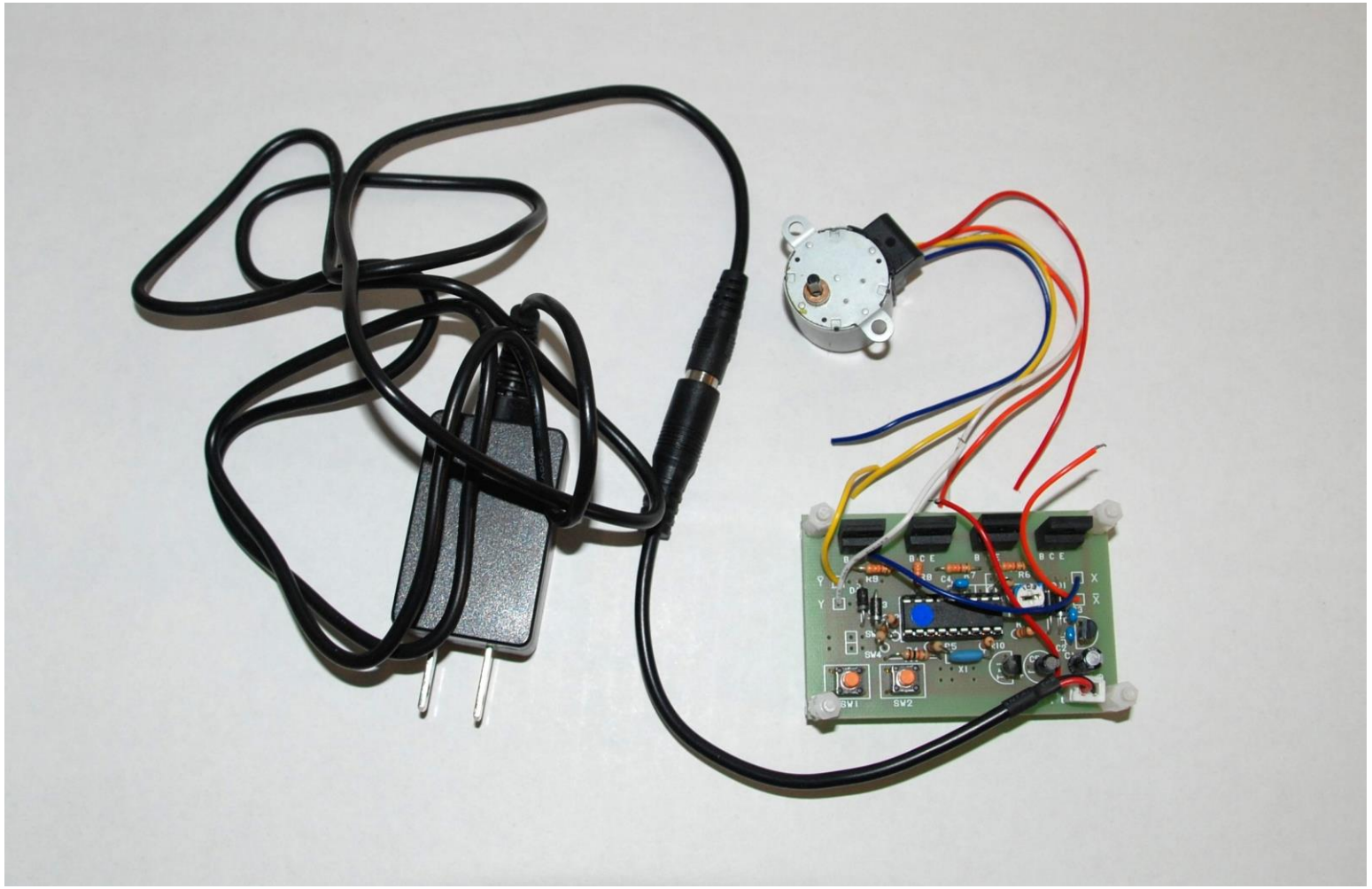
Dr-motor-test

Music Drone Project
2023/03/15
アッピョミュージックスタジオ事務所
2024/7/22 h.

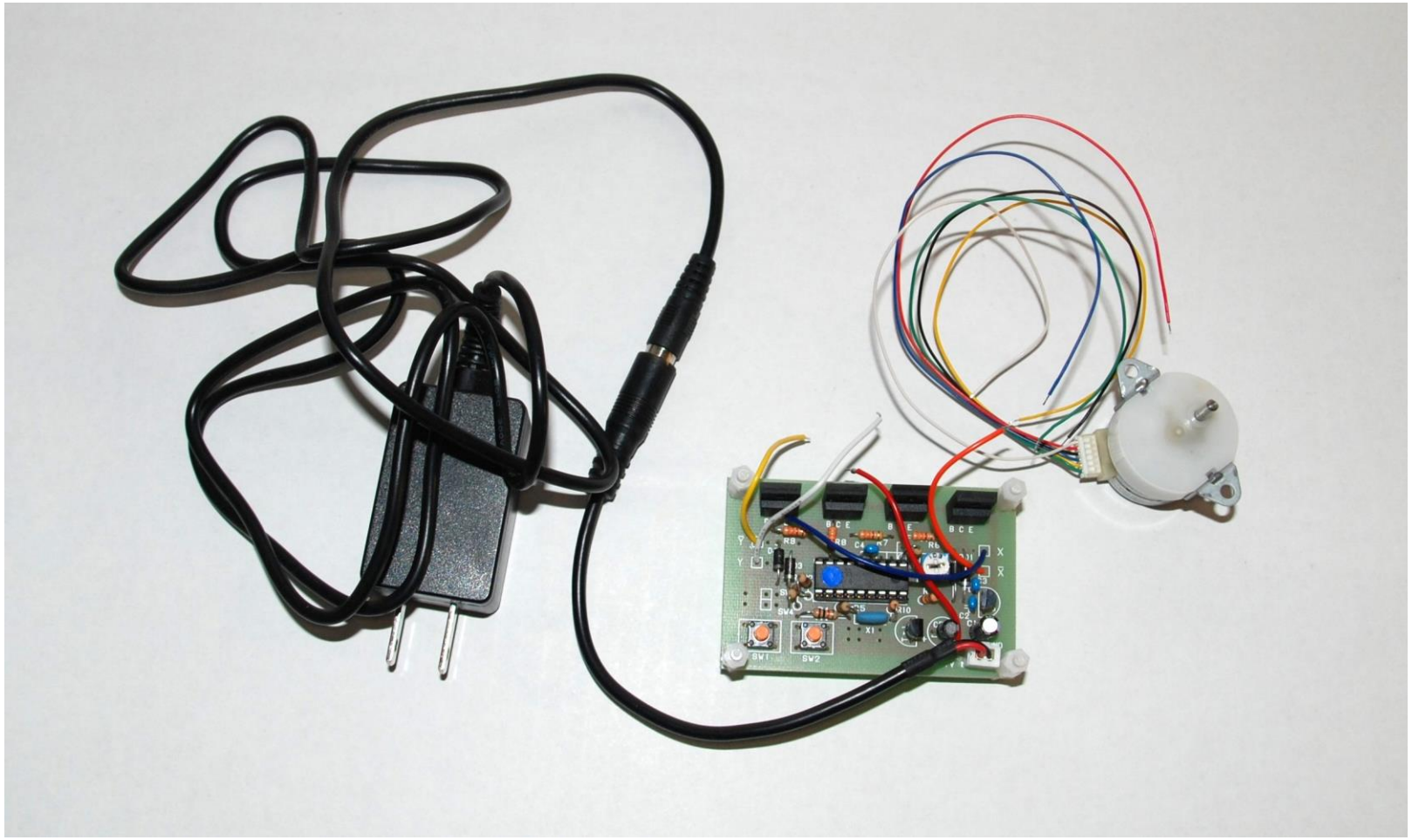
AE-STEP-KIT test

ステッピングモータ・コントロールキット
製作実験



AE-STEP-KIT　test

ステッピングモータ・コントロールキット
製作実験



AE-STEP-KIT
ステッピングモータ・コントロールキット
ユニポーラ
製作実験

キットの動作は正常に動作しました。
だだし、動作を長時間行うとモータが熱を持ち回転しなくなりました。
モータの故障かと、PS-25JC2 -> 595BA に変更
モーターを変えても回転はしません。
原因
　基板不良？
PS-25JC2を再度購入が必要？

