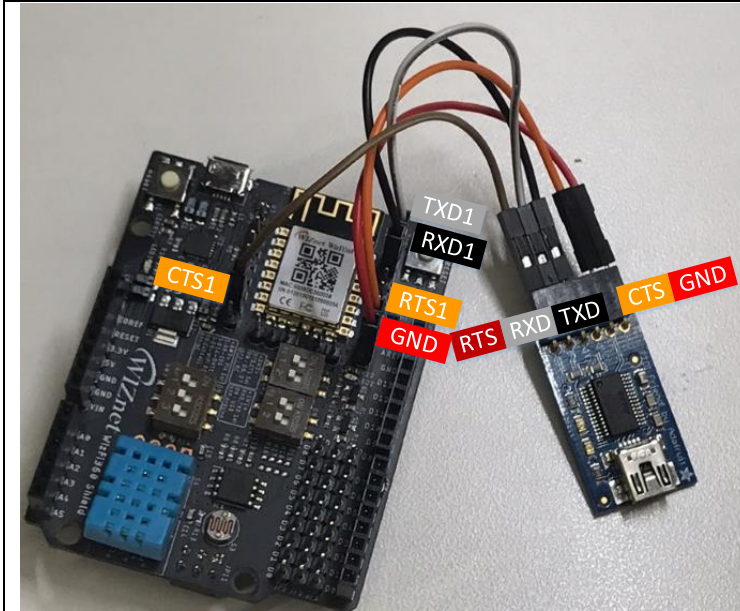
UART Throughput Test 를 하기 위해서는 CTS/RTS 를 이용한 제어가 필요하다.

- WizFi360 EVB or WizFi360io
- PC
- Serial Tool
 - o YAT Serial Tool(Data Mode)
 - o Python(Command Mode)
- 1Mbyte data file
- WiFi Router(SoftAP mode 를 사용할 경우 제외)

Data Mode 일 경우에는 YAT Serial Tool 를 이용하여 RTS/CTS 를 설정하고, DTR 로 Data Read 신호 설정해준다.
그런 후 RTS/CTS 가 제어하면서 데이터 전송이 이루어진다.

Command Mode 일 경우에는 AT+CIPSENDBUF command 를 이용하여 한번에 최대 보낼 수 있는 데이터인 2048 을 설정한 후 Data 2048 를 보내고, 또다시 AT+CIPSENDBUF 와 데이터를 반복해가면서 데이터 전송이 이루어진다.





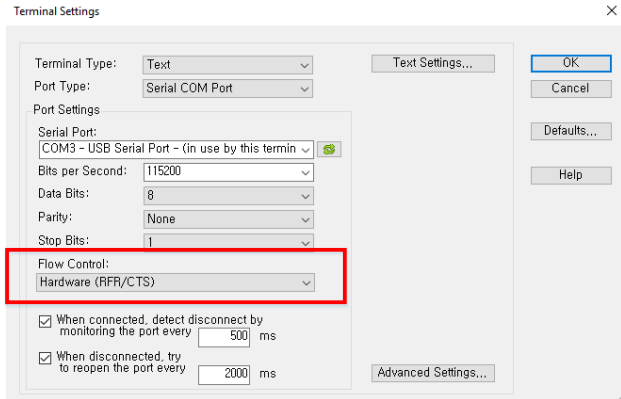
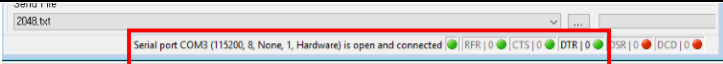
회로도 추가

