P053 Tester Muti-DUTx4

2019/10/18 v0.8

By CNHsu

|  |  |  |
| --- | --- | --- |
| Date | Version | Description |
| 2019/9/16 | V0.1 | First draft |
| 2019/9/19 | V0.2 | Modify reg. |
| 2019/9/20 | V0.3 | +step motor |
| 2019/9/24 | V0.4 | +BT addr |
| 2019/9/26 | V0.5 | Revise some text  +uart log reference |
| 2019/10/1 | V0.6 | +freq. test log |
| 2019/10/2 | V0.7 | Replace PWM.h to tone library  +’q’ to stop pwm output  Can direct drive speaker for “talk mic “ test |
| 2019/10/18 | V0.7 | +servo  \*remove step moto function |
|  |  |  |

# System block

UART=38400

UART=57600

PC

UART=115200

PCBA

Mega328p

(arduino nano)

1. 5V/4V EN
2. Check V/I
3. Check Tx
4. PWM &FFT
5. LED

12V

5V

4V

5V, 4V,TX,MIC,SP

Atmega2560

1. Select pcba
2. Command & results
3. BT control
4. Motor control?

PCBA

Mega328p

(arduino nano)

1. 5V/4V EN
2. Check V/I
3. Check Tx
4. PWM &FFT
5. LED

12V

5V

4V

5V, 4V,TX,MIC,SP

PCBA

Mega328p

(arduino nano)

1. 5V/4V EN
2. Check V/I
3. Check Tx
4. PWM &FFT
5. LED

12V

5V

4V

5V, 4V,TX,MIC,SP

PCBA

Mega328p

(arduino nano)

1. 5V/4V EN
2. Check V/I
3. Check Tx
4. PWM &FFT
5. LED

12V

5V

4V

5V, 4V,TX,MIC,SP

Tx,Rx

**Setup environment:**

1. **BES FW: 38400**

**\\192.168.1.50\p053\RDDOC\治具\開發文件\test-multi\bes-fw**

1. **Arduino FW:** **\\192.168.1.50\p053\RDDOC\治具\開發文件\test-multi\arduino**
2. **sch:** **\\192.168.1.50\p053\RDDOC\治具\開發文件\test-multi\sch**
3. **uart speed : BES to nano(38400), nano to mega(57600), mega to PC (115200)**
4. **mega uart port:**

**serial0(to pc): usb to pc**

**serial1(to BT):tx(18)rx(19)**

**serial2(to pcba1):tx(16)rx(17) ….. bes rx only**

**serial3(to pcba2):tx(14)rx(15) ….. bes rx only**

**serial4(to pcba3):tx(48)rx(50) ….. bes rx only**

**serial5(to pcba4):tx(49)rx(51) ….. bes rx only**

1. **mega driver:** **\\192.168.1.50\p053\RDDOC\治具\開發文件\test-multi\arduino**

**Atmega2560 to mega328p (char):**

Example: 1 means char ‘1’

V/I control & test

|  |  |  |
| --- | --- | --- |
| Command (char) | Action | Description |
| 1 | 5V on | Uart: "(1)\*\*5 ON\n\r"  (?) depends pcba |
| 2 | 5V off | "(1)\*\*5 OFF\n\r" |
| 3 | 4V on | "(1)\*\*4V ON\n\r" |
| 4 | 4V off | "(1)\*\*4 OFF\n\r" |
| 5 | Check 5V voltage | "(1)\*\*5 voltage\n"  “5VV”?????  ? show voltage value |
| 6 | Check 5V current | "(1)\*\*5 current\n"  “5VI”????  ? show current value |
| 7 | Check 4V voltage | "(1)\*\*5 voltage\n"  “4VV”?????  ? show voltage value |
| 8 | Check 4V current | "(1)\*\*5 current\n"  “4VI”????  ? show current value |
| 9 | Set uA | "(1)\*\*5V uA\n\r" |
| a | Set mA | "(1)\*\*5V mA\n\r" |
| B | Set uA | "(1)\*\*4V uA\n\r" |
| c | Set mA | "(1)\*\*4V mA\n\r" |
| d | PWM=500Hz  PWM2=1KHz | "(1)\*\*PWM”  D9  D9=1k,D3=500 \n\r" |
| e | PWM=1KHz  PWM2=500Hz | "(1)\*\*PWM”  D3  D9=500,D3=1k \n\r" |
| f | LED7=1 |  |
| g | LED7=0 |  |
| h | LED8=1 |  |
| I | LED8=0 |  |
| q | Stop tone | Stop pin9,3 pwm output |
| s | Check USB5V status  ‘1’=OK  ‘0’=fail | "(1)\*\*FLT state\n\r" |
| x | Read R/L frequency | "(1)\*\*FFT \n\r"  L=A7, R=A5  Show:  Data:  FFTL=????  Data:  FFTR=????  ? show frequency |
| z | Check key log  And  BT address | "(1)\*\*wait..."  "press key now ....."  Will count down 5,4,3,2,1…  "# Powe on  "#PWR"  "#V-"  "#v+"  "#ANC"  "#PWR off"  \*\* trip-click at normal power on state, can get BT address  “#BT : “ address string  \*\* time out ~7sec |