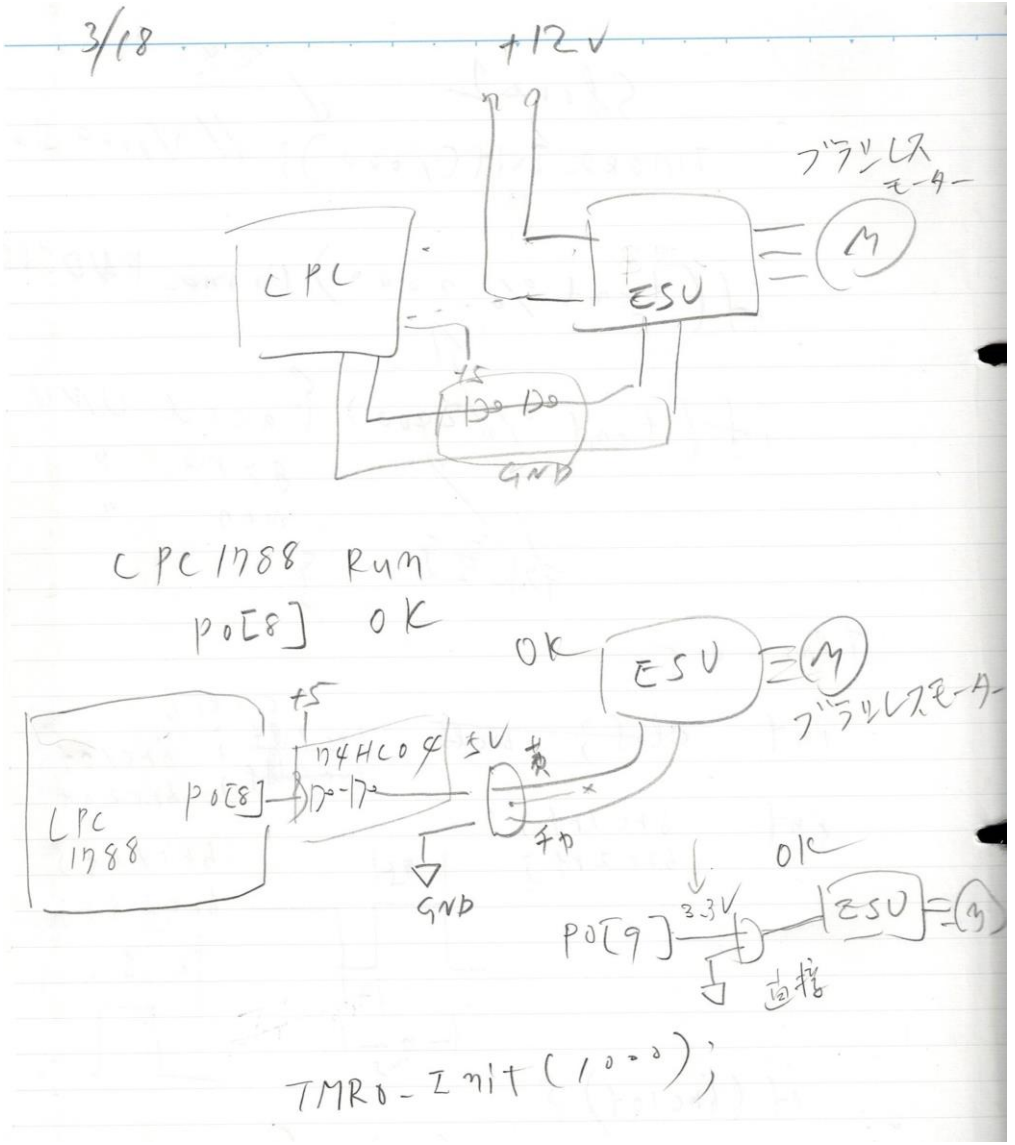
テストは中断

LPC1788でブラシレスモーターを駆動する。

LPC1788

ブラシレスモーター

ESU

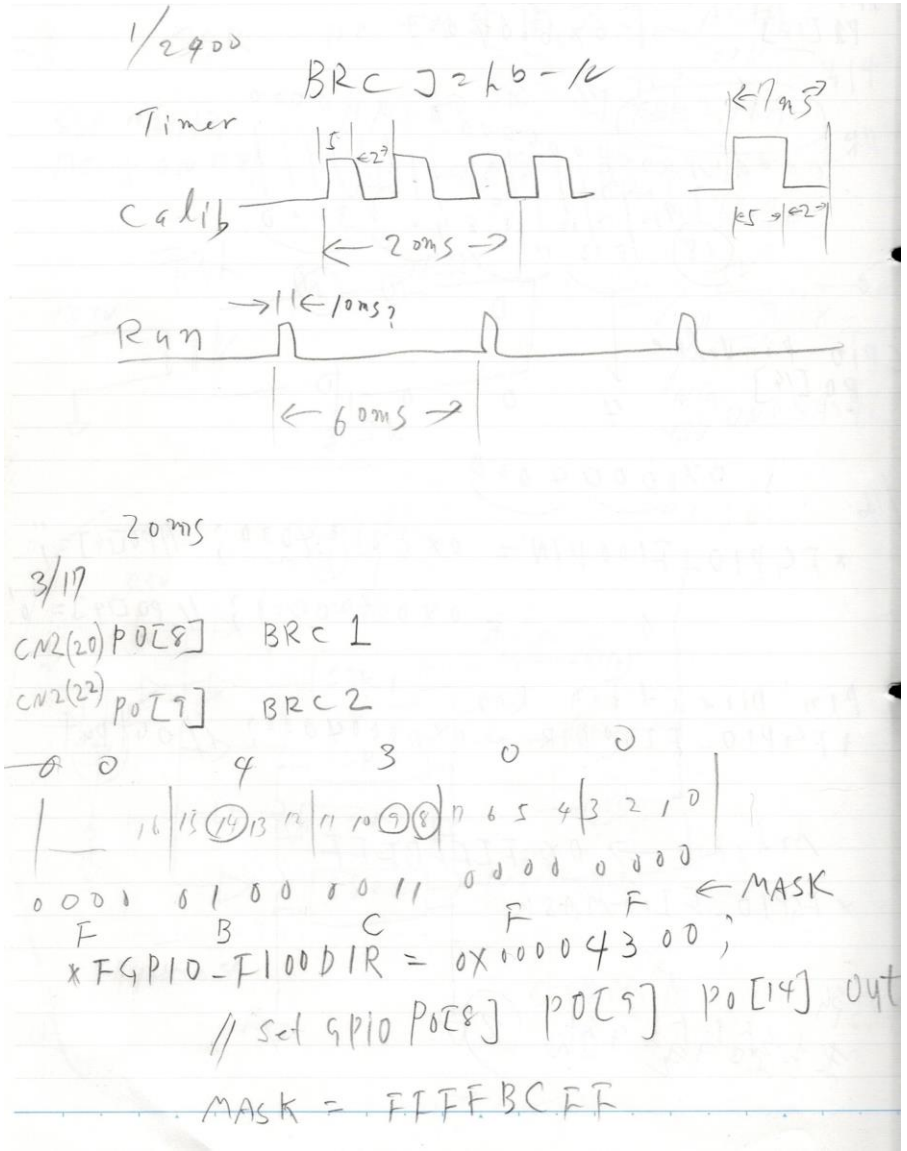


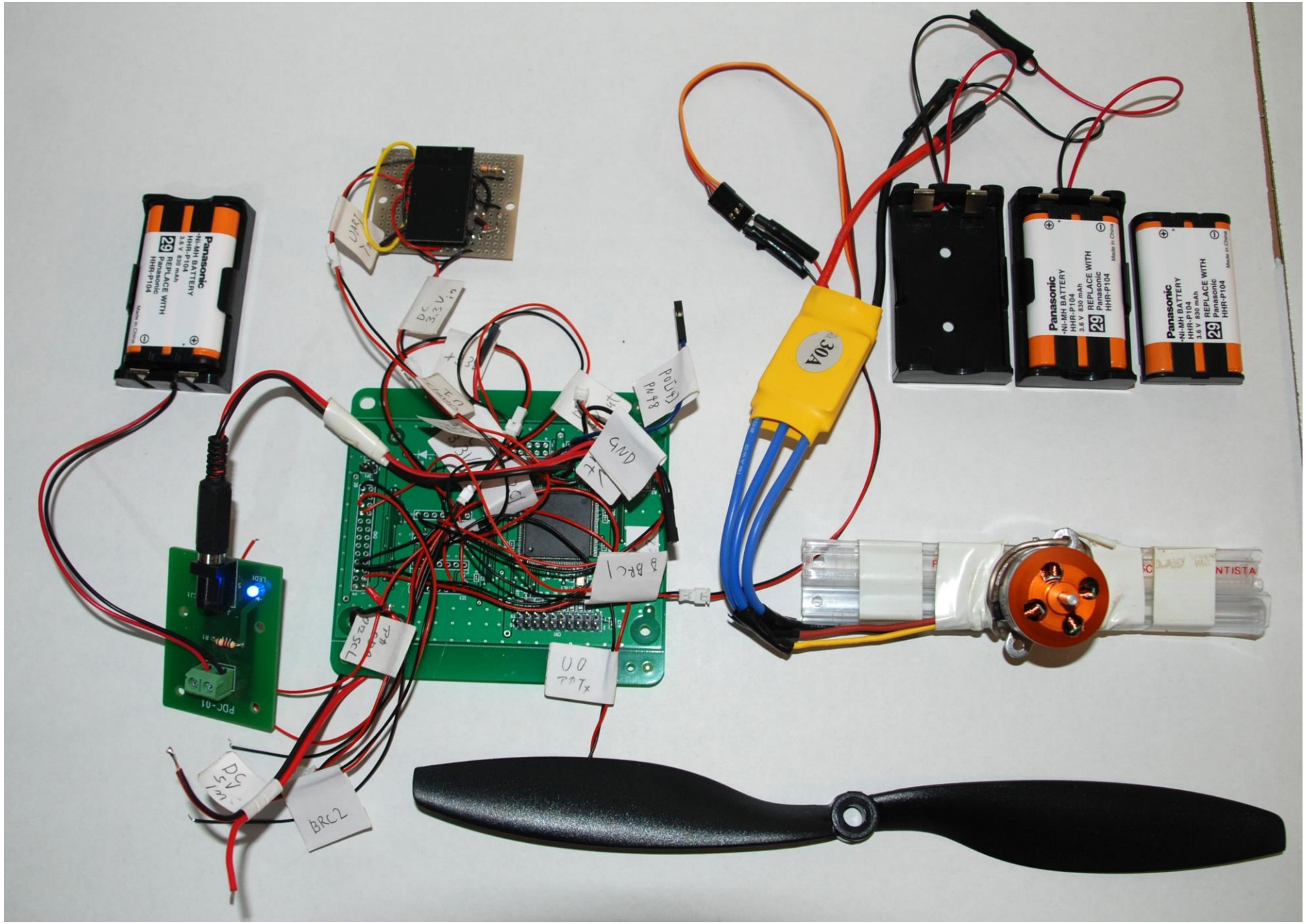
LPC1788でブラシレスモーターを駆動する。

LPC1788

ブラシレスモーター

ESU

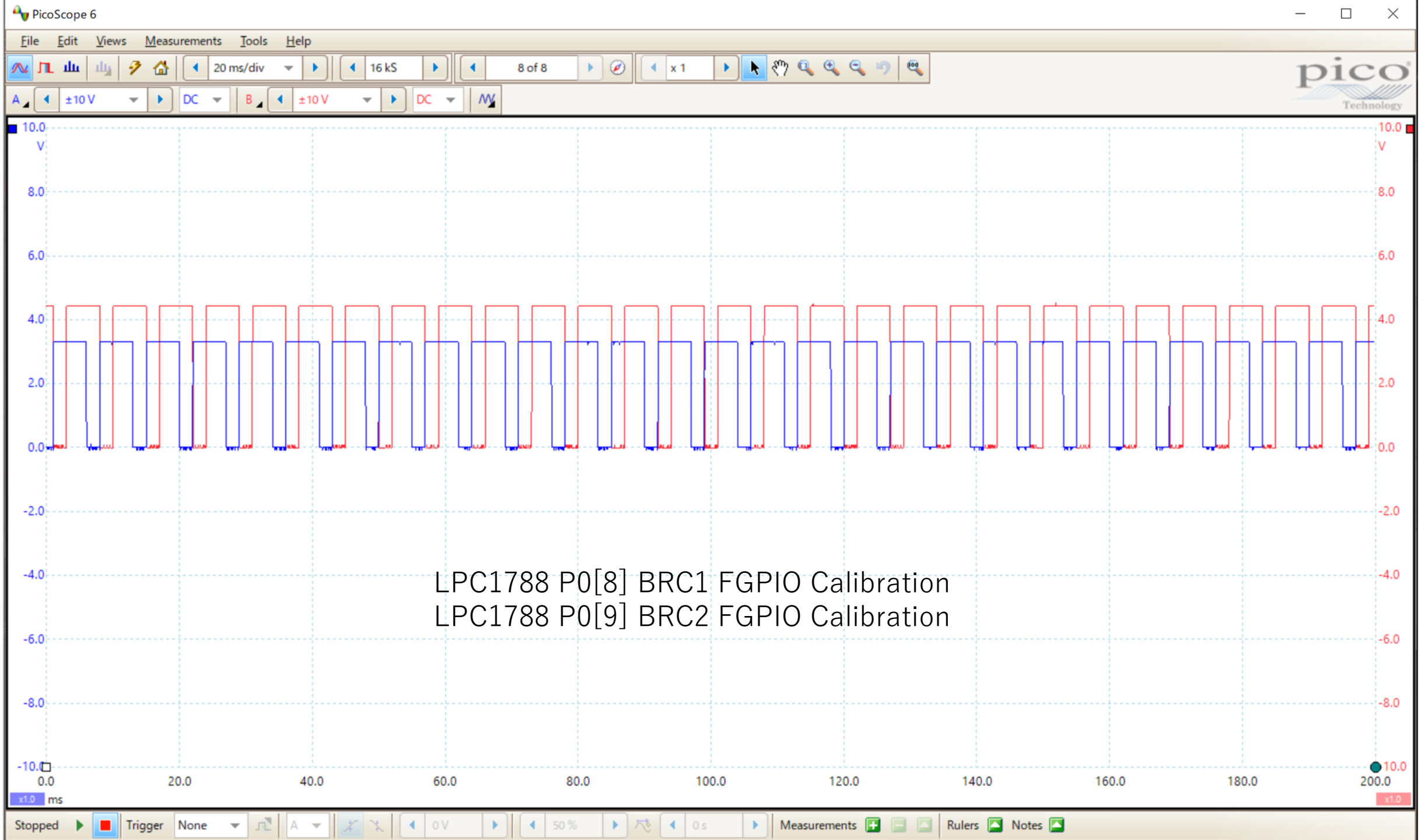


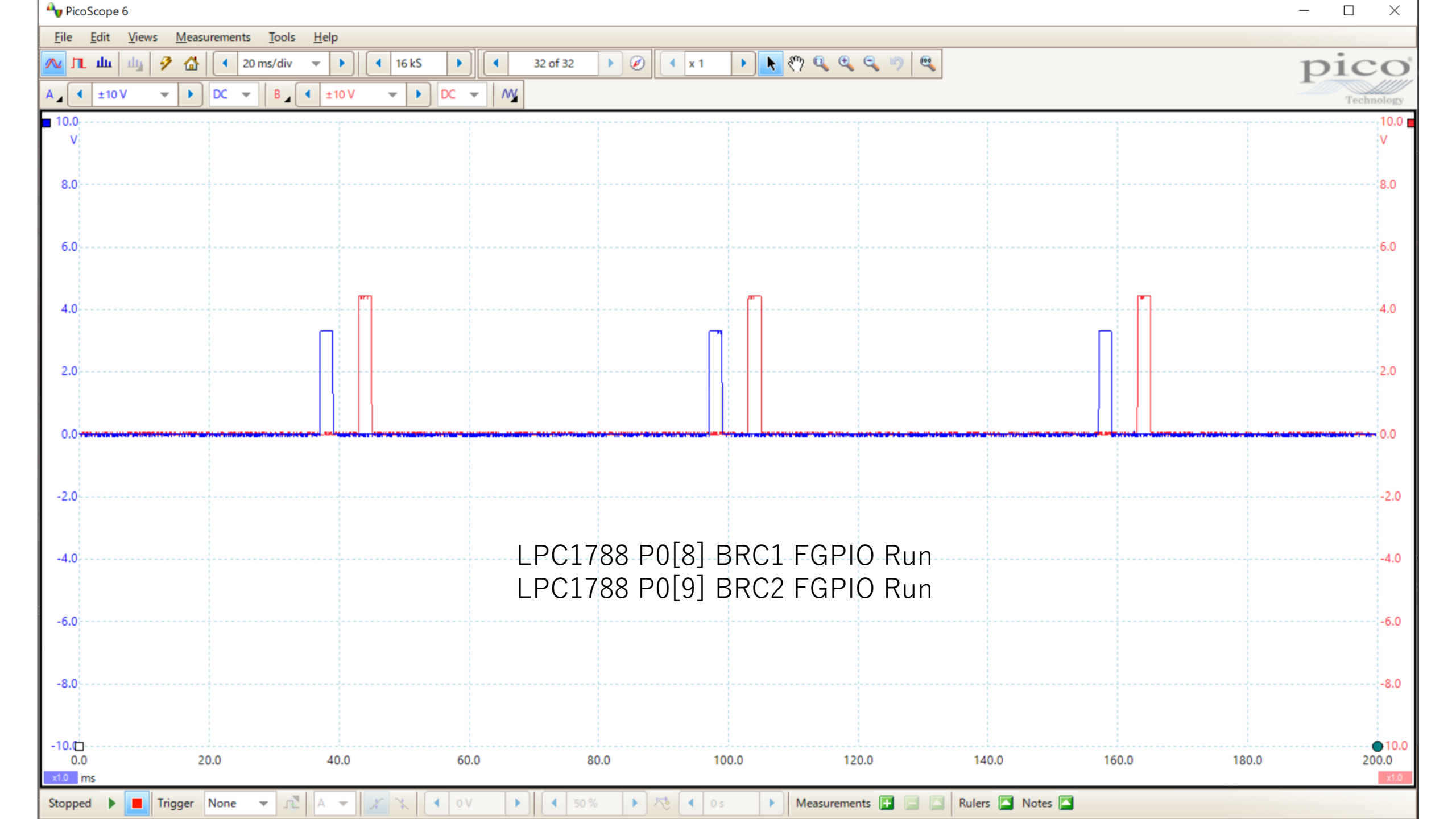

Panasonic
NI-MH BATTERY
HHR-P104
3.6 V 830 mAh
29
REPLACE WITH
Panasonic
HHR-P104
Made in China
30A
GND
PDC-01
BRC2

利用CPUとインターフェース

CPU基板、NXP LPC1788 ,ARM ,Cortex-M3
開発環境 IAR Embedded Workbench for Arm
基板製造 Appyo Music Studio

利用 Interface
　UART1　TWELITE uart（Music Server との通信）


PicoScope 6
File Edit Views Measurements Tools Help
20 ms/div
16 kS
8 of 8
x 1
A ±10 V DC
B ±10 V DC
pico Technology
10.0 V
8.0
6.0
4.0
2.0
0.0
-2.0
-4.0
-6.0
-8.0
-10.0
0.0 20.0 40.0 60.0 80.0 100.0 120.0 140.0 160.0 180.0 200.0
ms
LPC1788 P0[8] BRC1 FGPIO Calibration
LPC1788 P0[9] BRC2 FGPIO Calibration
Stopped
Trigger None
Measurements
Rulers
Notes


PicoScope 6
File Edit Views Measurements Tools Help
20 ms/div
16 kS
32 of 32
x 1
A ±10 V DC
B ±10 V DC
pico Technology
10.0 V
8.0
6.0
4.0
2.0
0.0
-2.0
-4.0
-6.0
-8.0
-10.0
0.0 20.0 40.0 60.0 80.0 100.0 120.0 140.0 160.0 180.0 200.0
ms
LPC1788 P0[8] BRC1 FGPIO Run
LPC1788 P0[9] BRC2 FGPIO Run
Stopped
Trigger None
Measurements
Rulers
Notes

```
// GPIO IOCON
#define IOCON_P0_4      ((volatile unsigned int *)(0x4002C010)) // Gstop=1
#define IOCON_P0_5      ((volatile unsigned int *)(0x4002C014)) // Pose
#define IOCON_P0_6      ((volatile unsigned int *)(0x4002C018)) // Gstop=4
#define IOCON_P0_7      ((volatile unsigned int *)(0x4002C01C)) // Stop
#define IOCON_P0_8      ((volatile unsigned int *)(0x4002C020))
#define IOCON_P0_9      ((volatile unsigned int *)(0x4002C024))
#define IOCON_P0_14     ((volatile unsigned int *)(0x4002C038)) // GPIO
#define IOCON_P1_18     ((volatile unsigned int *)(0x4002C0C8)) // LED2
```

P0[8] P0[9] FGPIO

```
*IOCON_P0_8 = 0x00;   // Set P0[8] -> GPIO BRC1
*IOCON_P0_9 = 0x00;   // Set P0[9] -> GPIO BRC2
*IOCON_P0_10 = 0x00;  // Tx2 D-Slave8 Tbr7
*IOCON_P0_14 = 0x00;  // Set P0[14] -> GPIO
*IOCON_P0_19 = 0x30;  // SDA1 -> GPIO Tbr5
*IOCON_P0_20 = 0x30;  // SCL1 -> GPIO Tbr6
*IOCON_P0_22 = 0x30;  // Tx4  -> GPIO test NG
//*IOCON_P0_25 = 0x30;  // Tx3  -> GPIO Tbr8
```

Main()

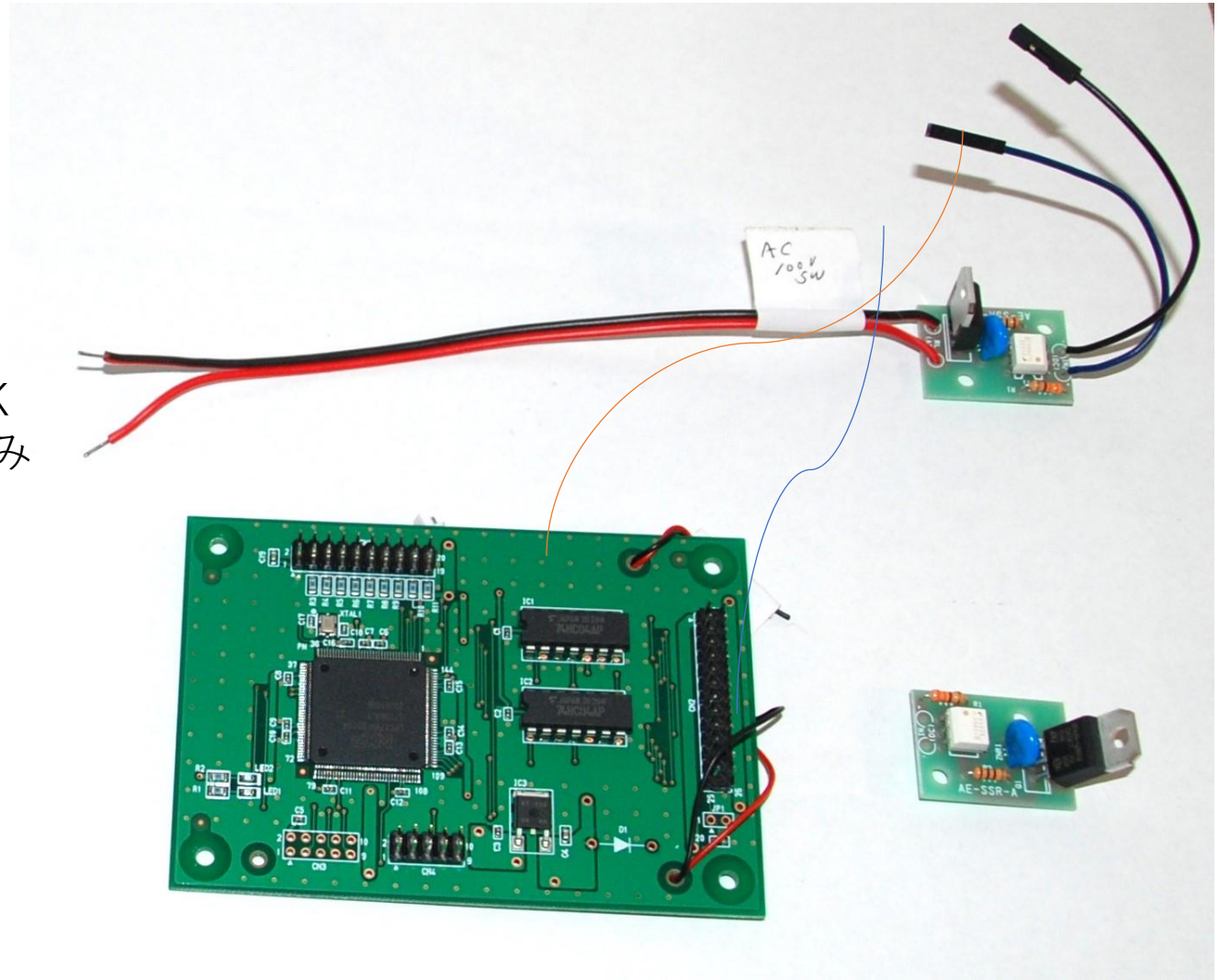
```
////////////  BRC controle ////////////
brc1cf = 0;
GPIO_P0_8_High(); // test
GPIO_P0_9_High(); // test
//GPIO_P0_14_High();  // Br SW ON
brcnt1 = 0;
brcnt2 = 0;
//brc1cf = 1; // BRC1 calibration ON
//for( i=0; i < 200000000; i++); //delay
//brc1cf = 0; // BRC1 calibration OFF
brc1rf = 2; // BRC1 Run
for( i=0; i < 60000; i++); //delay
brc2rf = 2; // BRC2 Run
Tbr_All_tst_5();    // Tbr All_5 test ok
for( i=0; i < 100000000; i++); //delay
brc1rf = 0; // BRC1 Run off
brc2rf = 0; // BRC2 Run off
//GPIO_P0_14_Low(); // Br SW OFF
GPIO_P0_8_Low();
GPIO_P0_9_Low();
```

Timer IRQ

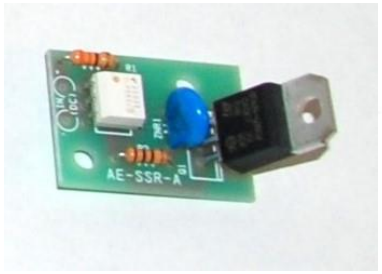
```
/*****************************************************************************
 * Function Name: TMR0_IRQHandler
 * Parameters: none
 *
 * Return: none
 *
 * Description: Timer 0 interrupt handler
 *
 *****************************************************************************/
void TMR0_IRQHandler (void)
{;
    if(brc1cf) {
        if(brcnt1 >= 2) GPIO_P0_8_High();
        if(brcnt1 >= 7) {
            GPIO_P0_8_Low();
            brcnt1 = 0;
        }
        brcnt1++;
    }

    if(brc1rf) {
        if(brcnt1 >= 58) GPIO_P0_8_High();
        //if(brcnt1 >= (60 - brc1rf)) GPIO_P0_8_High();
        if(brcnt1 >= 60) {
            GPIO_P0_8_Low();
            brcnt1 = 0;
        }
        brcnt1++;
    }

    if(brc2cf) {
        if(brcnt2 >= 2) GPIO_P0_9_High();
        if(brcnt2 >= 7) {
            GPIO_P0_9_Low();
            brcnt2 = 0;
        }
        brcnt2++;
    }

    if(brc2rf) {
        if(brcnt2 >= 58) GPIO_P0_9_High();
        //if(brcnt2 >= (60 - brc2rf)) GPIO_P0_9_High();
        if(brcnt2 >= 60) {
            GPIO_P0_9_Low();
            brcnt2 = 0;
        }
        brcnt2++;
    }
```