2. Using Serial command

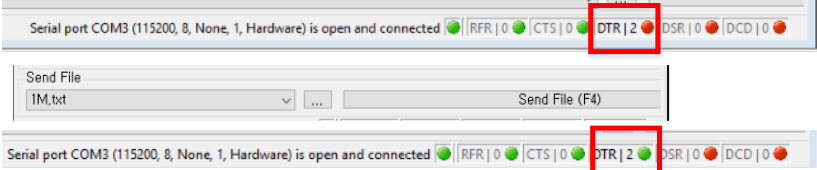
- Station Mode

AT command	Terminal
AT AT+CWMODE_CUR=1 AT+CWDHCP_CUR=1,1 AT+CWLAP AT+CWJAP_CUR="wizms1","maker0701" AT+CIPSTA_CUR?	<pre> AT<CR><LF> <CR><LF> OK<CR><LF> AT+CWMODE_CUR=1<CR><LF> <CR><LF> OK<CR><LF> AT+CWDHCP_CUR=1,1<CR><LF> <CR><LF> OK<CR><LF> AT+CWLAP<CR><LF> +CWLAP: (4,"DIR-815_Wiznet",-59,"",1)<CR><LF> +CWLAP: (0,"ESP_574935",-71,"",1)<CR><LF> +CWLAP: (3,"WIZnet_irina",-46,"",1)<CR><LF> +CWLAP: (3,"Matthew2.4",-63,"",2)<CR><LF> +CWLAP: (3,"rena",-46,"",3)<CR><LF> +CWLAP: (0,"iptime",-67,"",4)<CR><LF> +CWLAP: (3,"Dap",-63,"",5)<CR><LF> +CWLAP: (0,"ESP_577CC7",-67,"",6)<CR><LF> +CWLAP: (3,"wizms1",-63,"",6)<CR><LF> +CWLAP: (0,"Wizfi360",-69,"",6)<CR><LF> +CWLAP: (4,"DLINK-IPv6",-55,"",10)<CR><LF> +CWLAP: (0,"iptime",-59,"",11)<CR><LF> +CWLAP: (3,"WIZnet Scott",-51,"",11)<CR><LF> +CWLAP: (0,"WizFi360_A1B2D1",-69,"",11)<CR><LF> +CWLAP: (3,"Teddy_AP",-57,"",13)<CR><LF> <CR><LF> OK<CR><LF> AT+CWJAP_CUR="wizms1","maker0701"<CR><LF> WIFI_DISCONNECT<CR><LF> WIFI_CONNECTED<CR><LF> WIFI_GOT_IP<CR><LF> <CR><LF> OK<CR><LF> AT+CIPSTA_CUR?<CR><LF> +CIPSTA_CUR:ip:"192.168.1.120"<CR><LF> +CIPSTA_CUR:gateway:"192.168.1.1"<CR><LF> +CIPSTA_CUR:netmask:"255.255.255.0"<CR><LF> <CR><LF> OK<CR><LF> </pre>

- UART CTS/RTS Setting

AT command	Terminal
AT+CWUART_CUR = 115200,8,1,0,1	AT+UART_CUR=115200,8,1,0,1<CR><LF> <CR><LF> OK<CR><LF>
Terminal Setting	
<ol style="list-style-type: none"> 1. Ctrl+Shift+S > Open the Setiings 2. Flow Control안에 Hardware(RFR/CTS)로 변경 	
<ol style="list-style-type: none"> 3. Terminal창 아래에 오면 CTS/DTR이 초록으로 들어온 것을 확인할 수 있다. 	

- TCP Client /Data mode

AT command	Terminal
AT+CIPSTART="TCP","192.168.100.27",5001 AT+CIPMODE=1 AT+CIPSEND	AT+CIPSTART="TCP","192.168.100.27",5001<CR><LF> CONNECT<CR><LF> <CR><LF> OK<CR><LF> AT+CIPMODE=1<CR><LF> <CR><LF> OK<CR><LF> AT+CIPSEND<CR><LF> <CR><LF> >
Terminal Setting	
<ol style="list-style-type: none"> 1. DTR이 빨간불일 때, 1M.txt를 보내고, 2. DTR를 클릭해서 초록불로 바뀌면 데이터가 Serial을 통해 전송되게 된다. 	

- TCP Client / Command mode

AT command	Terminal
------------	----------

AT+CIPSTART="TCP","192.168.100.27",5001 AT+CIPMODE=0 AT+CIPSENDBUF=2048 Send the 1Mbyte.txt	AT+CIPSTART="TCP","192.168.100.27",5001<CR><LF> CONNECT<CR><LF> <CR><LF> OK<CR><LF> AT+CIPMODE=0<CR><LF> AT+CIPMODE=0<CR><LF> <CR><LF> OK<CR><LF> AT+CIPSENDBUF=2048<CR><LF> AT+CIPSENDBUF=2048<CR><LF> 1,0<CR><LF> <CR><LF> OK<CR><LF> >
--	---

3. The result of UART Throughput

1Mbyte를 PC에서 WizFi360의 Serial(UART1)로 데이터를 보내고, TCP Server로 데이터를 전송한다.

Baud rate	Data mode		Command mode	
	Time	Speed(bit/s)	Time	Speed(bit/s)
115200	123s	66K		
921600	16.3s	502K		
1000000	14.9s	550K		
1250000	12.7s	645K		
1500000	10.5s	780K		
2000000	9.7s	845K		

해당 속도는 WireShark 를 이용하여, 데이터 전송시작부터 완료되는 시점까지의 시간을 측정한 것은 Appendix 1을 보면 된다.