**PCBA DUT and signals select(Atmega2560 only):**

|  |  |  |
| --- | --- | --- |
| Command prefix “#” (char), ex. ‘#1” | Action | Description |
| #1  Or #2 or #3 or #4 | Select pcba  1 or 2, or 3 or 4 | "select serial2\n\r"  "select serial3\n\r"  "select serial4\n\r"  "select serial5\n\r"  \*default pcba=1 |
|  |  |  |

\*for multi-port test

Should select pcba ‘#?’ firstly before other operation. (default pcba select ‘1’)

**HC-05 PIO 2-8 & PIO 11:**

// "PIO=2,1" // port-2 hight,

// "PIO=7,0" // port-7 low

|  |  |  |
| --- | --- | --- |
| Command prefix “p” (char), ex. ‘#1” | Action | Description |
| PIO=2,1 | BT IO 2=high | Uart :"\*\*HC-05 PIO\n\r"  BT uart :”AT”+PIO2,1  BT uart:"AT+MPIO\n\r "  Return PIO status |
|  |  |  |

**Bluetooth (atmega2560 only) command:**

|  |  |  |
| --- | --- | --- |
| Command (char), ex. ‘1’ | Action | Description |
| u | "AT+RESET\n\r "  Delay(1000)  "AT+INIT\n\r "  Delay(1000)  "AT+state\n\r "  delay(500);  "AT+INQM=1,1,3\n\r  delay(500);  "AT+class=240404\n\r  Delay(500) | "\*\*LED1-BT INIT\n\r"  Initial BT  \*\* include 3.5sec delay |
| v | Check BT RSSI | "\*\*LED1-BT rx: scan\n\r"  "BT scan\n\r "  Rssi=????  ? BT energy  ==> ?  ? loop counter 0-9  "---end scan ---\n\r"  \*\*if scan fail, repeat ‘v’ command to re-scan |
| w | Check INQM status | "\*\*BT INQM\n\r"  It should be 1,1,3 |

**Step motor and other (atmega2560 only):**

**BEStest\_BT\_5**

**Disable step motor function**

|  |  |  |
| --- | --- | --- |
| Command prefix “%”  Command (char), ex. “%s100” | Action | Description |
| n | Reset nano board | D44,45,46,47 toggle  Low-delay 0.5s- high  \*nano ‘low’ reset |
| s | Speed ‘rmp’ | "set stepper speed(rpm) : " |
| d | steps | "set stepper steps : "   1. full cycle=4096 2. neg-value counterclockwise   \*after stpes, set all INTx to low |
| r | running | Run with pre-set value in ‘s’ &’d’ |
| t | Terminal step motor | Set all INTx to low |

\*step motor: 28BYJ-48 (5V, 4096steps for 1-cycle)

**Servo motor and other (atmega2560 only):**

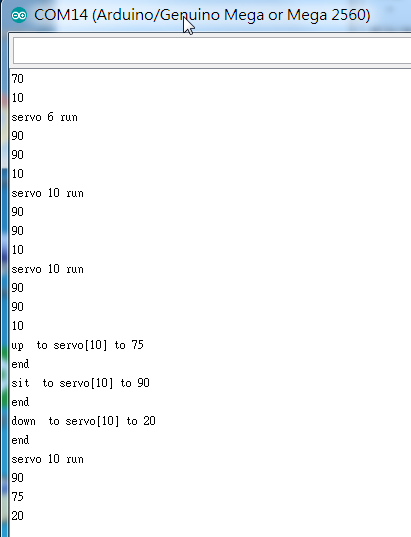
**Servo CH1:D22,23,24,25 (servo0,1,2,3)**

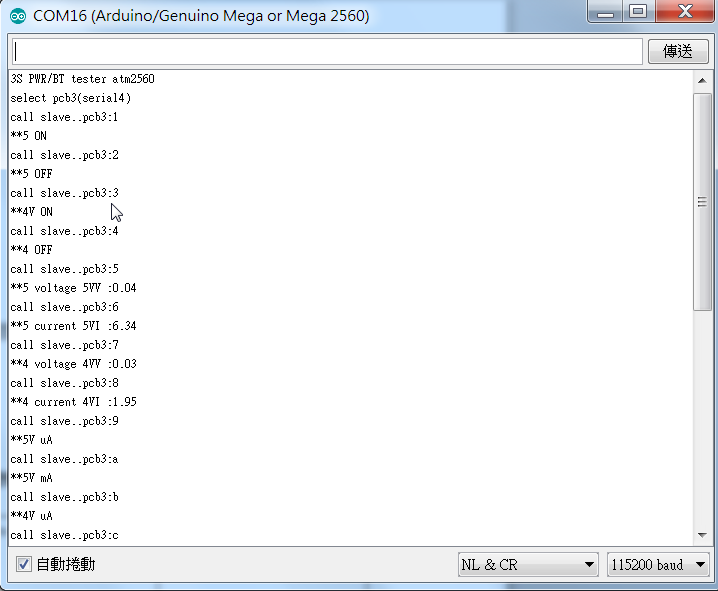
**Servo CH2:D26,27,28,29(servo4,5,6,7)**