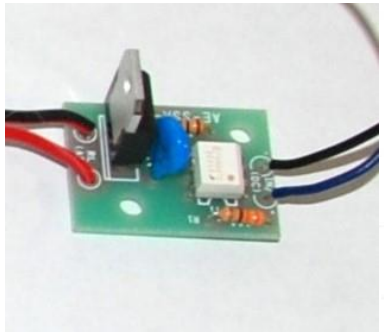
SSRキットを使ってモーターを直接駆動してみる。

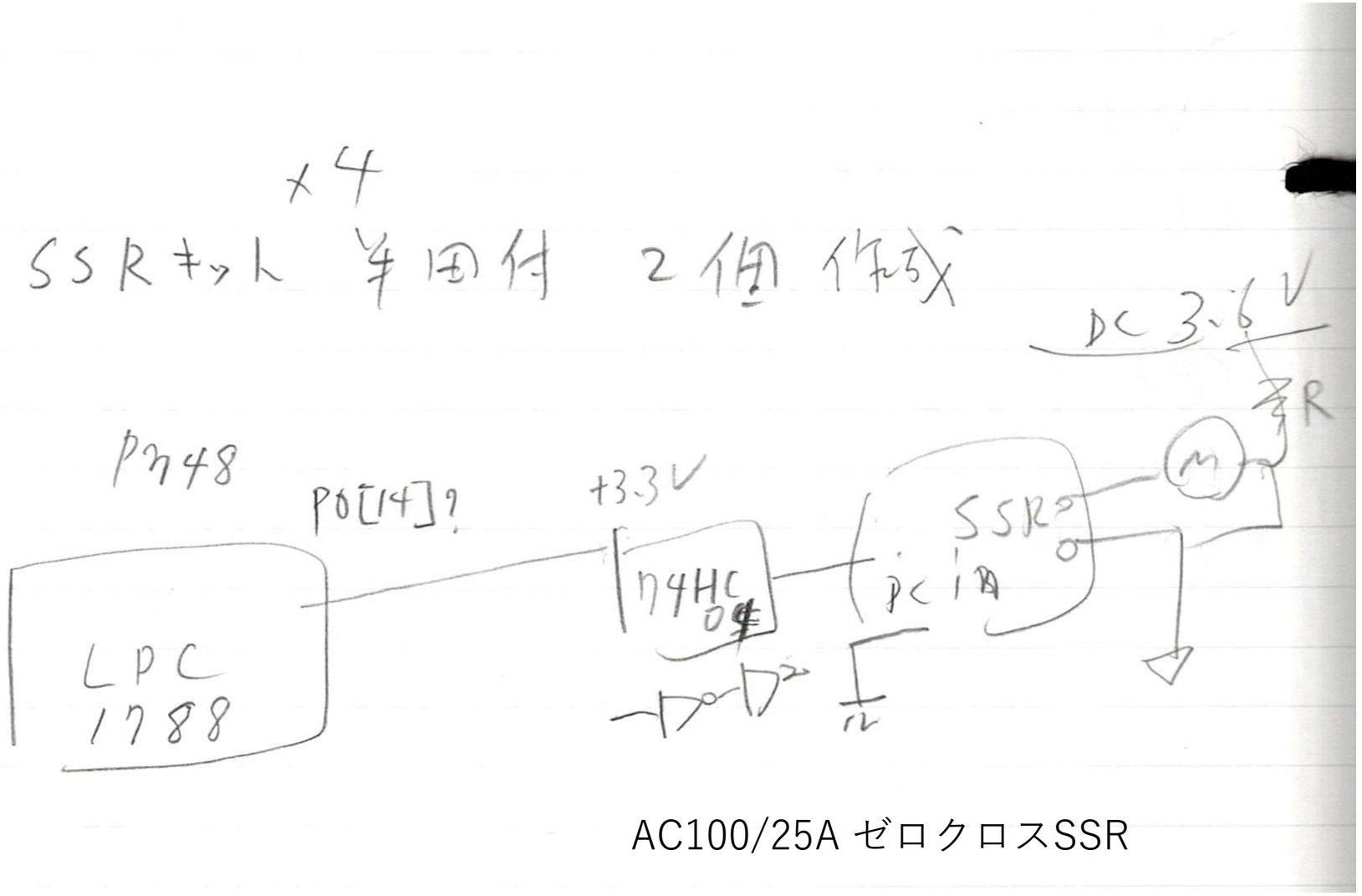
AC 100V 接続 OK
DCはNG / ONのみ



SSRキットを使ってモーターを直接駆動してみる。

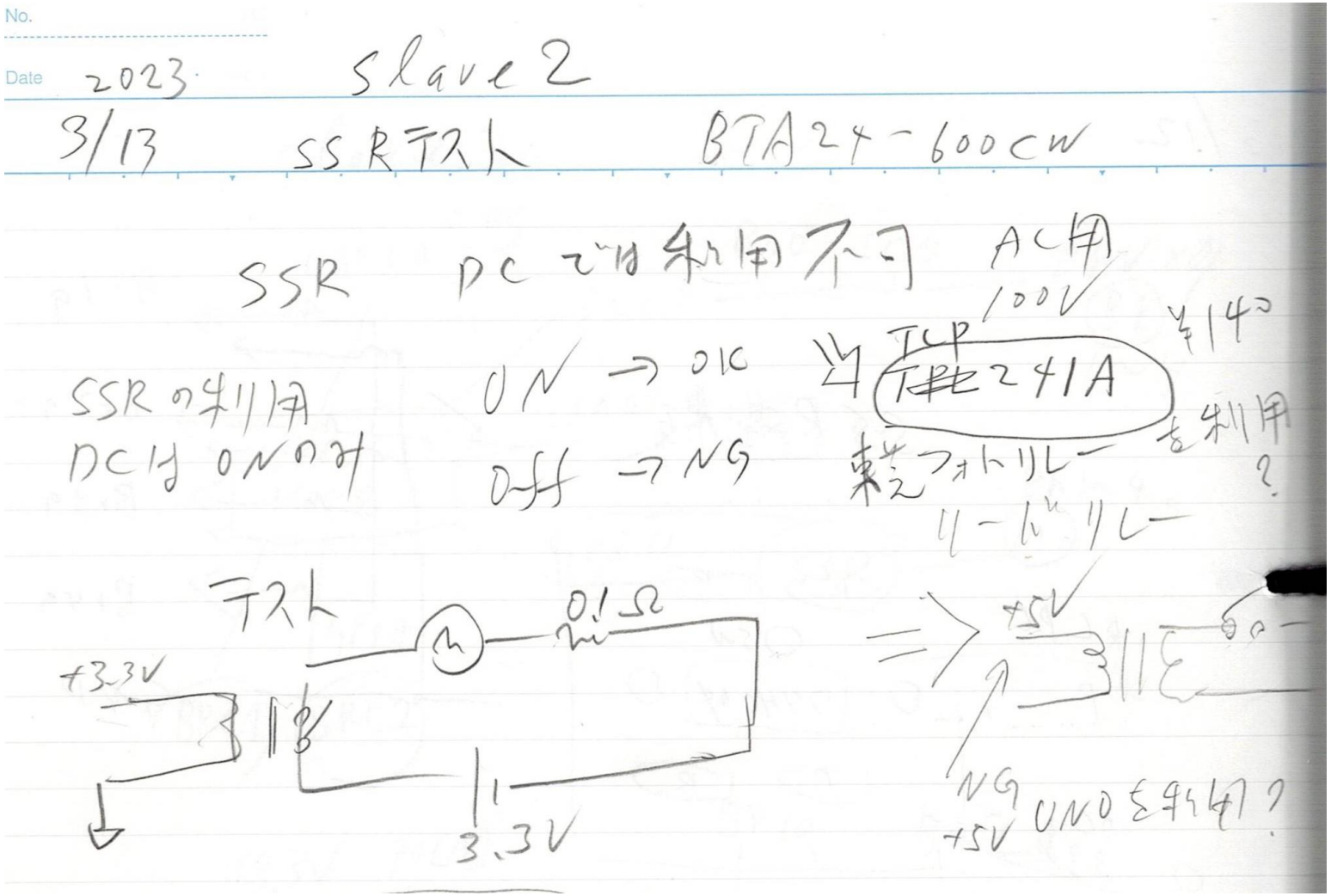






AC100/25A ゼロクロスSSR

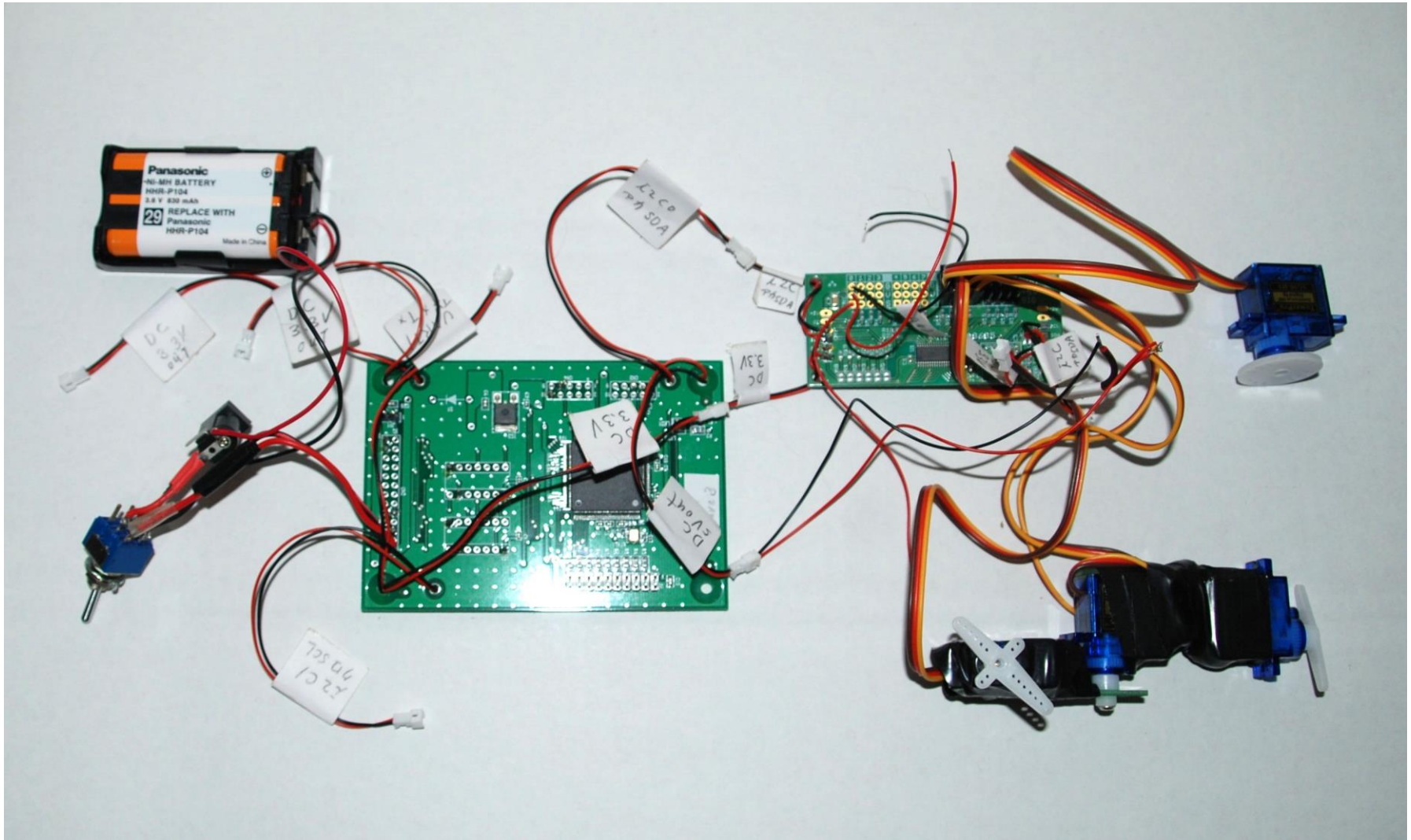
SSRキットを使ってモーターを直接駆動してみる。
(今回のキットTLP561J利用では、DCでの利用は不可、AC接続はOK)



NXP AE-PCA9685

Micro Servo test

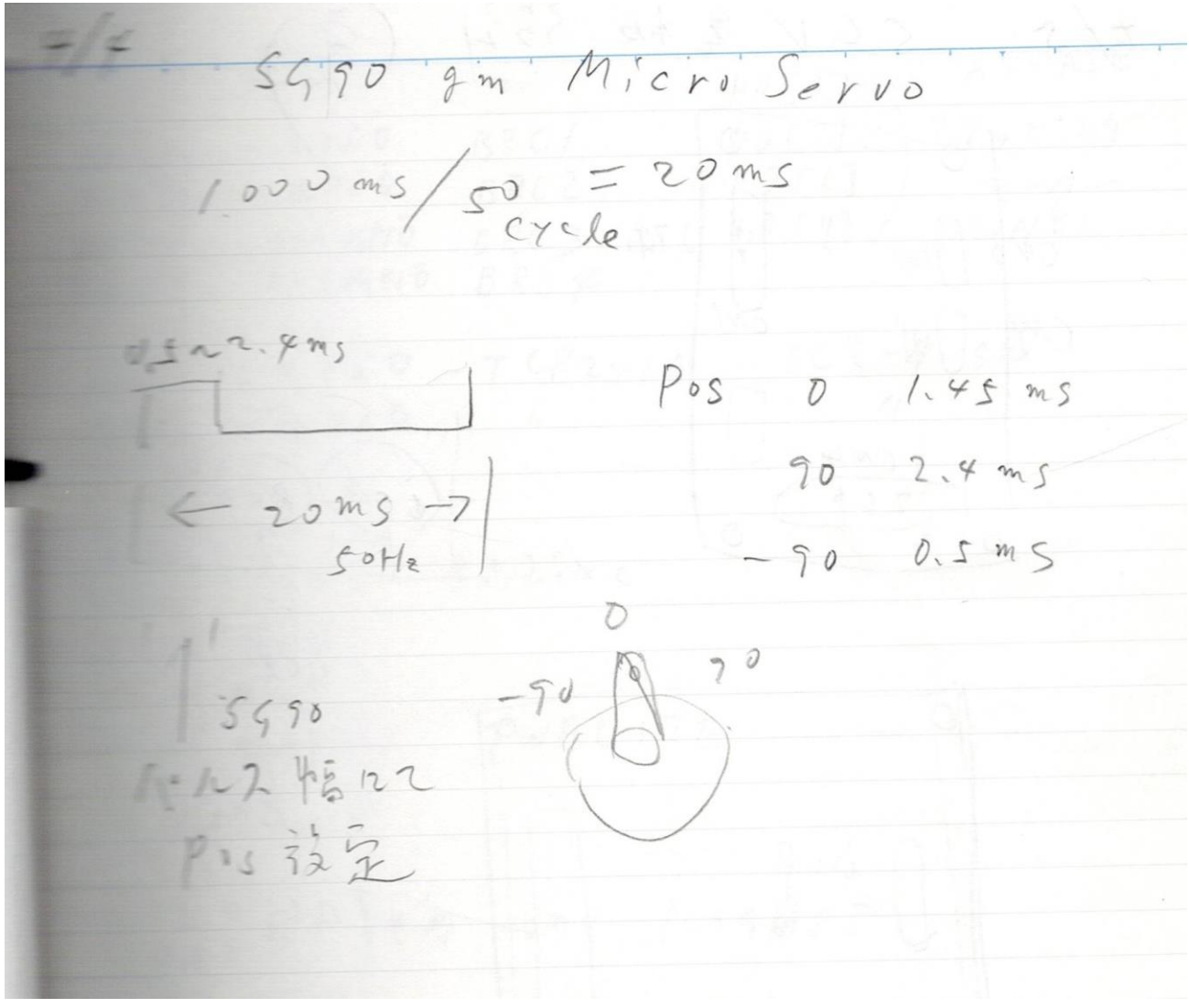
D-Slave3



NXP AE-PCA9685

Micro Servo test

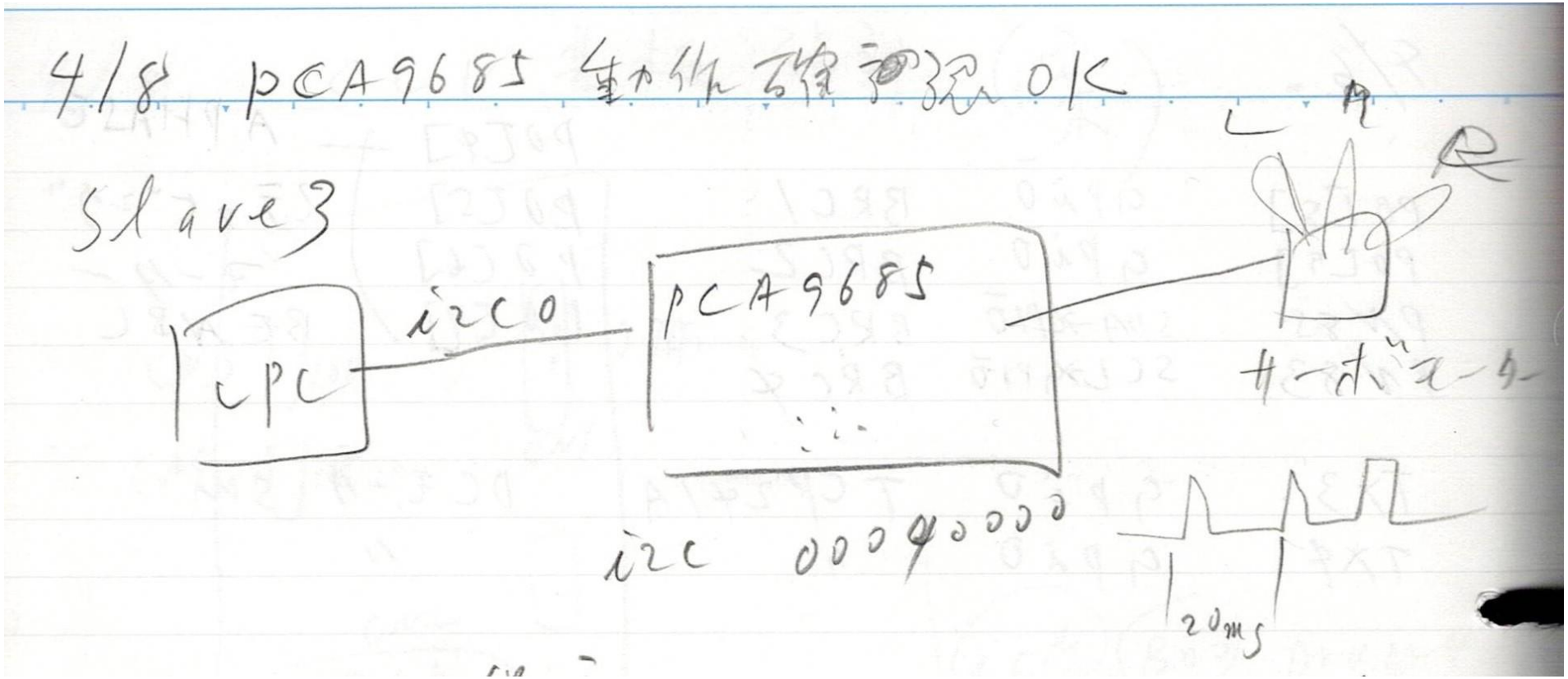
D-Slave3 i2c0 i2c0_PCA9685_0_tst();

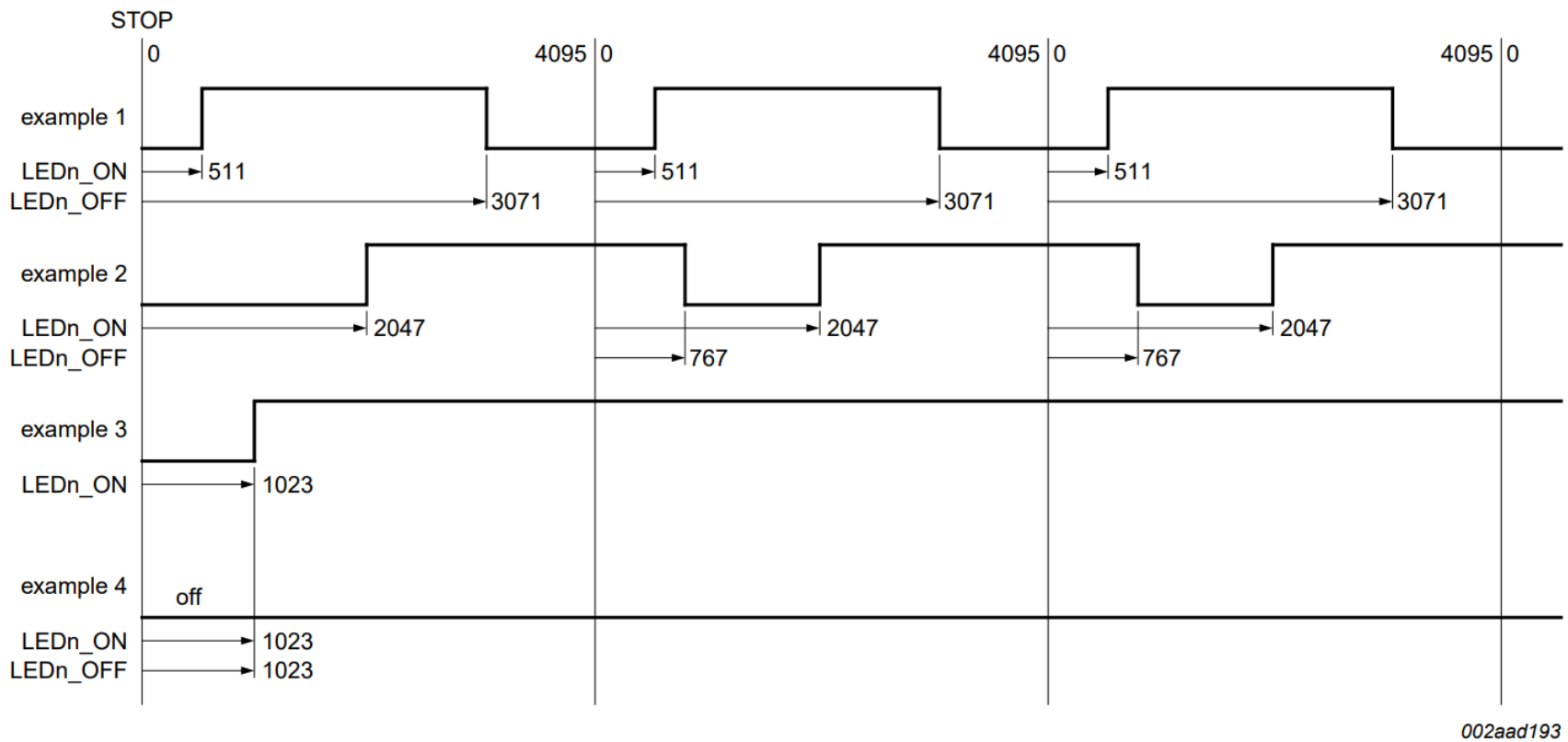


NXP AE-PCA9685

Micro Servo test

D-Slave3





Example 1: LEDn_ON (511) < LEDn_OFF (3071)

Example 2: LEDn_ON (2047) > LEDn_OFF (767)

Example 3: LEDn_ON[12] = 1; LEDn_ON[11:0] = 1022; LEDn_OFF[12] = 0; LEDn_OFF[11:0] = don't care

Example 4: LEDn_ON[12] = 0; LEDn_OFF[12] = 0; LEDn_ON[11:0] = LEDn_OFF[11:0]

Fig 9. Output example

Example 1: (assumes that the LED0 output is used and (delay time) + (PWM duty cycle) ≤ 100 %)

Delay time = 10 %; PWM duty cycle = 20 % (LED on time = 20 %; LED off time = 80 %).

Delay time = 10 % = 409.6 ~ 410 counts = 19Ah.

Since the counter starts at 0 and ends at 4095, we will subtract 1, so delay time = 199h counts.

LED0_ON_H = 1h; LED0_ON_L = 99h (LED start turn on after this delay count to 409)

LED on time = 20 % = 819.2 ~ 819 counts.

LED off time = 4CCh (decimal 410 + 819 − 1 = 1228)

LED0_OFF_H = 4h; LED0_OFF_L = CCh (LED start turn off after this count to 1228)

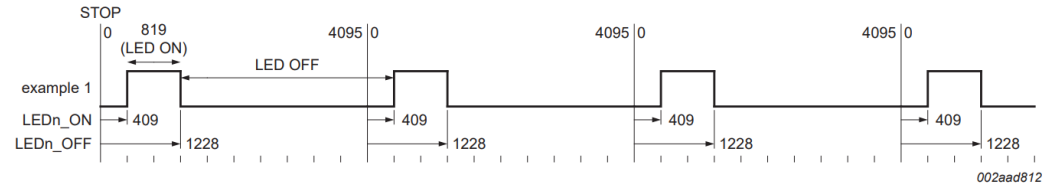


Fig 7. LED output, example 1

Example 2: (assumes that the LED4 output is used and (delay time) + (PWM duty cycle > 100 %)

Delay time = 90 %; PWM duty cycle = 90 % (LED on time = 90 %; LED off time = 10 %).

Delay time = 90 % = 3686.4 ~ 3686 counts − 1 = 3685 = E65h.

LED4_ON_H = Eh; LED4_ON_L = 65h (LED start turn on after this delay count to 3685)

LED on time = 90 % = 3686 counts.

Since the delay time and LED on period of the duty cycle is greater than 4096 counts, the LEDn_OFF count will occur in the next frame. Therefore, 4096 is subtracted from the LEDn_OFF count to get the correct LEDn_OFF count. See Figure 9, Figure 10 and Figure 11.

LED off time = CCBh (decimal 3685 + 3686 = 7372 − 4096 = 3275)

LED4_OFF_H = Ch; LED4_OFF_L = CBh (LED start turn off after this count to 3275)

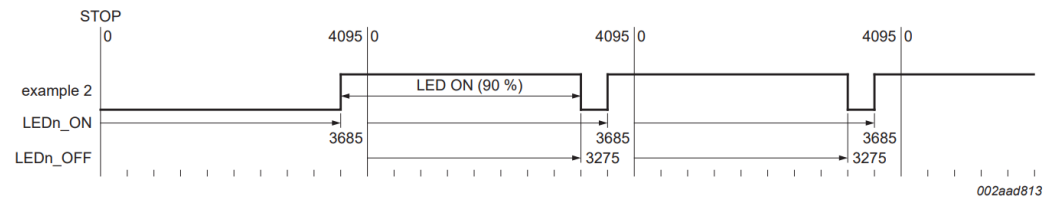
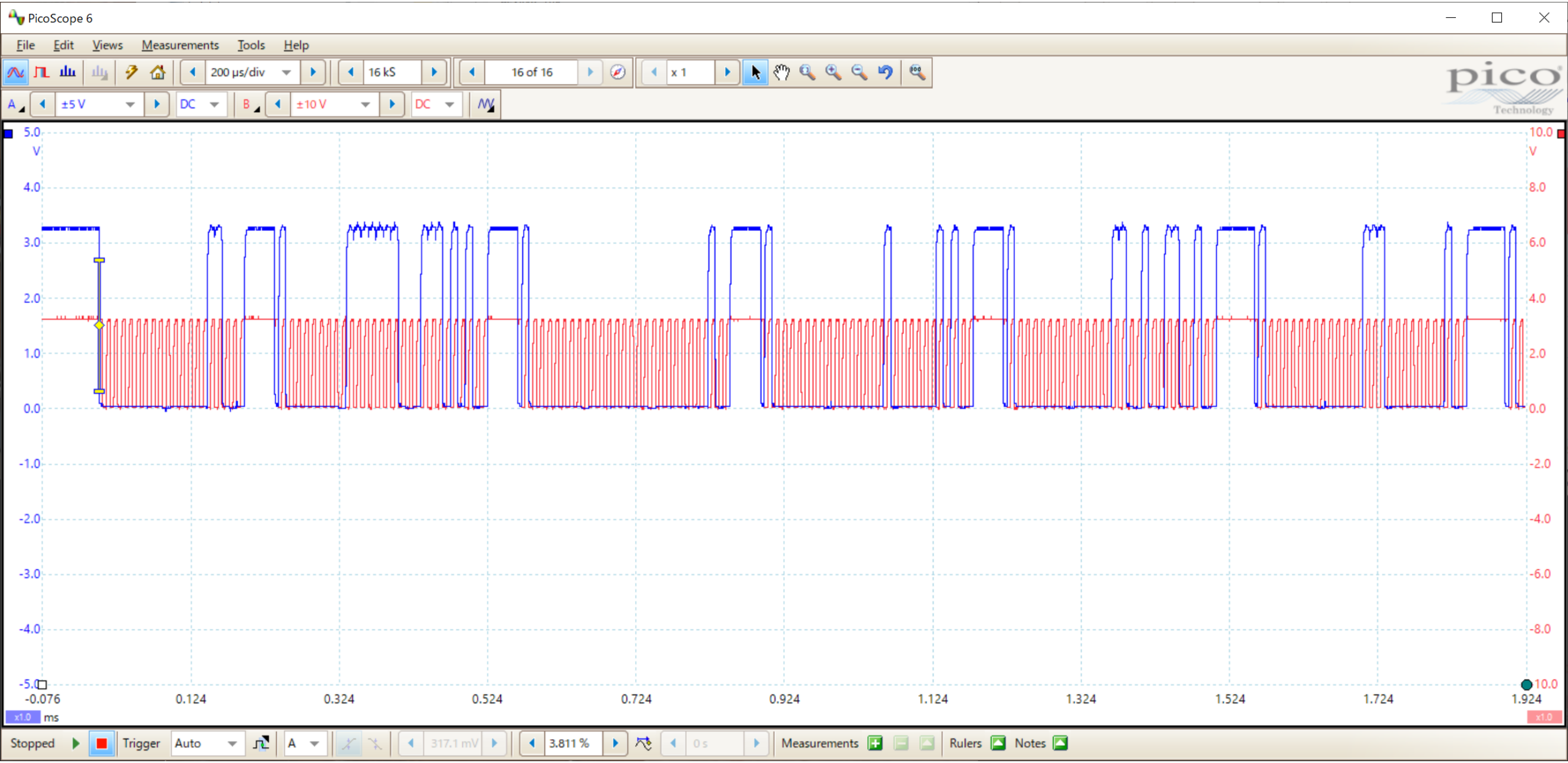
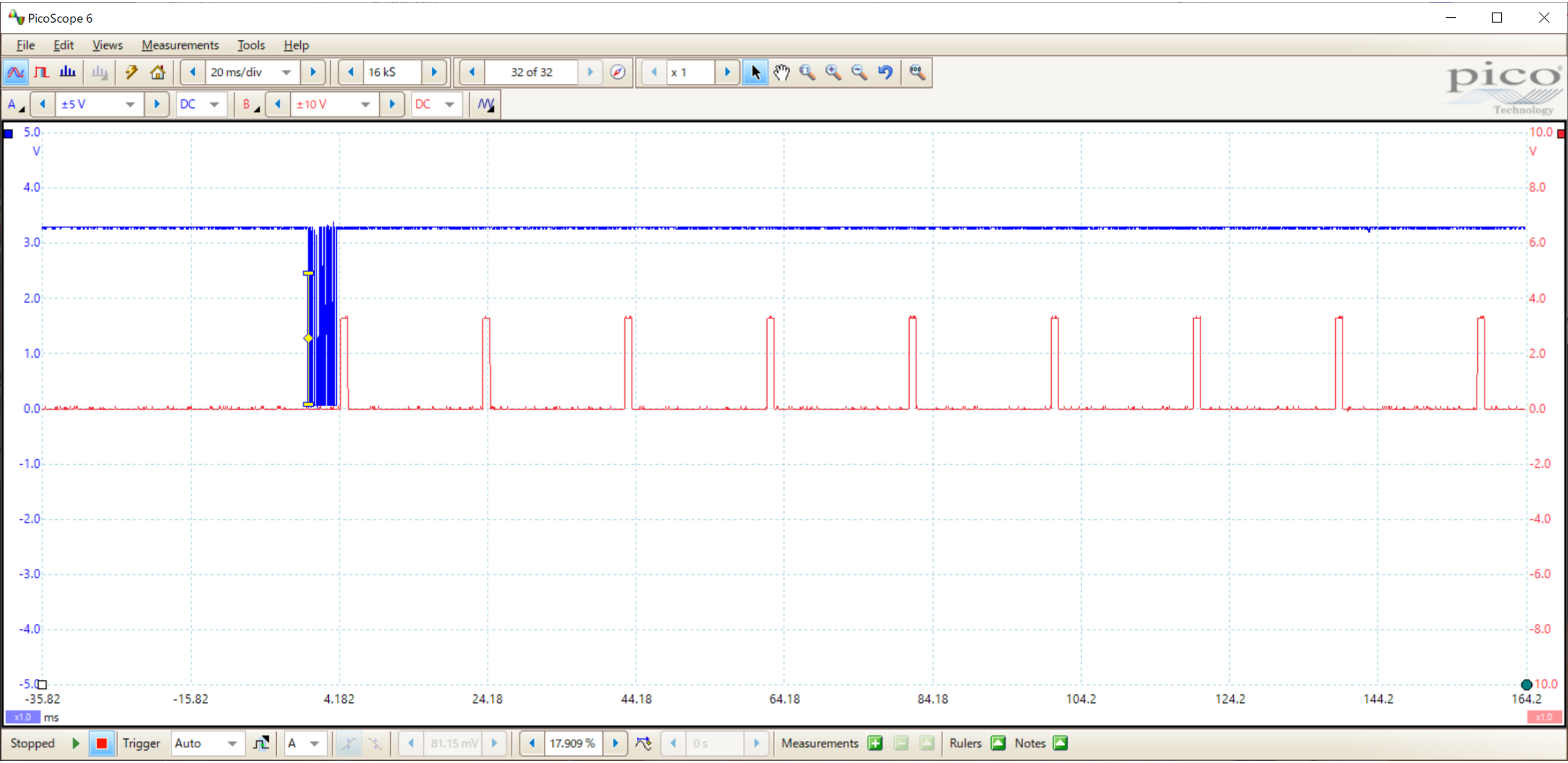