Appendix 1

Baud rate	Data mode	Command mode
115200	123s : 66Kbit/s <pre> 3823 122.869907 192.168.100.27 192.168.100.28 TCP 54 5001 → 52161 [ACK] Seq=1023825 Win=65535 Len=0 3824 122.865190 192.168.100.28 192.168.100.27 TCP 490 52161 → 5001 [PSH, ACK] Seq=1023825 Ack=1 Win=6144 Len=436 [TCP segment of... 3825 122.900628 192.168.100.27 192.168.100.28 TCP 54 5001 → 52161 [ACK] Seq=1023461 Win=65099 Len=0 3826 122.912979 192.168.100.28 192.168.100.27 TCP 594 52161 → 5001 [PSH, ACK] Seq=1023461 Ack=1 Win=6144 Len=540 [TCP segment of... 3827 122.958838 192.168.100.27 192.168.100.28 TCP 54 5001 → 52161 [ACK] Seq=1024801 Win=64559 Len=0 </pre>	
921600	16.3s : 502Kbit/s <pre> 2547 16.217822 192.168.100.28 192.168.100.27 TCP 1878 52165 → 5001 [ACK] Seq=1023185 Ack=1 Win=6144 Len=1034 [TCP segment of... 2548 16.217968 192.168.100.27 192.168.100.28 TCP 54 5001 → 52165 [ACK] Seq=1023185 Win=65535 Len=0 2549 16.317138 192.168.100.28 192.168.100.27 TCP 870 52165 → 5001 [PSH, ACK] Seq=1023185 Ack=1 Win=6144 Len=836 [TCP segment of... 2550 16.357729 192.168.100.27 192.168.100.28 TCP 54 5001 → 52165 [ACK] Seq=1024801 Win=64719 Len=0 </pre>	
1000000	14.9s : 550Kbit/s <pre> 3868 14.774321 192.168.100.28 192.168.100.27 TCP 490 58128 → 5001 [PSH, ACK] Seq=1023125 Ack=1 Win=6144 Len=436 [TCP segment of... 3869 14.815213 192.168.100.27 192.168.100.28 TCP 54 5001 → 58128 [ACK] Seq=1023561 Win=65099 Len=0 3870 14.828495 192.168.100.28 192.168.100.27 TCP 494 58128 → 5001 [PSH, ACK] Seq=1023561 Ack=1 Win=6144 Len=440 [TCP segment of... 3871 14.895281 192.168.100.27 192.168.100.28 TCP 54 5001 → 58128 [ACK] Seq=1024801 Win=64559 Len=0 </pre>	
1250000	12.7s : 645Kbit/s <pre> 2863 12.592400 192.168.100.28 192.168.100.27 TCP 490 58136 → 5001 [PSH, ACK] Seq=1023185 Ack=1 Win=6144 Len=436 [TCP segment of... 2864 12.631883 192.168.100.27 192.168.100.28 TCP 54 5001 → 58136 [ACK] Seq=1023621 Win=65099 Len=0 2865 12.631959 192.168.100.28 192.168.100.27 TCP 434 58136 → 5001 [PSH, ACK] Seq=1023621 Ack=1 Win=6144 Len=436 [TCP segment of... 2866 12.674856 192.168.100.27 192.168.100.28 TCP 54 5001 → 58136 [ACK] Seq=1024801 Win=64719 Len=0 </pre>	
1500000	10.5s : 780Kbit/s	

	<div> <div>2242 10.389973</div> <div>192.168.100.28</div> <div>192.168.100.27</div> <div>TCP</div> <div>490 65021 → 5001 [PSH, ACK] Seq=1023074 Ack=1 Win=6144 Len=436 [TCP</div> </div> <div> <div>2244 10.430942</div> <div>192.168.100.27</div> <div>192.168.100.28</div> <div>TCP</div> <div>54 5001 → 65021 [ACK] Seq=1 Ack=1023510 Win=65099 Len=0</div> </div> <div> <div>2245 10.445987</div> <div>192.168.100.28</div> <div>192.168.100.27</div> <div>TCP</div> <div>545 65021 → 5001 [PSH, ACK] Seq=1023510 Ack=1 Win=6144 Len=491 [TCP</div> </div> <div> <div>2247 10.486915</div> <div>192.168.100.27</div> <div>192.168.100.28</div> <div>TCP</div> <div>54 5001 → 65021 [ACK] Seq=1 Ack=1024001 Win=64608 Len=0</div> </div>	
2000000	<div>9.7s : 845Kbit/s</div> <div> <div>6316 9.646307</div> <div>192.168.100.28</div> <div>192.168.100.27</div> <div>TCP</div> <div>490 65031 → 5001 [PSH, ACK] Seq=1023245 Ack=1 Win=6144 Len=436</div> </div> <div> <div>6317 9.686546</div> <div>192.168.100.27</div> <div>192.168.100.28</div> <div>TCP</div> <div>54 5001 → 65031 [ACK] Seq=1 Ack=1023681 Win=65099 Len=0</div> </div> <div> <div>6318 9.690489</div> <div>192.168.100.28</div> <div>192.168.100.27</div> <div>TCP</div> <div>374 65031 → 5001 [PSH, ACK] Seq=1023681 Ack=1 Win=6144 Len=320</div> </div> <div> <div>6319 9.731538</div> <div>192.168.100.27</div> <div>192.168.100.28</div> <div>TCP</div> <div>54 5001 → 65031 [ACK] Seq=1 Ack=1024001 Win=64779 Len=0</div> </div>	

History

Ver	Date	Description
1.0	Aug.2019	Initial version