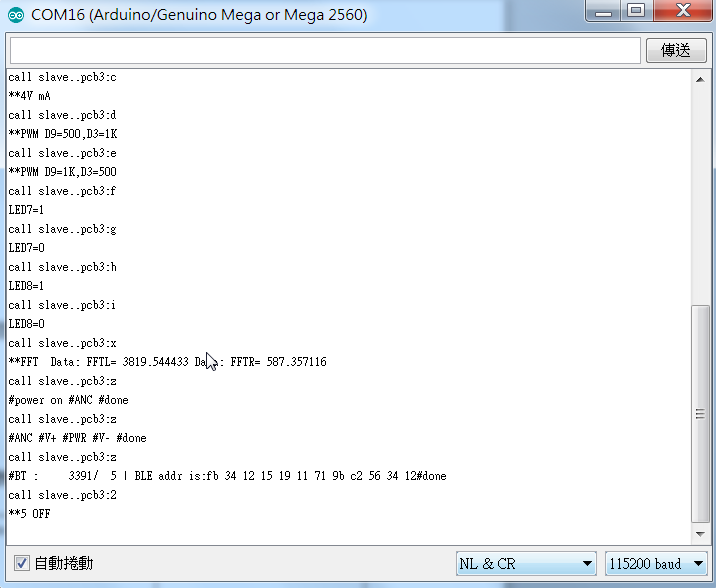
**Servo CH3:D30,31,32,33(servo8,9,10,11)**

**Servo CH4:D34,35,36,37(servo12,13,14,15)**

|  |  |  |
| --- | --- | --- |
| Command prefix “%”  Command (char), ex. “%i11” | Action | Description |
| n | Reset nano board | "\*\*Reset nano 0.5sec  D44,45,46,47 toggle  Low-delay 0.5s- high  \*nano ‘low’ reset |
| s[] | Set servo ‘sit’ position deg | "set sit deg:  %s100: set all servo ‘sit’ to 100deg |
| u[] | Set servo ‘up’ position deg | "set up deg:  %u90: set all servo ‘up’ to 90deg |
| d[] | Set servo ‘down’ position deg | "set down deg:  %d45: set all servo ‘down’ to 45deg |
| I[x][y][z] | set specific servo’s sit or up or down deg | “sit/up/down to servo [y] to [z]”  Set servo z, x=s/u/d, z=deg  %s01100: set servo1 ‘sit’ deg100  $u1245: set servo12 ‘up deg45  Valid range:  x: ‘s’,’d’,’u’  y:0-15  z: 1-175 (deg) |
| r[x] | Run servo[x]  With pre-define  Sit,up,down | “servo x run”  Sit  Up  Down  sit |







Command ‘d’, ‘e’ and ‘x’ test log

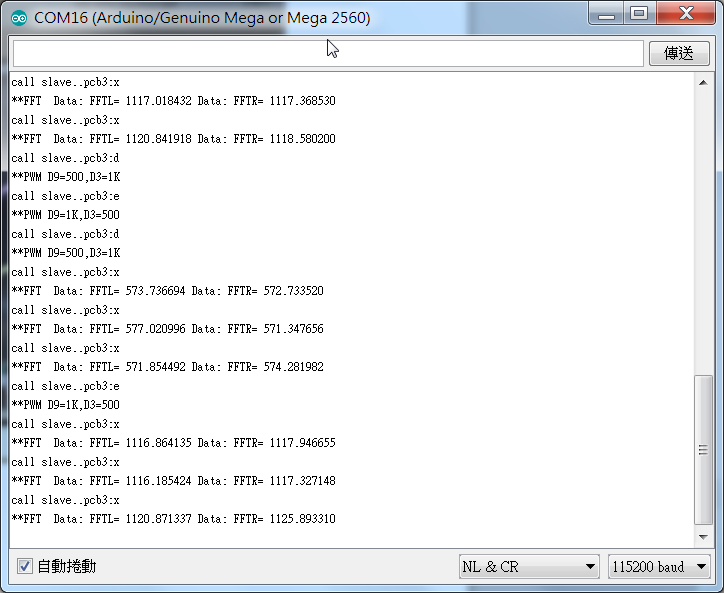


MICP

Sp+, sp-

MICP: Arduino freq. output

SP+SP-: BES speaker output



Check FFT

Check FFT

Set 500Hz

Set 1KHz

Sp+, sp-