Fig 8. LED output, example 2


PicoScope 6
File
Edit
Views
Measurements
Tools
Help
200 µs/div
16 kS
16 of 16
x 1
A
±5 V
DC
B
±10 V
DC
pico
Technology
5.0
V
4.0
3.0
2.0
1.0
0.0
-1.0
-2.0
-3.0
-4.0
-5.0
10.0
V
8.0
6.0
4.0
2.0
0.0
-2.0
-4.0
-6.0
-8.0
10.0
-0.076
0.124
0.324
0.524
0.724
0.924
1.124
1.324
1.524
1.724
1.924
x1.0
ms
x1.0
Stopped
Trigger
Auto
A
317.1 mV
3.811 %
0 s
Measurements
Rulers
Notes

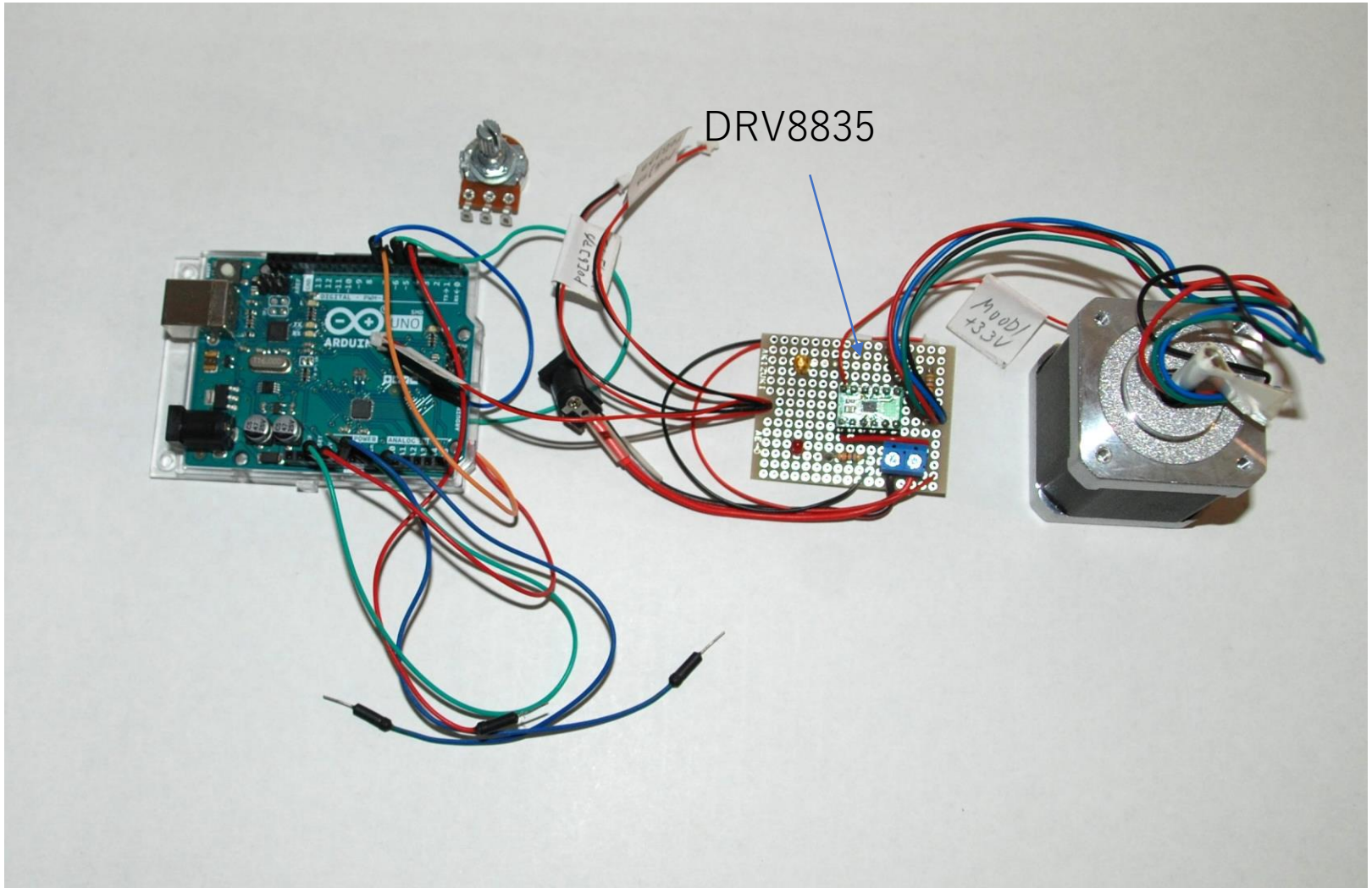

PicoScope 6
File
Edit
Views
Measurements
Tools
Help
20 ms/div
16 kS
32 of 32
x 1
A
±5 V
DC
B
±10 V
DC
pico
Technology
5.0
V
4.0
3.0
2.0
1.0
0.0
-1.0
-2.0
-3.0
-4.0
-5.0
10.0
V
8.0
6.0
4.0
2.0
0.0
-2.0
-4.0
-6.0
-8.0
10.0
-35.82
-15.82
4.182
24.18
44.18
64.18
84.18
104.2
124.2
144.2
164.2
x1.0
ms
x1.0
Stopped
Trigger
Auto
A
81.15 mV
17.909 %
0 s
Measurements
Rulers
Notes

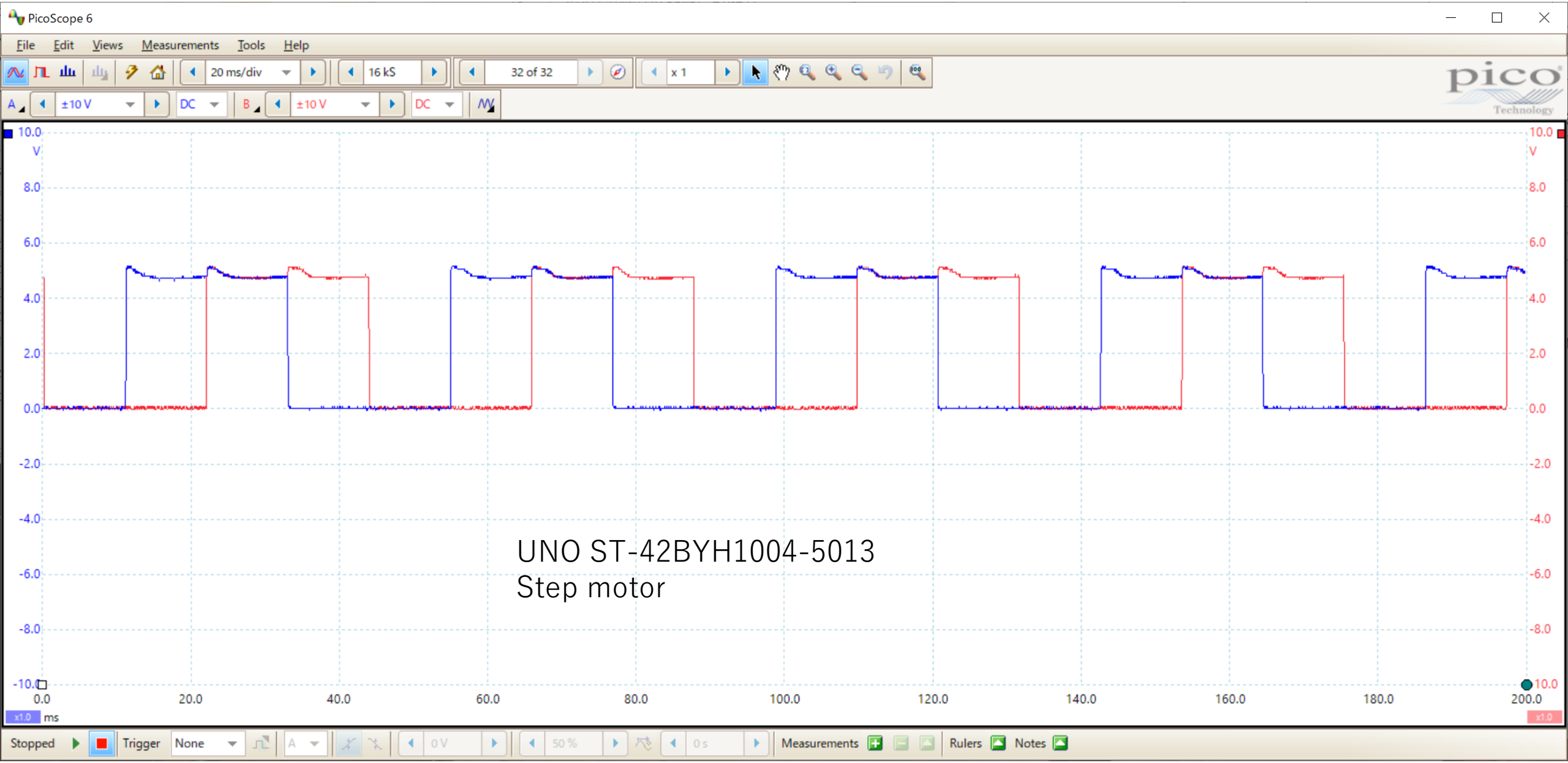
```c
/*
// PCA9685 i2c0 2023.4.7 h.  ---
*/
#include "includes.h"
#include "playlpc.h"
#include "drone.h"

int i2c0_PCA9685_0_tst()
{

    int  i;

    unsigned char pAddr;
    unsigned char gAddr; // general
    unsigned char pMsg[8];


    gAddr = 0x00;  // General call address
    pMsg[0] = 0x06;  // Software Reset
    I2C_MasterWrite (gAddr, pMsg, 1);
    for( i=0; i < 1000; i++); //delay

    pAddr = 0x40;  // OK PCA9685 address default

    pMsg[0] = 0xFE; //  PRE_SCALE address
    pMsg[1] = 0x75; //  set PWM 50Hz
    //pMsg[1] = 0x75; //  set PWM 1256Hz
    I2C_MasterWrite (pAddr, pMsg, 2);
    for( i=0; i < 1000; i++); //delay

    pMsg[0] = 0x00; //  Mode1 reg. address
    pMsg[1] = 0x01; //  SLEEP Nomal, ALLCALL responds
    I2C_MasterWrite (pAddr, pMsg, 2);
    for( i=0; i < 1000; i++); //delay

    pMsg[0] = 0x01; //  Mode2 reg. address
    pMsg[1] = 0x05; //  OUTDRV 1 , OUTNE 01
    I2C_MasterWrite (pAddr, pMsg, 2);
    for( i=0; i < 1000; i++); //delay

    for(i=0; i<1; i++) {
        pMsg[0] = 0x06;   // LED0 ON_L
        pMsg[1] = 0x99;   //
        I2C_MasterWrite (pAddr, pMsg, 2);
        for( i=0; i < 1000; i++); //delay

        pMsg[0] = 0x07;   // LED0_ON_H
        pMsg[1] = 0x01;   //
        I2C_MasterWrite (pAddr, pMsg, 2);
        for( i=0; i < 1000; i++); //delay

        pMsg[0] = 0x08;   // LED0_OFF_L
        pMsg[1] = 0x6C;   // middle
        //pMsg[1] = 0x00;   // right
        //pMsg[1] = 0xfC;   // left
        I2C_MasterWrite (pAddr, pMsg, 2);
        for( i=0; i < 1000; i++); //delay

        pMsg[0] = 0x09;   // LED0_OFF_H
        pMsg[1] = 0x02;   //
        I2C_MasterWrite (pAddr, pMsg, 2);
        for( i=0; i < 1000; i++); //delay
    }

    for(i=0; i<1; i++) {
        pMsg[0] = 0x26;   // LED8 ON_L
        pMsg[1] = 0x99;   //
        I2C_MasterWrite (pAddr, pMsg, 2);
        for( i=0; i < 1000; i++); //delay

        pMsg[0] = 0x27;   // LED8_ON_H
        pMsg[1] = 0x01;   //
        I2C_MasterWrite (pAddr, pMsg, 2);
        for( i=0; i < 1000; i++); //delay

        pMsg[0] = 0x28;   // LED8_OFF_L
        pMsg[1] = 0x6C;   // middle
        //pMsg[1] = 0x00;   // right
        //pMsg[1] = 0xfC;   // left
        I2C_MasterWrite (pAddr, pMsg, 2);
        for( i=0; i < 1000; i++); //delay

        pMsg[0] = 0x29;   // LED8_OFF_H
        pMsg[1] = 0x02;   //
        I2C_MasterWrite (pAddr, pMsg, 2);
        for( i=0; i < 1000; i++); //delay
    }

    for(i=0; i<1; i++) {
        pMsg[0] = 0x2a;   // LED9 ON_L
        pMsg[1] = 0x99;   //
        I2C_MasterWrite (pAddr, pMsg, 2);
        for( i=0; i < 1000; i++); //delay

        pMsg[0] = 0x2b;   // LED9_ON_H
        pMsg[1] = 0x01;   //
        I2C_MasterWrite (pAddr, pMsg, 2);
        for( i=0; i < 1000; i++); //delay

        pMsg[0] = 0x2c;   // LED9_OFF_L
        pMsg[1] = 0x6C;   // middle
        //pMsg[1] = 0x00;   // right
        //pMsg[1] = 0xfC;   // left
        I2C_MasterWrite (pAddr, pMsg, 2);
        for( i=0; i < 1000; i++); //delay

        pMsg[0] = 0x2d;   // LED9_OFF_H
        pMsg[1] = 0x02;   //
        I2C_MasterWrite (pAddr, pMsg, 2);
        for( i=0; i < 1000; i++); //delay
    }

    for(i=0; i<1; i++) {
        pMsg[0] = 0x2e;   // LED10 ON_L
        pMsg[1] = 0x99;   //
        I2C_MasterWrite (pAddr, pMsg, 2);
        for( i=0; i < 1000; i++); //delay

        pMsg[0] = 0x2f;   // LED10_ON_H
        pMsg[1] = 0x01;   //
        I2C_MasterWrite (pAddr, pMsg, 2);
        for( i=0; i < 1000; i++); //delay

        pMsg[0] = 0x30;   // LED10_OFF_L
        pMsg[1] = 0x6C;   // middle
        //pMsg[1] = 0x00;   // right
        //pMsg[1] = 0xfC;   // left
        I2C_MasterWrite (pAddr, pMsg, 2);
        for( i=0; i < 1000; i++); //delay

        pMsg[0] = 0x31;   // LED10_OFF_H
        pMsg[1] = 0x02;   //
        I2C_MasterWrite (pAddr, pMsg, 2);
        for( i=0; i < 1000; i++); //delay
    }

    for(i=0; i<1; i++) {
        pMsg[0] = 0x32;   // LED11 ON_L
        pMsg[1] = 0x99;   //
        I2C_MasterWrite (pAddr, pMsg, 2);
        for( i=0; i < 1000; i++); //delay

        pMsg[0] = 0x33;   // LED11_ON_H
        pMsg[1] = 0x01;   //
        I2C_MasterWrite (pAddr, pMsg, 2);
        for( i=0; i < 1000; i++); //delay

        pMsg[0] = 0x34;   // LED11_OFF_L
        pMsg[1] = 0x6C;   // middle
        //pMsg[1] = 0x00;   // right
        //pMsg[1] = 0xfC;   // left
        I2C_MasterWrite (pAddr, pMsg, 2);
        for( i=0; i < 1000; i++); //delay

        pMsg[0] = 0x35;   // LED11_OFF_H
        pMsg[1] = 0x02;   //
        I2C_MasterWrite (pAddr, pMsg, 2);
        for( i=0; i < 1000; i++); //delay
    }

    for( i=0; i < 10000000; i++); //delay
    SMove_tst8(pAddr);
    SMove_tst9(pAddr);
    SMove_tst10(pAddr);
    SMove_tst11(pAddr);

    // Soft were Reset
    for( i=0; i < 20000000; i++); //delay
    gAddr = 0x00;  // General call address
    pMsg[0] = 0x06;  // Software Reset
    I2C_MasterWrite (gAddr, pMsg, 1);
    for( i=0; i < 1000; i++); //delay

    /*
    // Mode2 reg.
    pMsg[0] = 0x01; //  Mode2 reg. address
    pMsg[1] = 0x05; //  OUTDRV 1 , OUTNE 01
    I2C_MasterWrite (pAddr, pMsg, 2);
    for( i=0; i < 1000; i++); //delay

    // Mode1 reg.
    pMsg[0] = 0x00; //  Mode1 reg. address
    pMsg[1] = 0x00; //  SLEEP ON, ALLCALL responds
    I2C_MasterWrite (pAddr, pMsg, 2);
    for( i=0; i < 1000; i++); //delay
    */

    return(0);
}
```

Texas DRV8835

Steper motor バイポーラ駆動

ST-42-BYH1004-5013 uno test




PicoScope 6
File Edit Views Measurements Tools Help
20 ms/div
16 kS
32 of 32
x 1
A ±10 V DC
B ±10 V DC
pico Technology
10.0 V
8.0
6.0
4.0
2.0
0.0
-2.0
-4.0
-6.0
-8.0
-10.0
0.0 20.0 40.0 60.0 80.0 100.0 120.0 140.0 160.0 180.0 200.0
ms
UNO ST-42BYH1004-5013
Step motor
Stopped
Trigger None
Measurements
Rulers
Notes

DRV8835_STEP_SAMPLE | Arduino IDE 2.2.1

File Edit Sketch Tools Help

Select Board

DRV8835_STEP_SAMPLE.ino

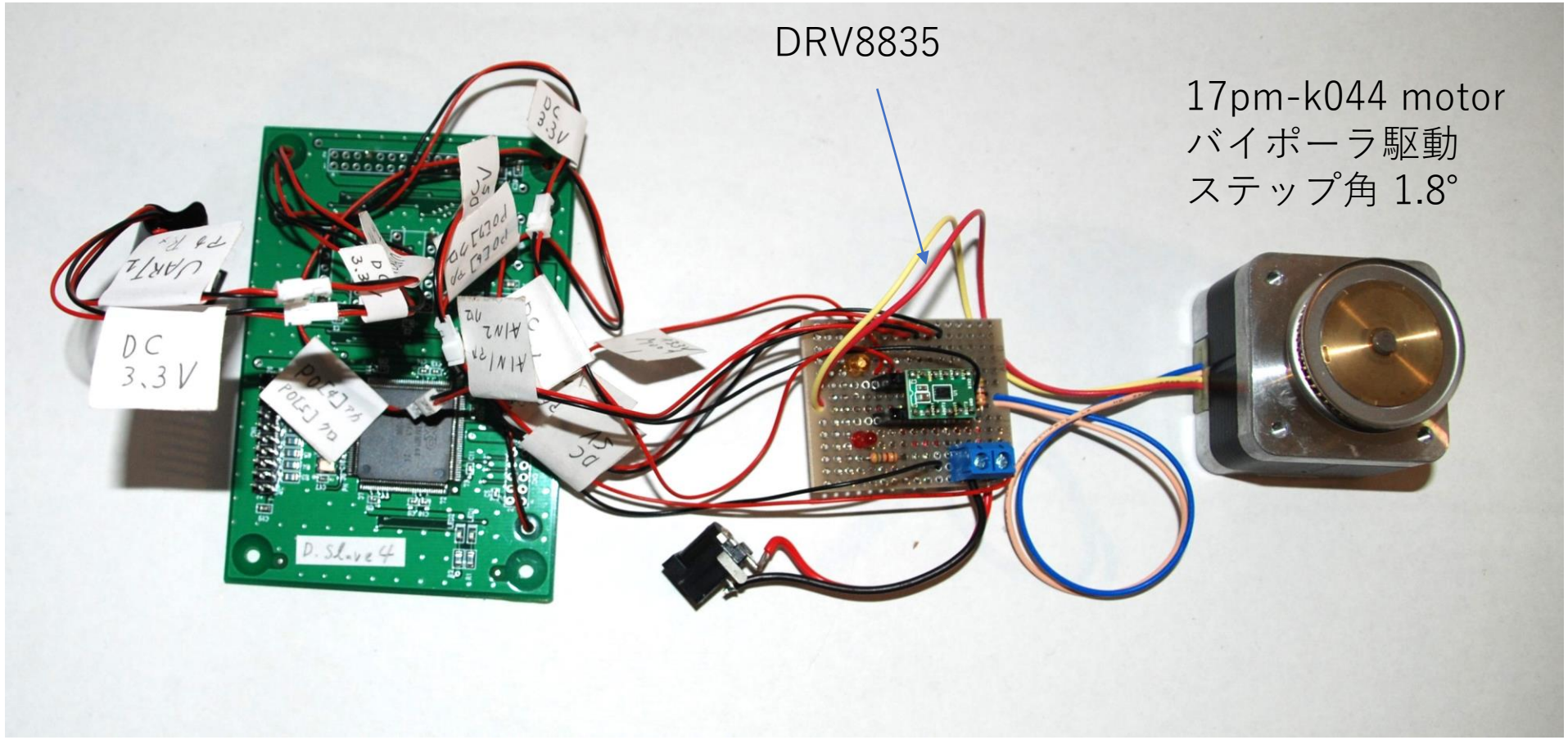
```
int APHASE  = 4;
int AENBL   = 5;
int BPHASE  = 6;
int BENBL   = 7;
int VR_PIN  = A0;
unsigned long VR_VALUE = 0;

void setup()
{
  pinMode(APHASE,OUTPUT);
  pinMode(AENBL,OUTPUT);
  pinMode(BPHASE,OUTPUT);
  pinMode(BENBL,OUTPUT);
  digitalWrite(AENBL,HIGH);
  digitalWrite(BENBL,HIGH);
  //digitalWrite(AENBL,LOW);
  //digitalWrite(BENBL,LOW);
}
void READ_VR(void)
{
  VR_VALUE = analogRead(VR_PIN);
}
void DELAY_WAIT(void)
{
  for (int i = 0; i < (VR_VALUE /10 + 7) ; i++) delayMicroseconds(100);
}
void loop()
{
  READ_VR();
  digitalWrite(APHASE,HIGH);
  DELAY_WAIT();
  digitalWrite(BPHASE,HIGH);
  DELAY_WAIT();
  digitalWrite(APHASE,LOW);
  DELAY_WAIT();
  digitalWrite(BPHASE,LOW);
  DELAY_WAIT();
}

```

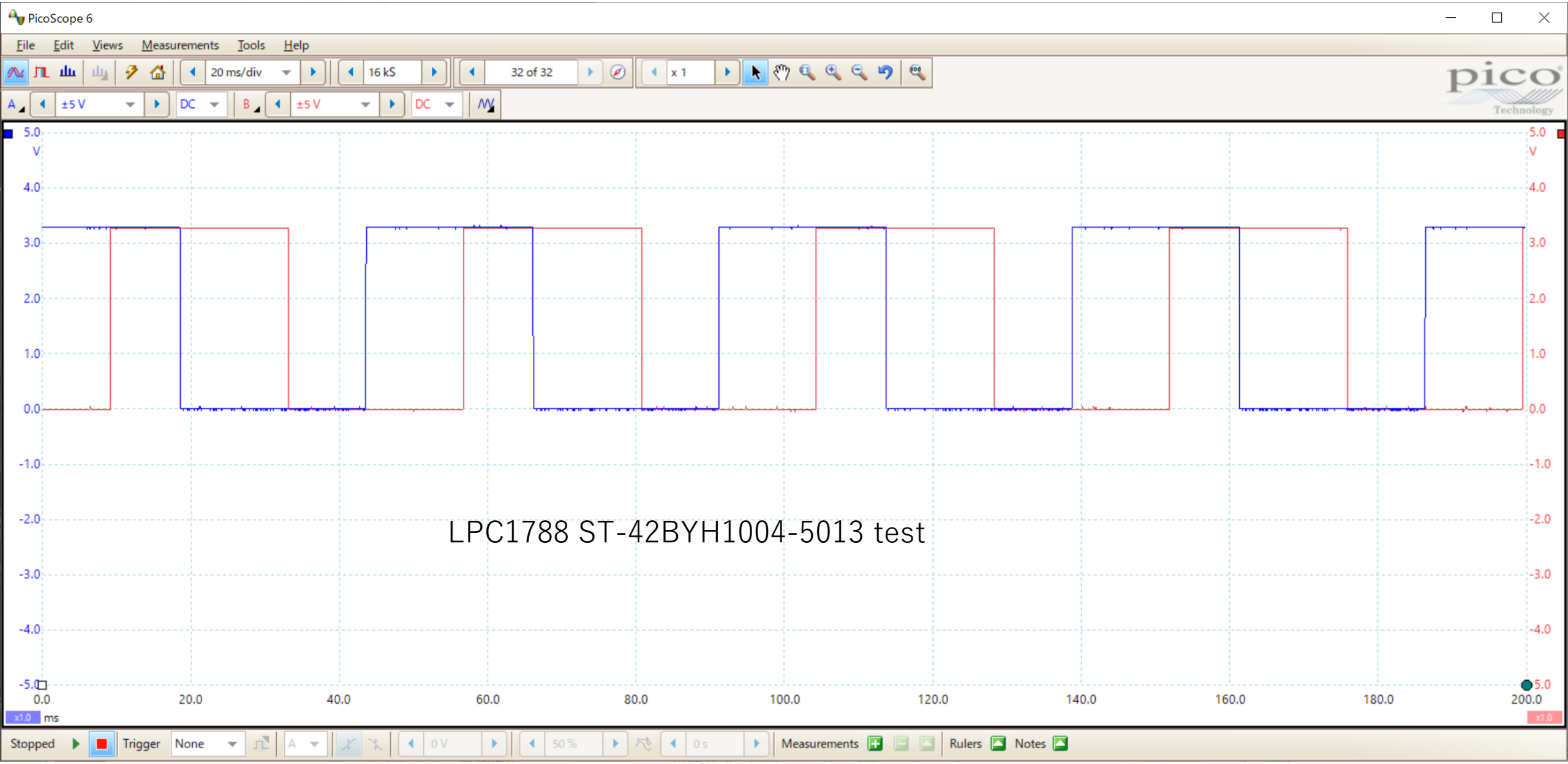
Texas DRV8835

Steper motor バイポーラ駆動

17pm_k044_akz_a LPC1788 test

D-Slave4




PicoScope 6
File Edit Views Measurements Tools Help
20 ms/div
16 kS
32 of 32
x 1
A ±5 V DC
B ±5 V DC
pico Technology
LPC1788 ST-42BYH1004-5013 test
5.0 V
4.0
3.0
2.0
1.0
0.0
-1.0
-2.0
-3.0
-4.0
-5.0
0.0 20.0 40.0 60.0 80.0 100.0 120.0 140.0 160.0 180.0 200.0
ms
Stopped
Trigger None
Measurements
Rulers
Notes

```
// GPIO IOCON
#define IOCON_P0_4      ((volatile unsigned int *)(0x4002C010)) // Gstop=1
#define IOCON_P0_5      ((volatile unsigned int *)(0x4002C014)) // Pose
#define IOCON_P0_6      ((volatile unsigned int *)(0x4002C018)) // Gstop=4
#define IOCON_P0_7      ((volatile unsigned int *)(0x4002C01C)) // Stop
#define IOCON_P0_8      ((volatile unsigned int *)(0x4002C020))
#define IOCON_P0_9      ((volatile unsigned int *)(0x4002C024))
#define IOCON_P0_14     ((volatile unsigned int *)(0x4002C038)) // GPIO
#define IOCON_P1_18     ((volatile unsigned int *)(0x4002C0C8)) // LED2
```

```
*IOCON_P0_4 = 0x10;  // Set P0[4] Pull up enabled -> GPIO DRV8835 APHASE
*IOCON_P0_5 = 0x10;  // Set P0[5] Pull up enabled -> GPIO DRV8835 AENBL
*IOCON_P0_6 = 0x10;  // Set P0[6] Pull up enabled -> GPIO DRV8835 BPHASE
*IOCON_P0_7 = 0x10;  // Set P0[7] Pull up enavled -> GPIO DRV8835 BENBL
*IOCON_P0_8 = 0x00;          // Set P0[8] -> GPIO BRC1
*IOCON_P0_9 = 0x00;          // Set P0[9] -> GPIO BRC2
*IOCON_P0_14 = 0x00;         // Set P0[14] -> GPIO

//*FGPIO_FIO0DIR = 0x00004300;  // GPIO P0[8] P0[9] P0[14] OutPut
//*FGPIO_FIO0DIR = 0x00000100;  // GPIO P0[8] OutPut
//*FGPIO_FIO0DIR = 0x00000200;  // GPIO P0[9] OutPut
*FGPIO_FIO0DIR = 0x000043f0;  // p0[4] [5] [6] [7] [8] [9] [14] OutPut

//*FGPIO_FIO0MASK = 0x00004000;  // P0[14] Mask test
gpio_drv8835_tst();       // stepping moter test
```

```
int gpio_drv8835_tst()
{
    int  i,j;

    GPIO_P0_4_Low();  // DRV8835 APHASE Low
    GPIO_P0_5_High(); // DRV8835 AENBL high
    GPIO_P0_6_Low();  // DRV8835 BPHASE Low
    GPIO_P0_7_High(); // DRV8835 BENBL high


    // Right turn test
    for(j=0 ; j<26; j++) {
        GPIO_P0_4_High();  // DRV8835 APHASE high
        for( i=0; i < 180000; i++); //delay

        GPIO_P0_6_High();  // DRV8835 BPHASE high
        for( i=0; i < 180000; i++); //delay

        GPIO_P0_4_Low();  // DRV8835 APHASE Low
        for( i=0; i < 200000; i++); //delay

        GPIO_P0_6_Low();  // DRV8835 BPHASE Low
        for( i=0; i < 200000; i++); //delay
    }

    for( i=0; i < 4000000; i++); //delay

    GPIO_P0_4_Low();  // DRV8835 APHASE Low
    GPIO_P0_5_High(); // DRV8835 AENBL high
    GPIO_P0_6_Low();  // DRV8835 BPHASE Low
    GPIO_P0_7_High(); // DRV8835 BENBL high

    // Left turn test
    for(j=0 ; j<26; j++) {
        GPIO_P0_6_High();  // DRV8835 BPHASE high
        for( i=0; i < 180000; i++); //delay

        GPIO_P0_4_High();  // DRV8835 APHASE high
        for( i=0; i < 180000; i++); //delay

        GPIO_P0_6_Low();  // DRV8835 BPHASE Low
        for( i=0; i < 200000; i++); //delay

        GPIO_P0_4_Low();  // DRV8835 APHASE Low
        for( i=0; i < 200000; i++); //delay
    }

    GPIO_P0_4_Low();  // DRV8835 APHASE Low
    GPIO_P0_5_Low();  // DRV8835 AENBL Low
    GPIO_P0_6_Low();  // DRV8835 BPHASE Low
    GPIO_P0_7_Low();  // DRV8835 BENBL Low
    //while(1); test
    return(0);
}
```

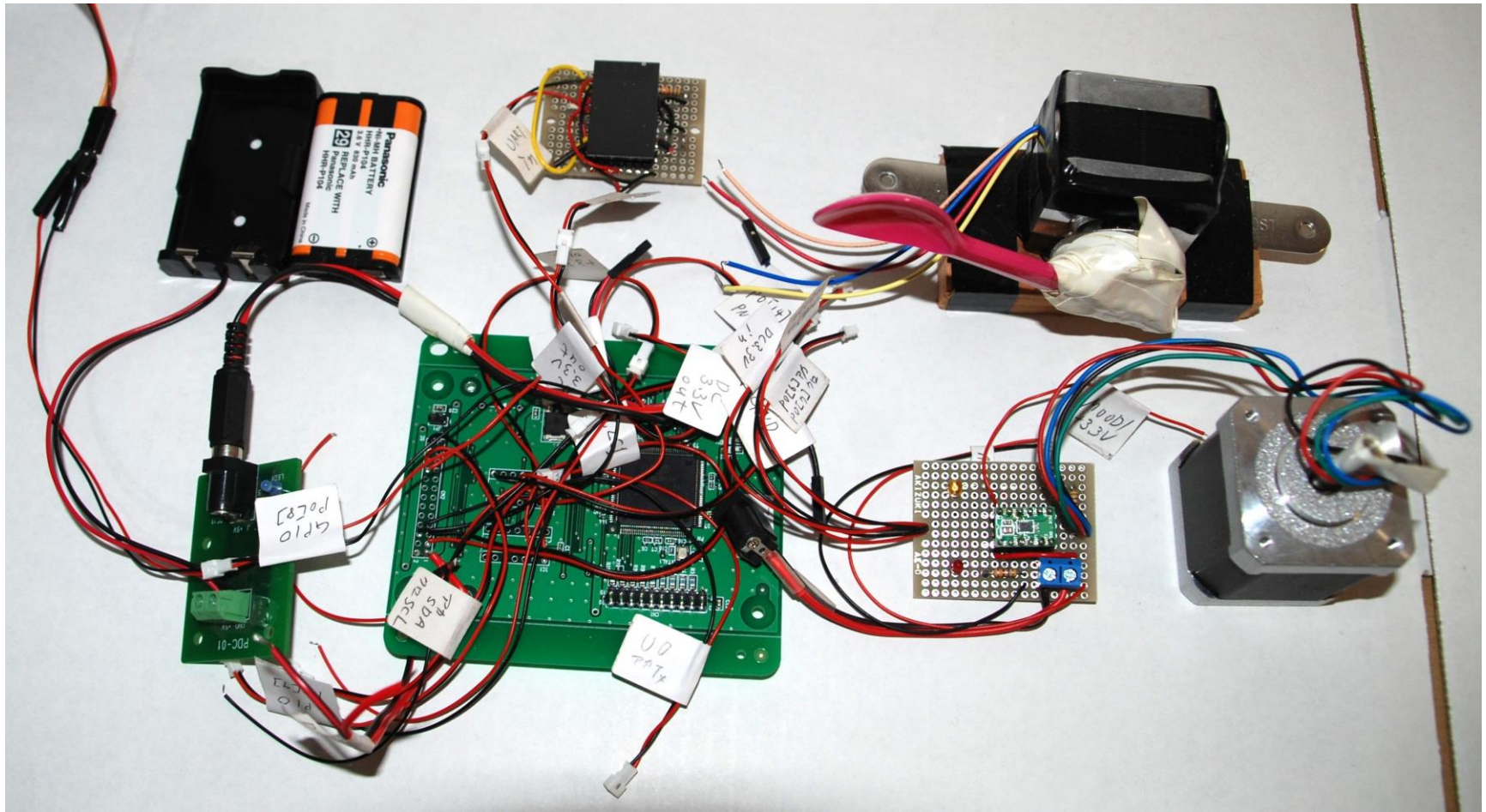
Texas DRV8835

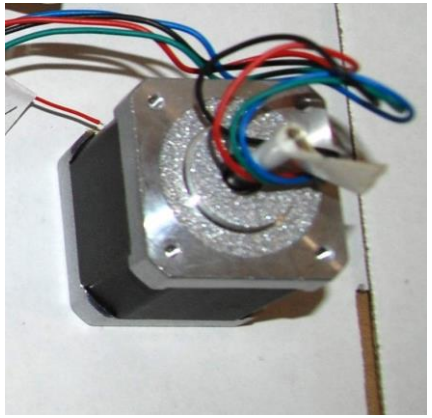
Step motor test D-Slave4

LPC1788 p0[4] p0[5] p0[6] p0[7]

Twelite midi test

Mercury motor
ST-42BYH1004-5013
2Phase 0.9° 32/21



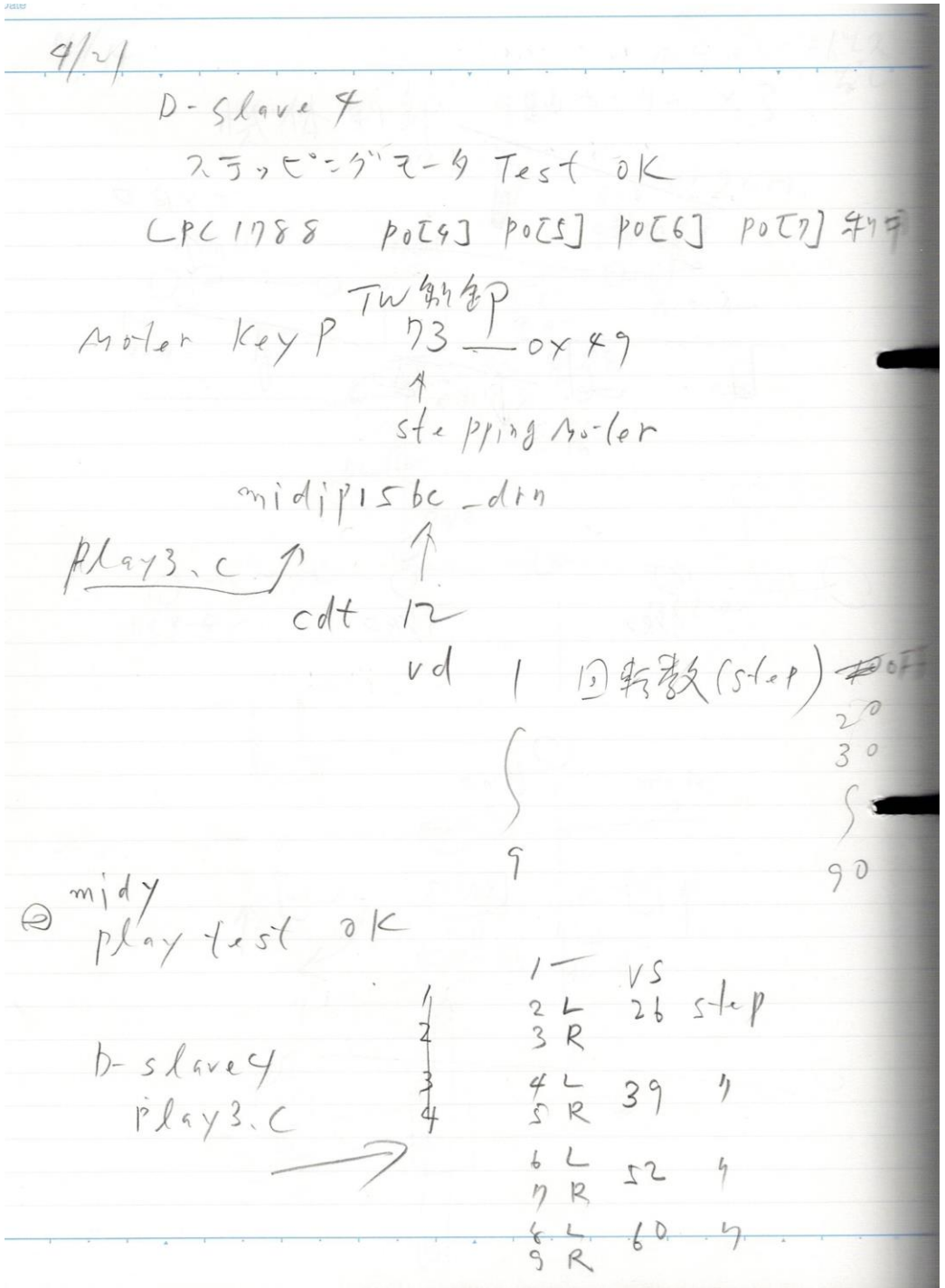


Texas DRV8835

Step motor test D-Slave4

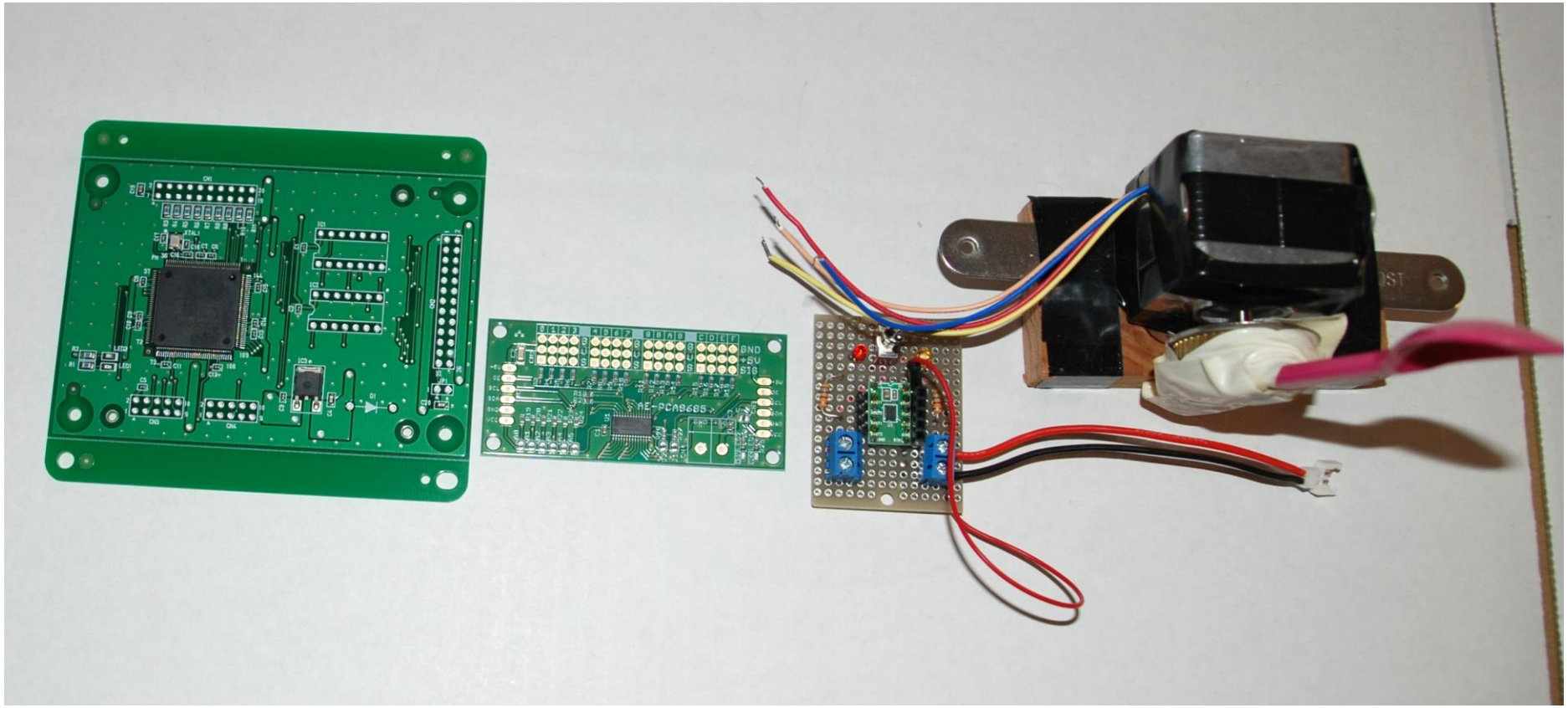
LPC1788 p0[4] p0[5] p0[6] p0[7]

```
int midiP15bc_drn(vd)
int  vd;
{
    switch(vd) {
        case  0 : //
                  break;
        case  1 : //
                  break;
        case  2 : gpio_drv8835_L(26);  // 26 step Left turn
                  break;
        case  3 : gpio_drv8835_R(26);  // 26 step Right turn
                  break;
        case  4 : gpio_drv8835_L(39);
                  break;
        case  5 : gpio_drv8835_R(39);
                  break;
        case  6 : gpio_drv8835_L(52);
                  break;
        case  7 : gpio_drv8835_R(52);
                  break;
        case  8 : gpio_drv8835_L(60);
                  break;
        case  9 : gpio_drv8835_R(60);
                  break;
        default : //
                  break;
    }
    //gpio_drv8835_tst();       // stepping moter test
    return(0);
}
```



LPC1788-PCA9685-DRV8835-17PM test

D-Slave5

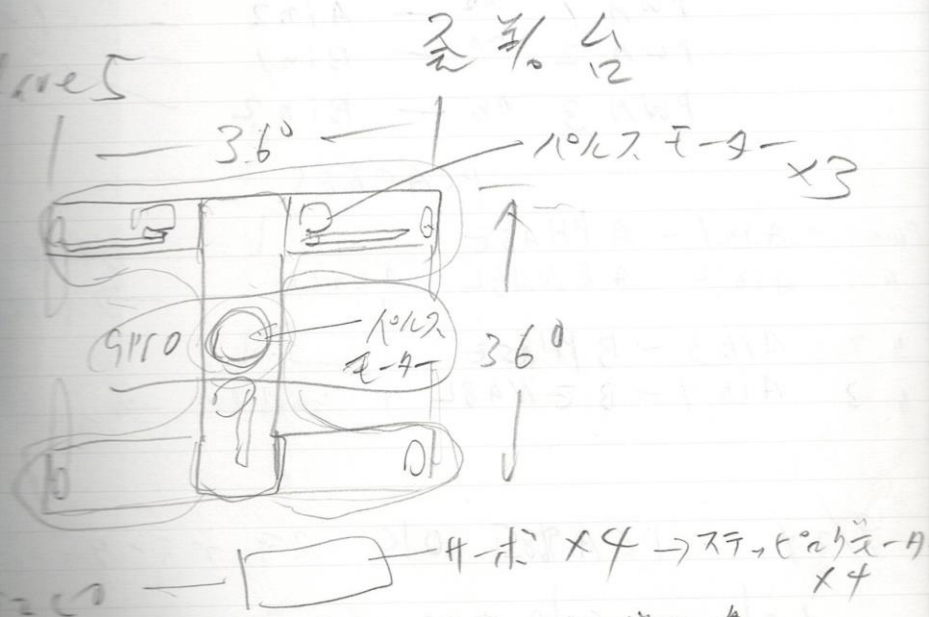


TWE LITE UART

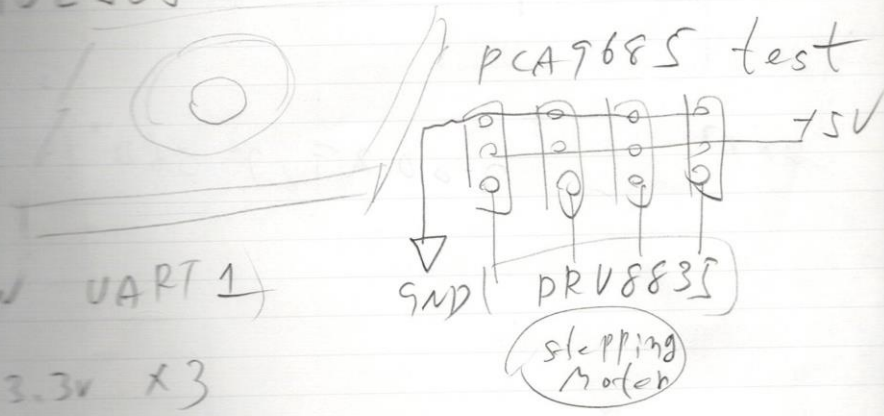
TW Blue ×2 セット設定

b 19200 kBPS
0 00010100

D-Slave5



- 電源 — ステッピングモーター
- GPIO [9] [5] [6] [7] — ステッピングモーター



- TW UART 1)
- DC 3.3V ×3
- DC 1V ×1

4/23 PCA9685 I²C Slave 5 test

PCA9685		DRV8835	
PWM 0	アカ	—	Ain1
PWM 1	クロ	—	Ain2
PWM 2	アカ	—	Bin1
PWM 3	クロ	—	Bin2

ステッピングモーター ×3

PCA9685

PWM0 - Ain1 - A PHASE
" 1 Ain2 A ENABL 1 : Enable 0
→ DRV8835 → full ON

" 2 Ain3 - B PHASE
" 3 Ain4 - B ENABL 1 : Enable → full ON

4/24 PCA9685 I²C ステッピングモーター

d c	b a 9 8	7 6 5 4	3 2 1 0
12	11 10		

0x10

200ステップ 360°（1回転）

9/25

200ステップ → 360°
?

40ステップ用 Function

17PM-K044-AKZ

ステップ角 1.8deg

360° → 200

40 × 5 = 200

i2c0 - PCA9685

4 _Forward()

5 _Reverse()

4/29

D-Slave 5 接続テスト

ステップモーター OK 5台

DRV8835 基板×5 作成

TWELITE

M Serve ~ TW ~ D-Slave 5

cdt vd

13 - step motor gpio

14

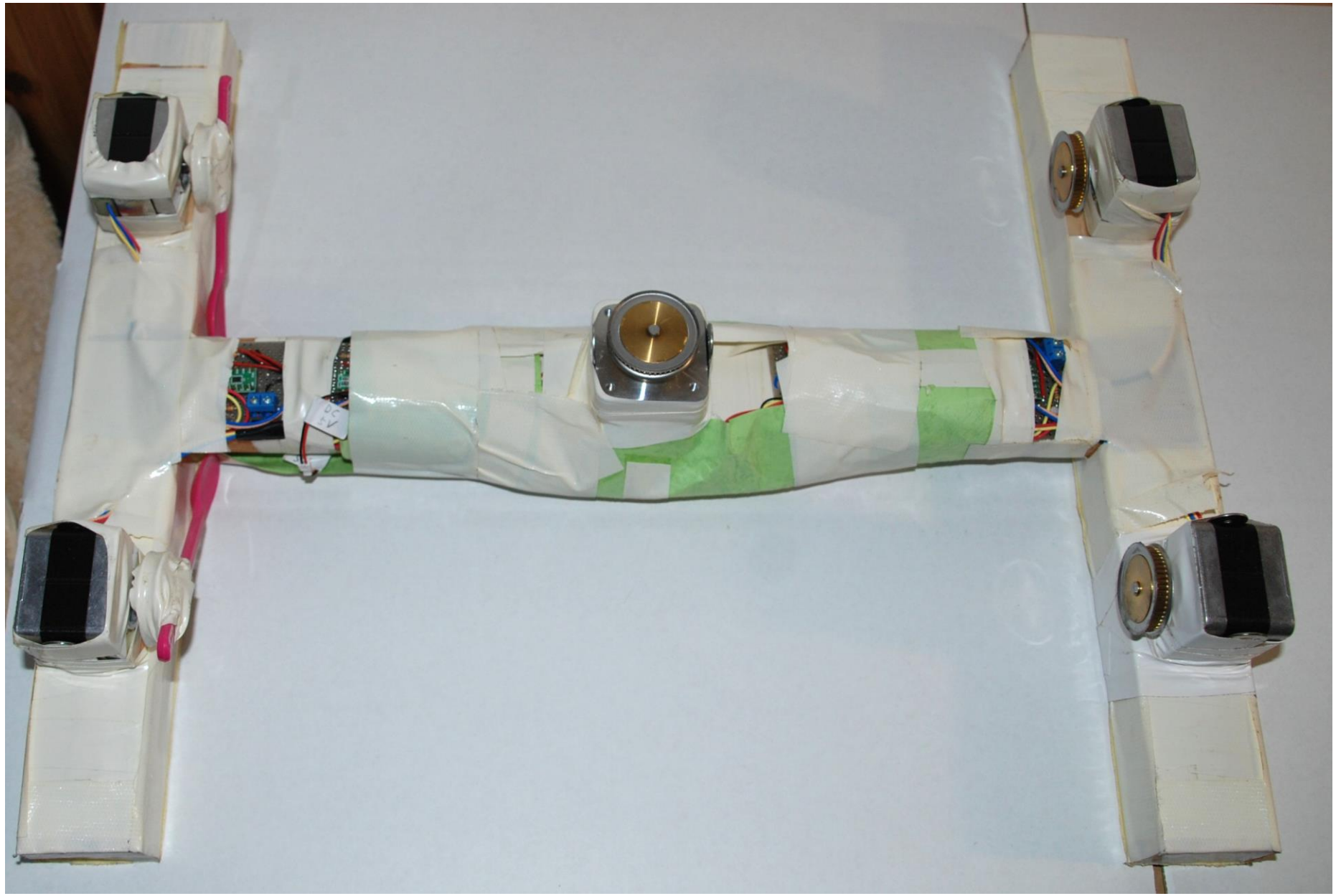
15

val

Motion 1~9

1
2
3
4
5
6
7
8
9

D-Slaves

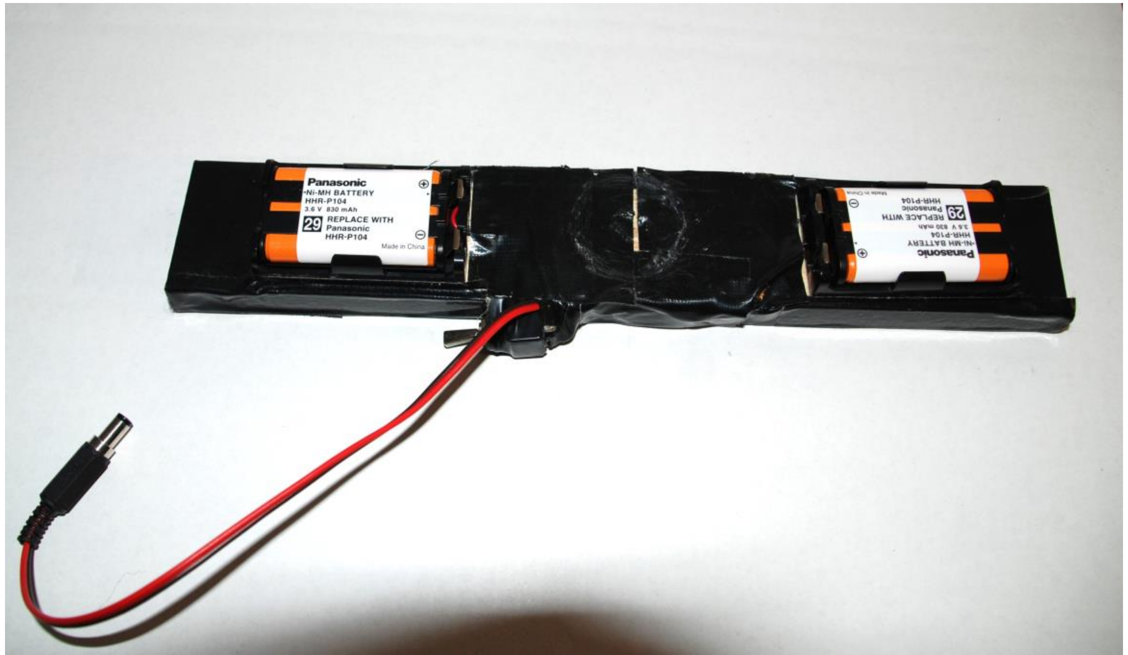


5/1

D-Slave5
完成

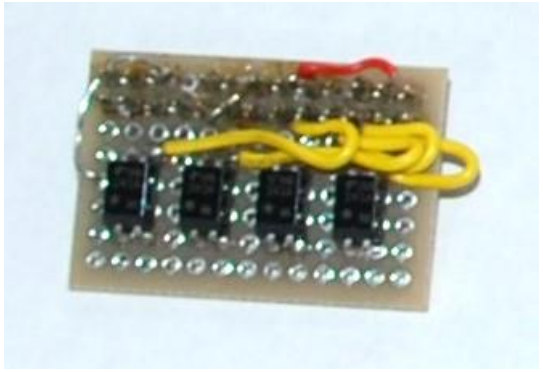
MServer Web —TW→ D-slave5
OK

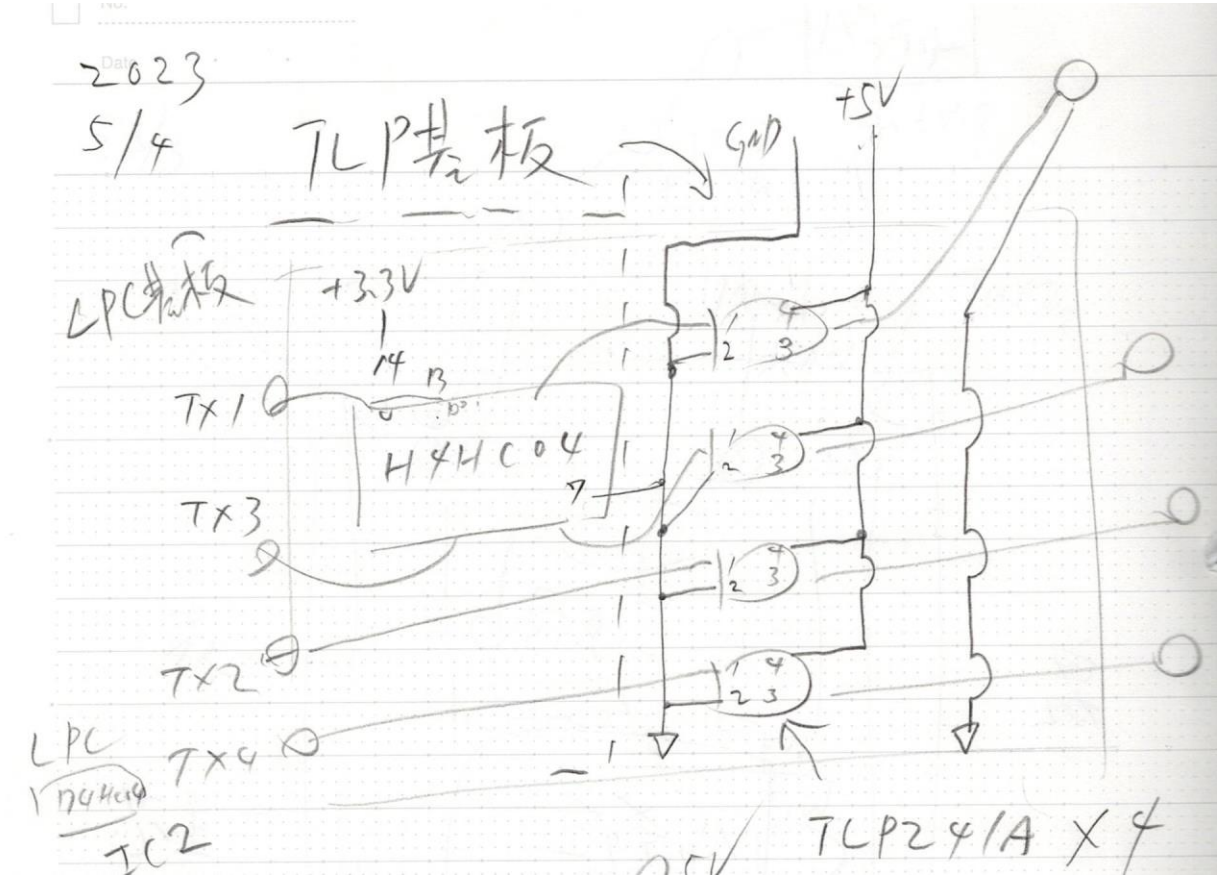
スタンド作成



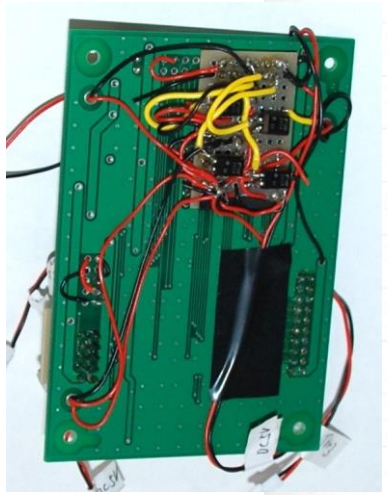

Roland
XP

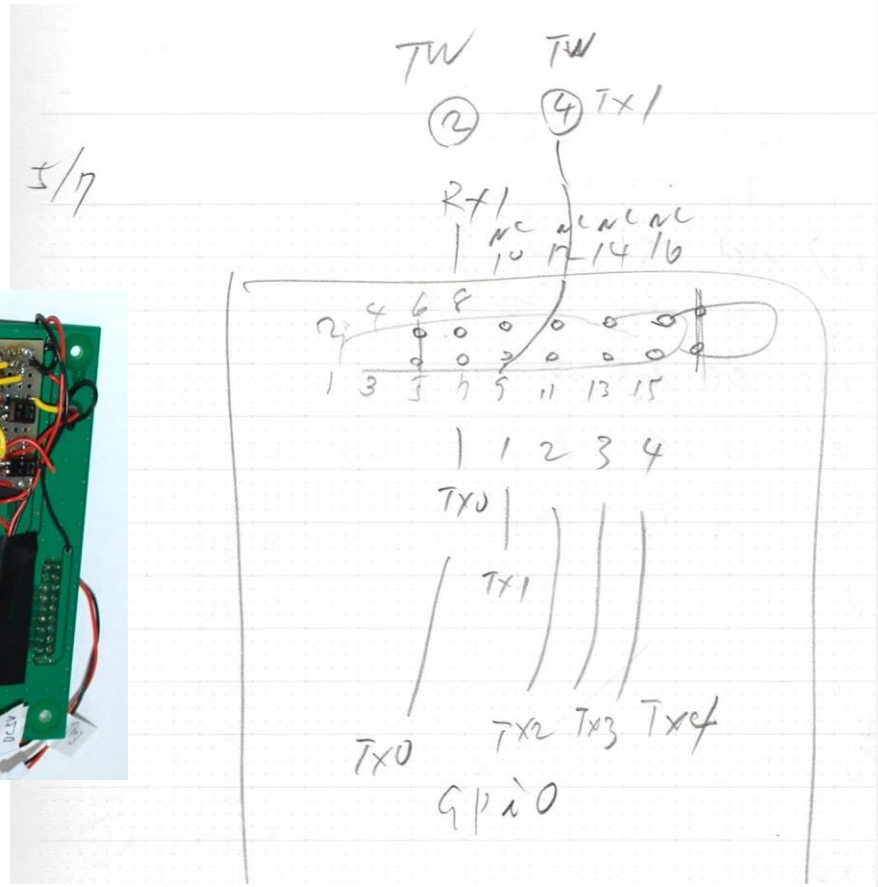
TLP基板

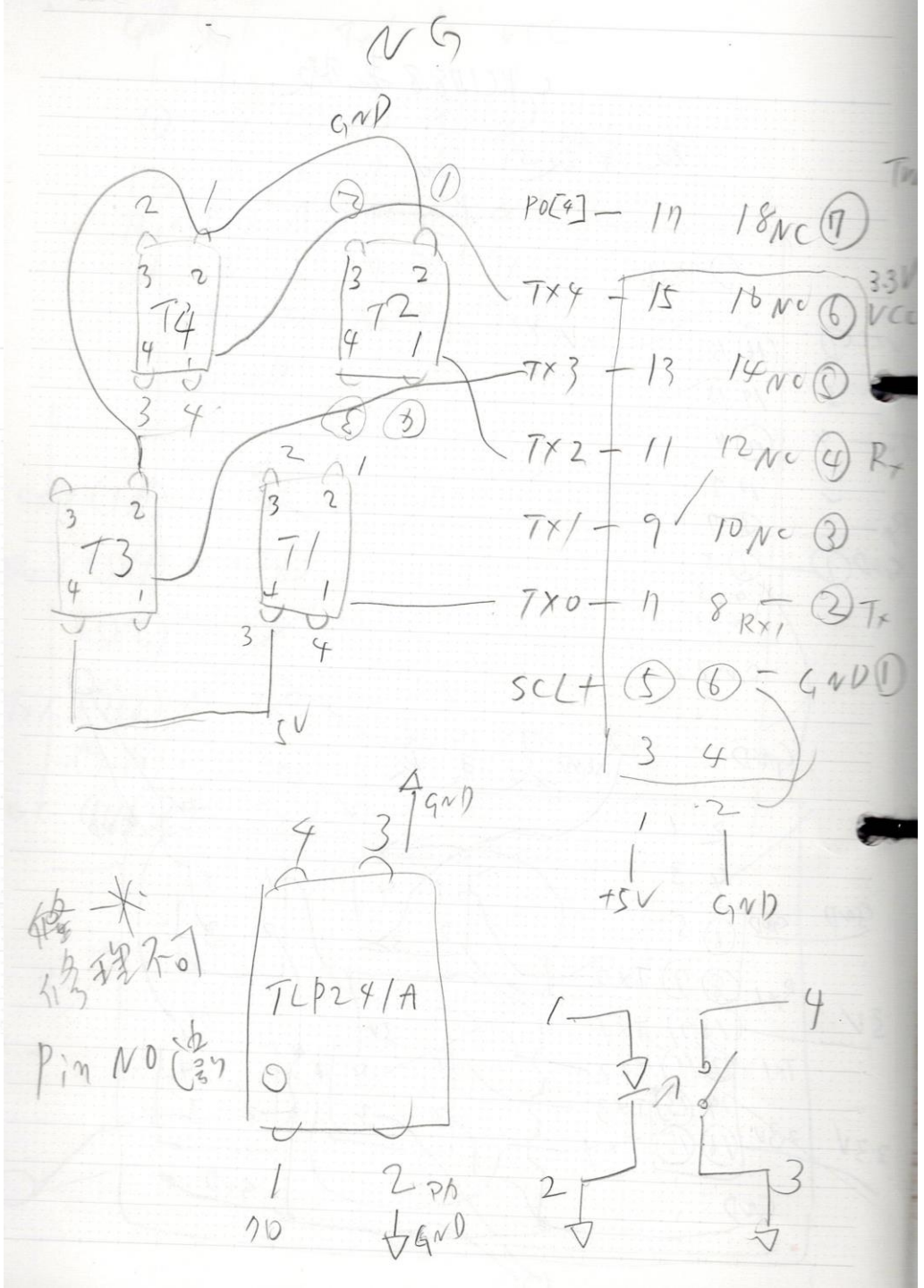



2023
5/4
TLP基板
GND
+5V
LPC基板
+3.3V
TX1
TX3
TX2
TX4
LPC
IC2
HC04
TLP241A ×4

TLP基板 test




TX0
TX1
TX2
TX3
TX4
GPIO


GND
T4
T2
T3
T1
5V
TX4
TX3
TX2
TX1
TX0
SCL+
+5V
GND
TLP241A
Pin NO

ソフト チェック OK

5/9

P0[2] TX0 – GPIO tbr1 tbr1f

IOCON

18 17 16 | 15 14 13 12 | 11 10 9 8 | 7 6 5 4 | 3 2 1 0

P0

1 0 0 1 1

4

1 0 1 1

P0[10] TX2 → GPIO tbr2

tbr2f

P0[0] TX3 → GPIO tbr3 tbr3f

tbr4f

tbr4

P0[22] TX4 → GPIO (NG) → OK

P0[20]

Pn83_5<<1

R変更

CN2(5)

4 0

– 24 | 23 22 21 20 | 19 18 17 16 | 15 –

1 0 1 1

MASK

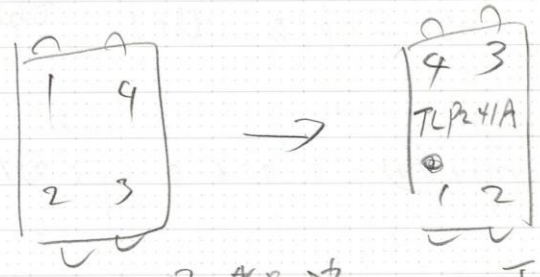
0x B

xFGPIO_FIDpin

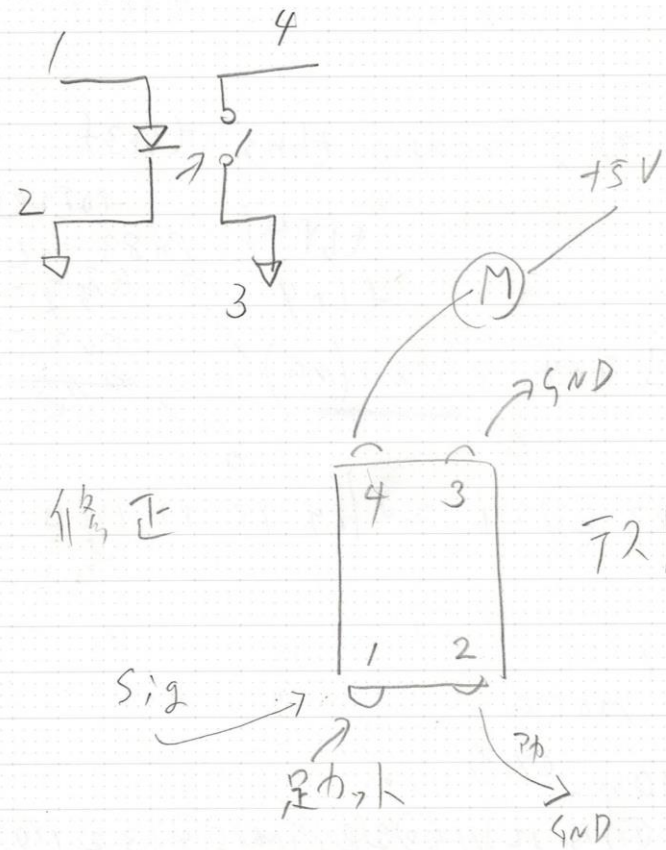
19 18 17 16 | 15 14 13 12 | 11 10 9 8 | 7 6 5 4 | 3 2 1 0

4 0 4 7 F 5

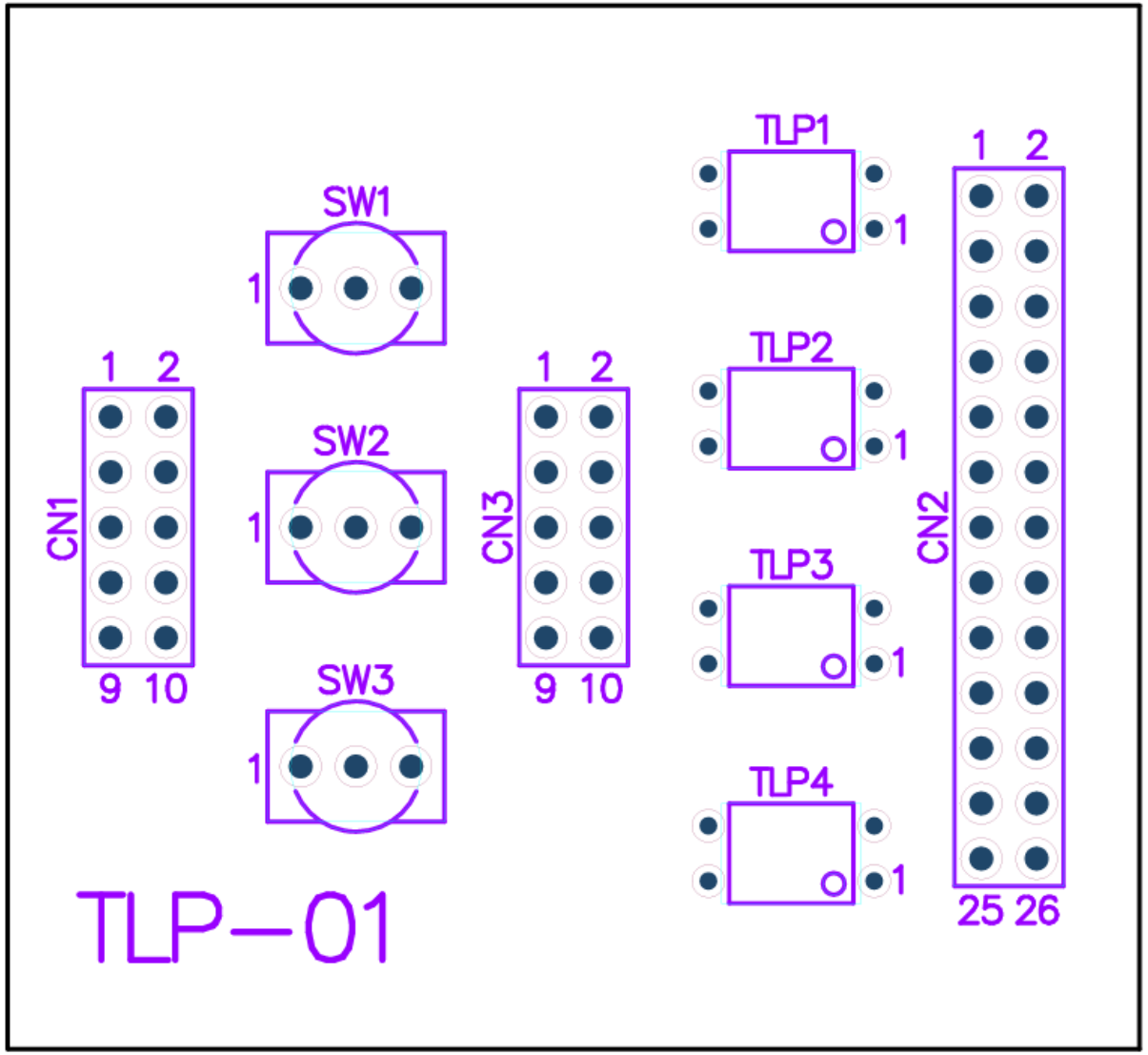
5/10 TCP基板 動作確認NG

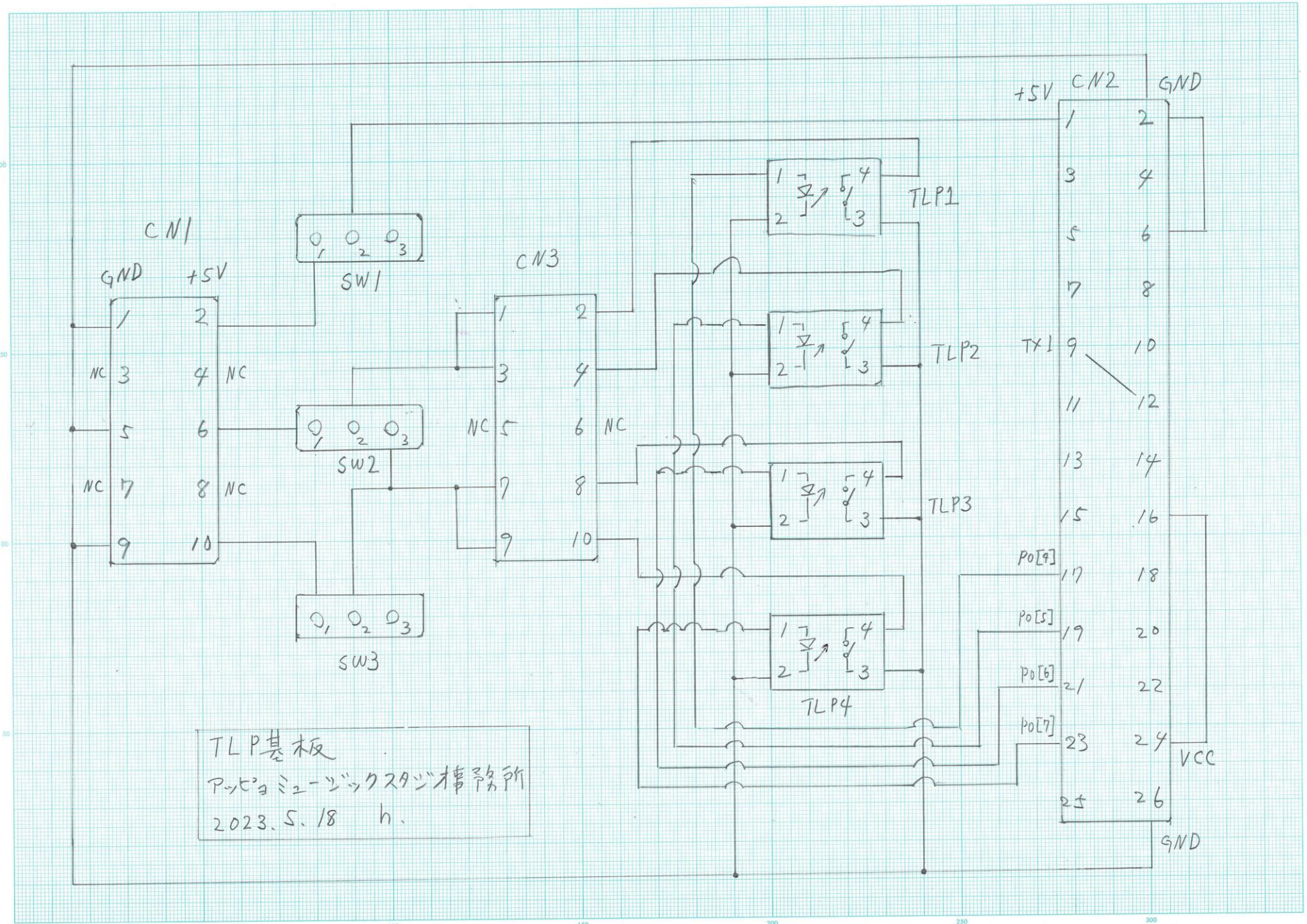


足参考違い 正

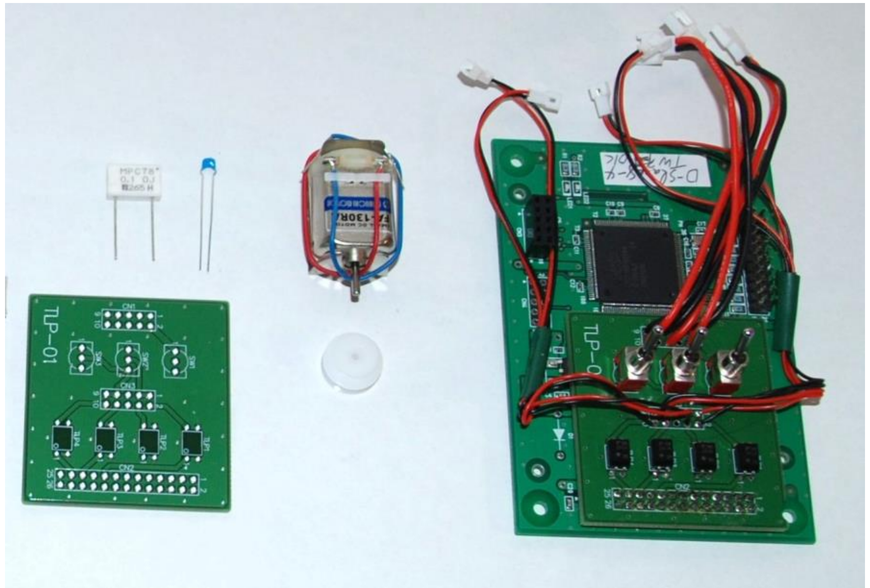


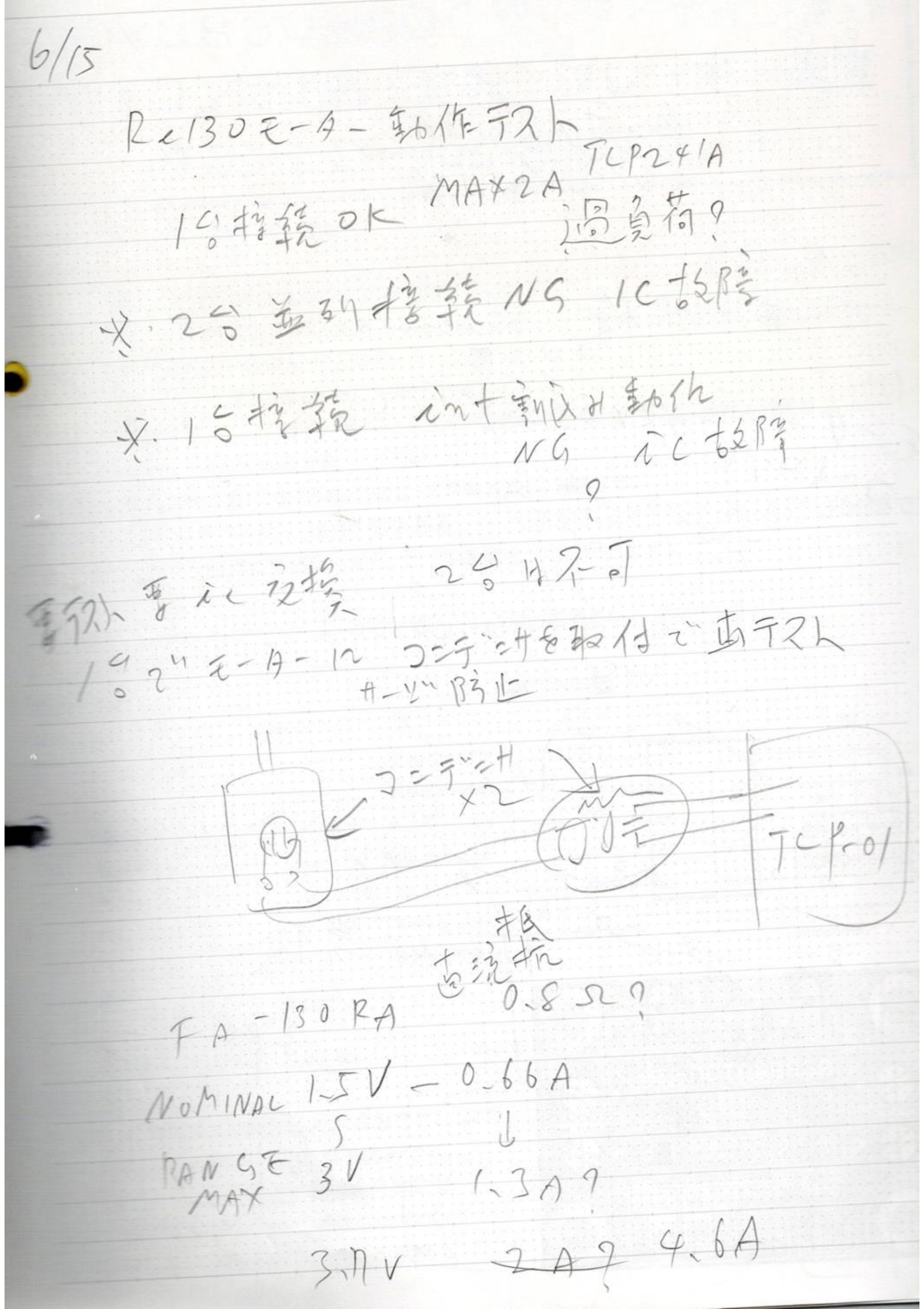
TLP基板製造




CN1
GND
+5V
NC
SW1
SW2
SW3
CN3
TLP1
TLP2
TLP3
TLP4
CN2
TX1
+5V
GND
Po[4]
Po[5]
Po[6]
Po[7]
VCC
TLP基板
アッピョミュージックスタジオ事務所
2023.5.18 h.

TLP基板モーターFA-130RA駆動テスト





6/15

RE130モーター動作テスト

TLP241A
MAX2A

1台接続OK 過負荷?

※2台並列接続NG IC故障

※1台接続 ON+割込み動作 NG IC故障

?

要IC交換 2台は不可

1台でモーターにコンデンサを取付で再テスト
サージ防止

直流抵抗 0.8Ω?

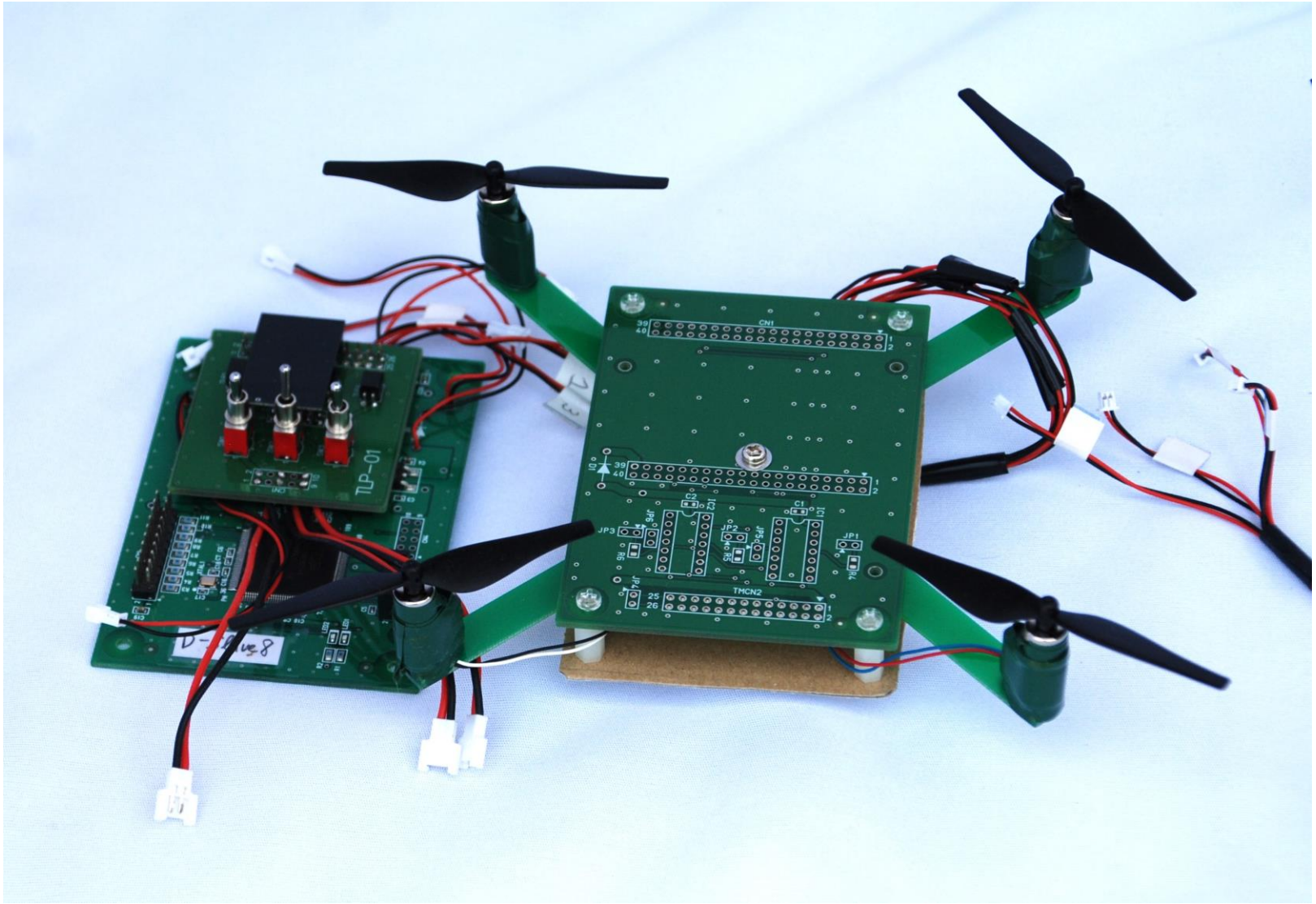
FA-130RA

NOMINAL 1.5V — 0.66A

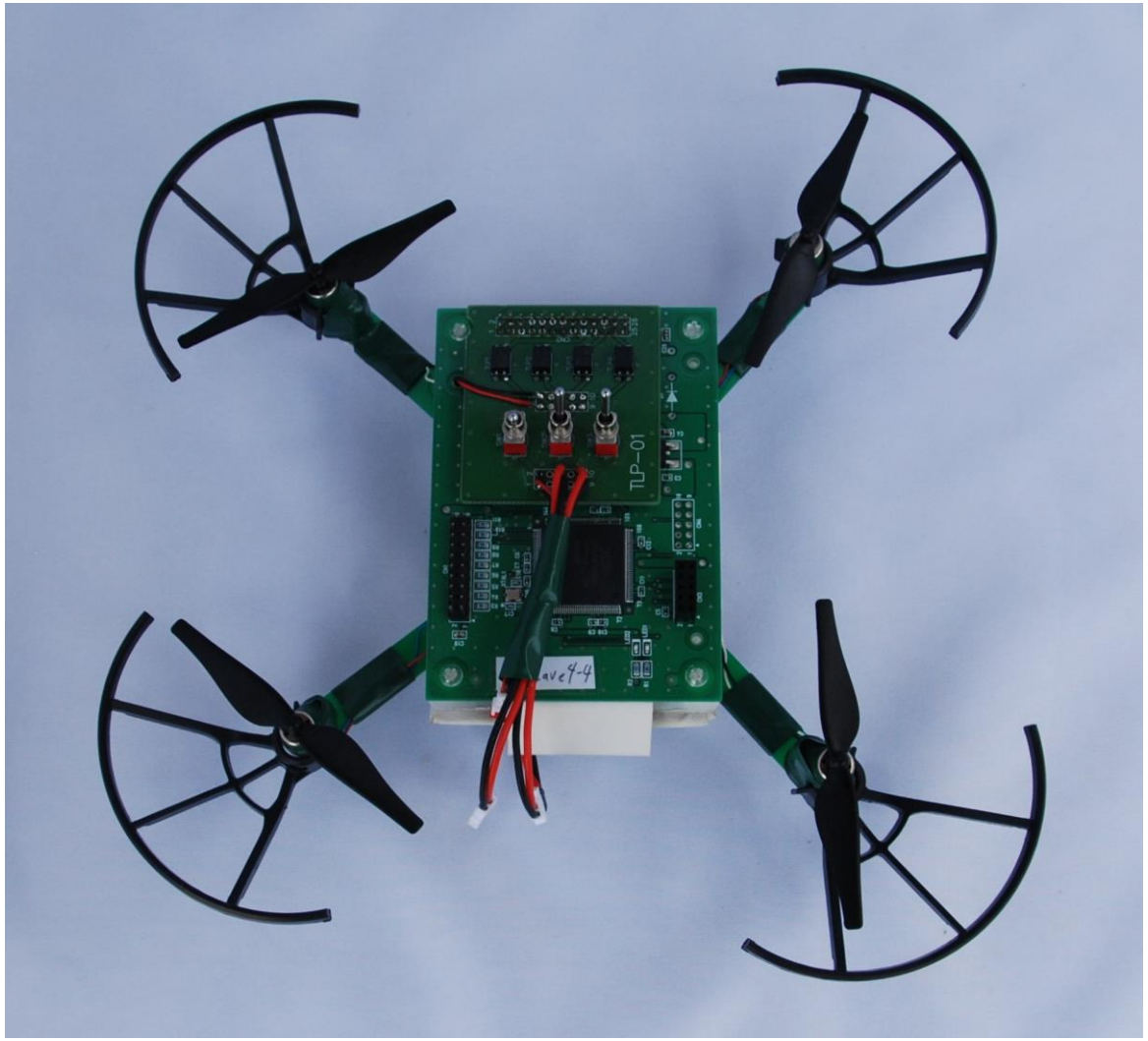
RANGE MAX 3V 1.3A?

3.7V ~~2A?~~ 4.6A

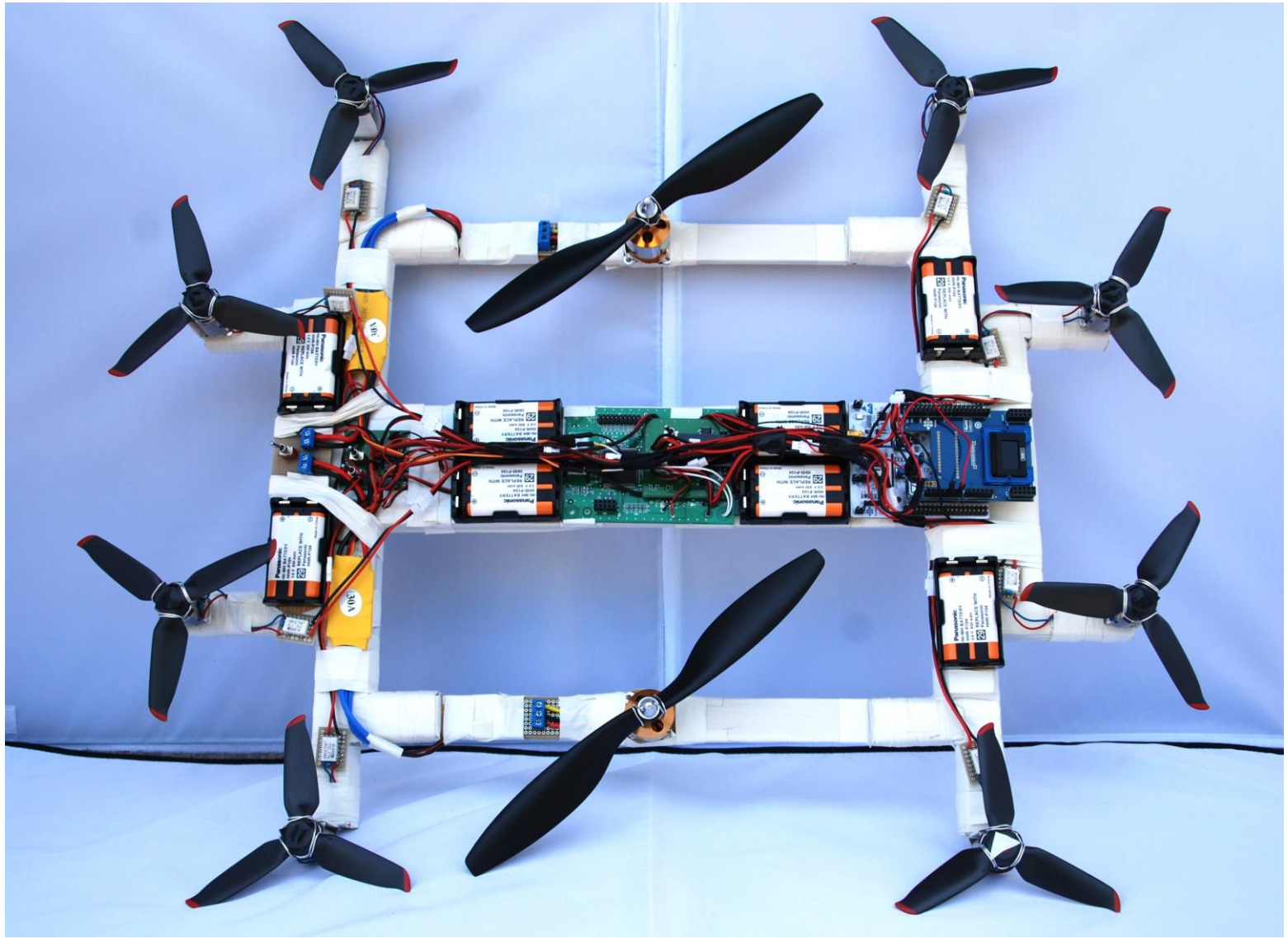
TLP基板の利用例



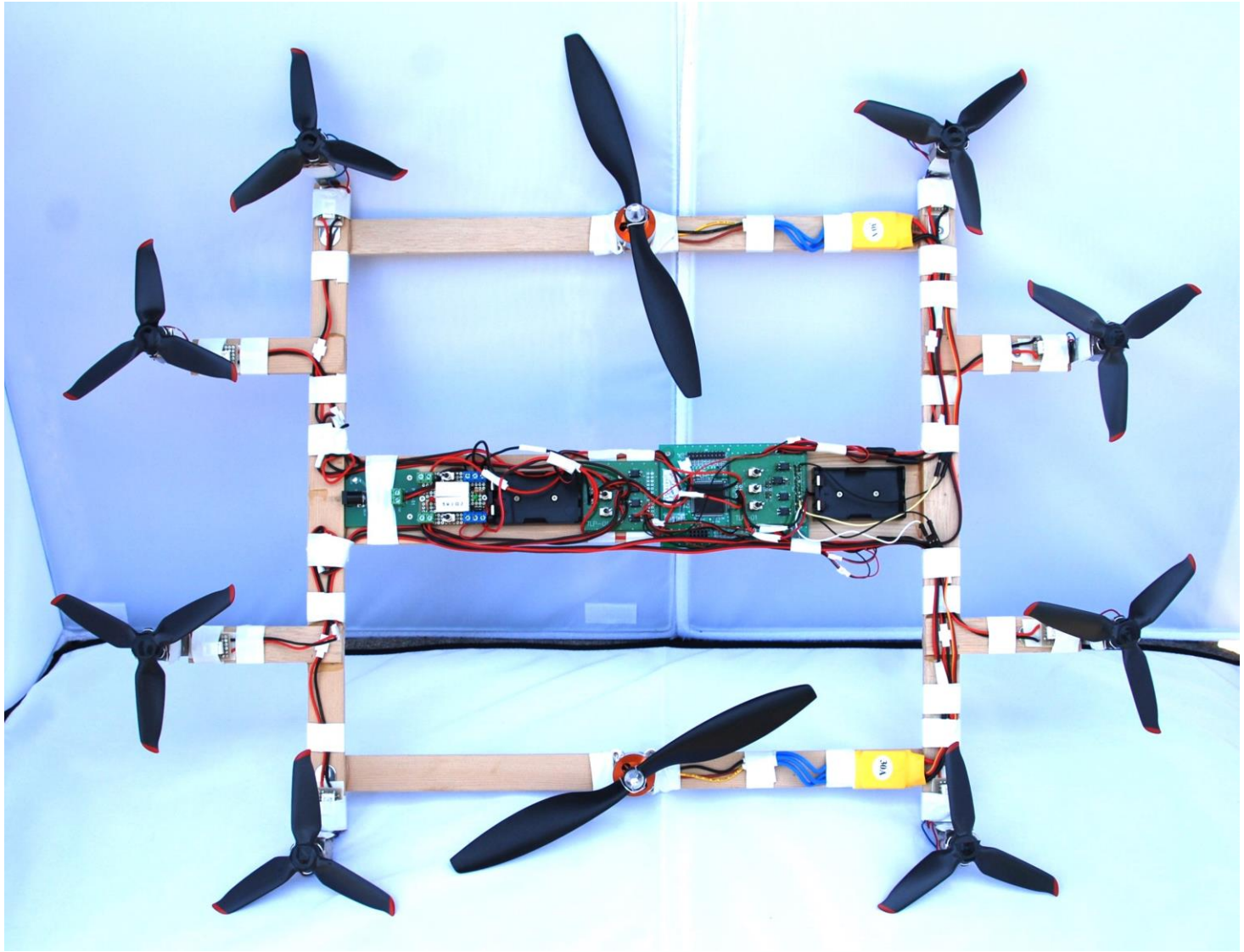
TLP基板の利用例



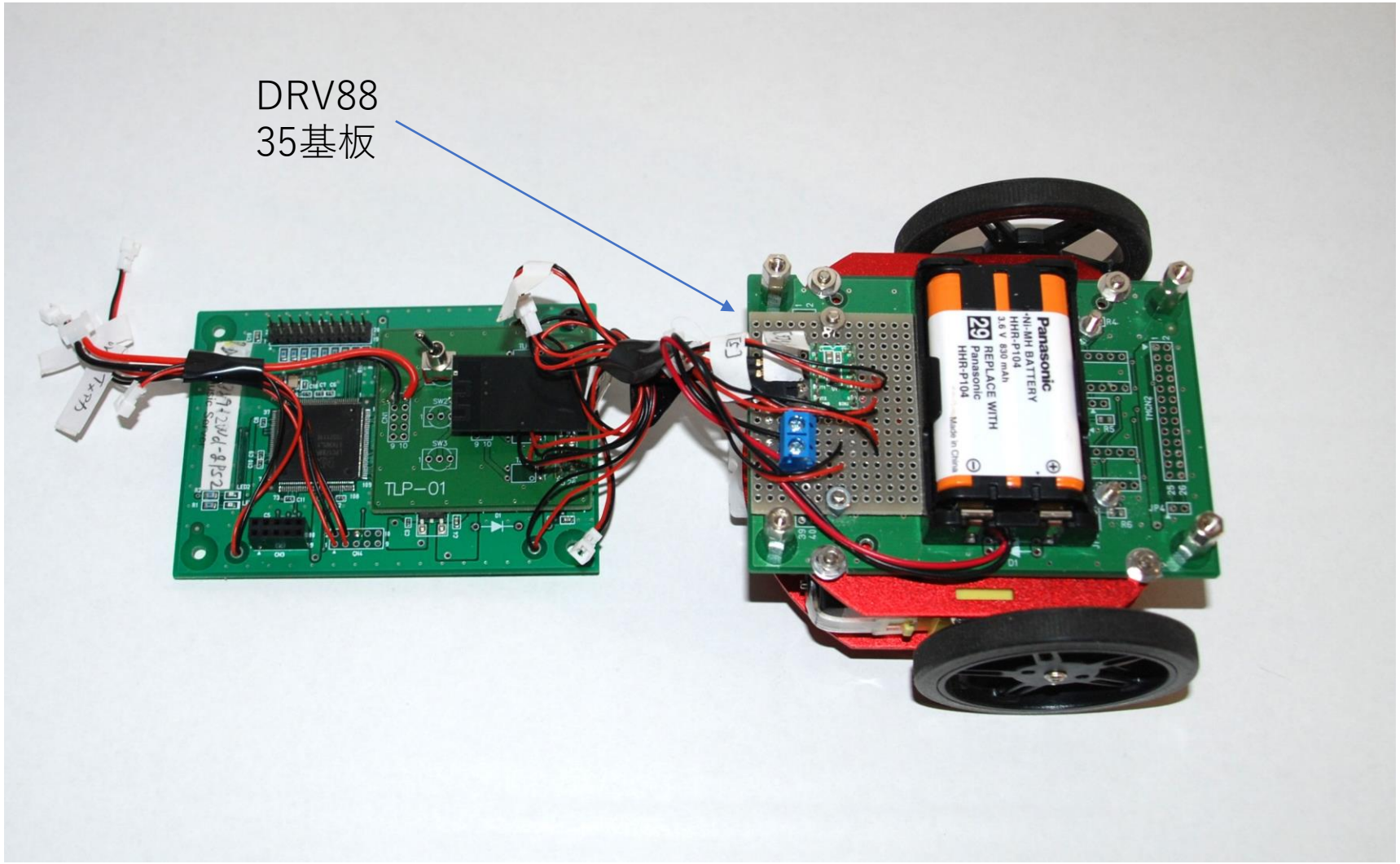
TLP基板の利用例



TLP基板の利用例



Texas DRV8835 DC motor x 2 drive test

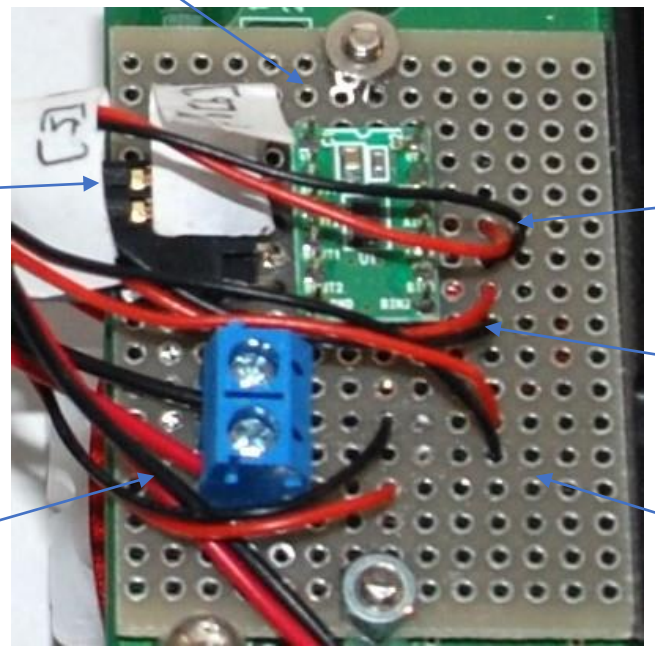


Texas DRV8835 DC motor x 2 drive test

```
/*
*IOCON_P0_4 = 0x10;  // Set P0[4] Pull up enabled -> GPIO DRV8835 APHASE
*IOCON_P0_5 = 0x10;  // Set P0[5] Pull up enabled -> GPIO DRV8835 AENBL
*IOCON_P0_6 = 0x10;  // Set P0[6] Pull up enabled -> GPIO DRV8835 BPHASE
*IOCON_P0_7 = 0x10;  // Set P0[7] Pull up enavled -> GPIO DRV8835 BENBL
*/
```

DRV8835基板

DCモーター x 2
L R
Phase/Enable mode



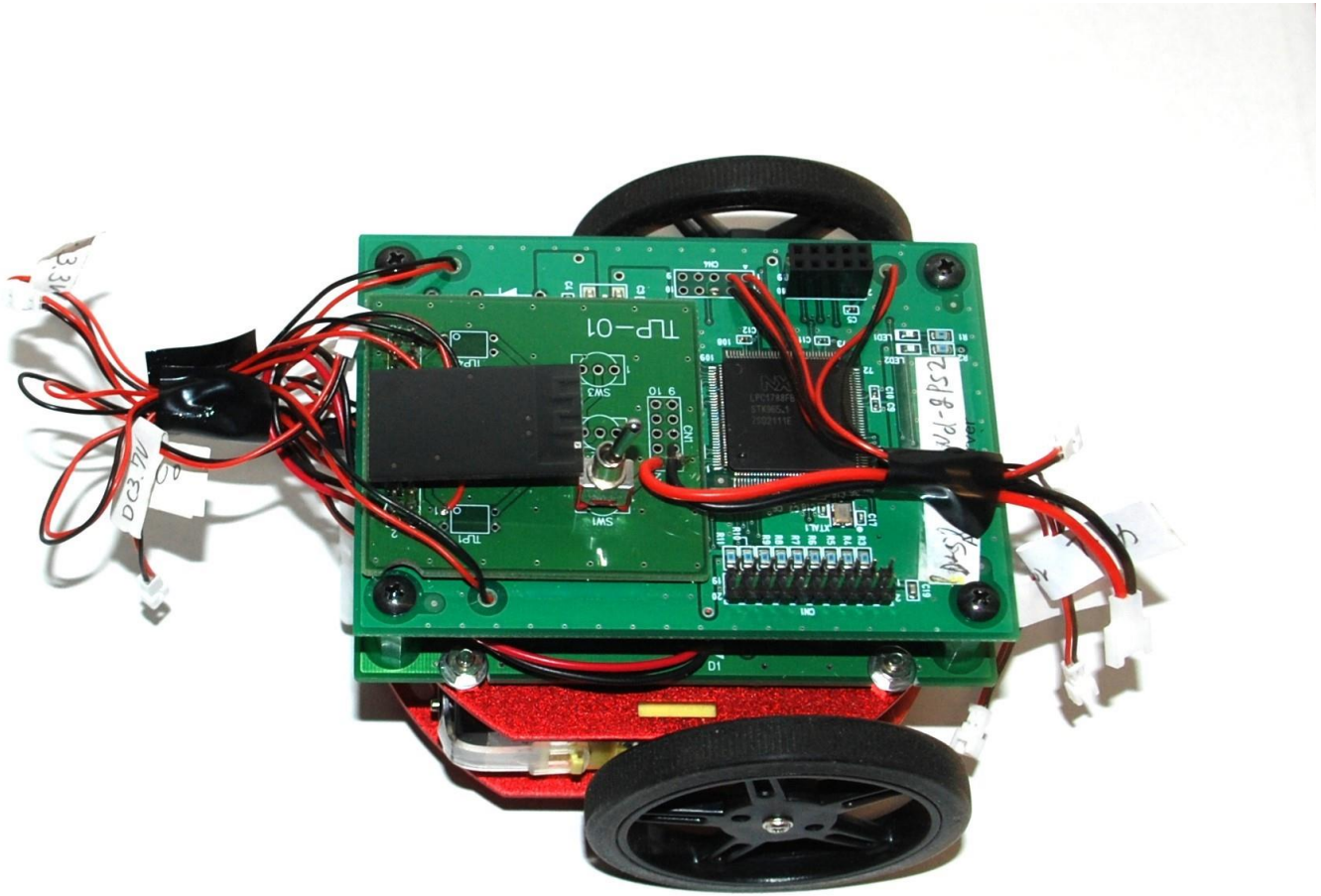
P0[4] P0[5]

P0[6] P0[7]

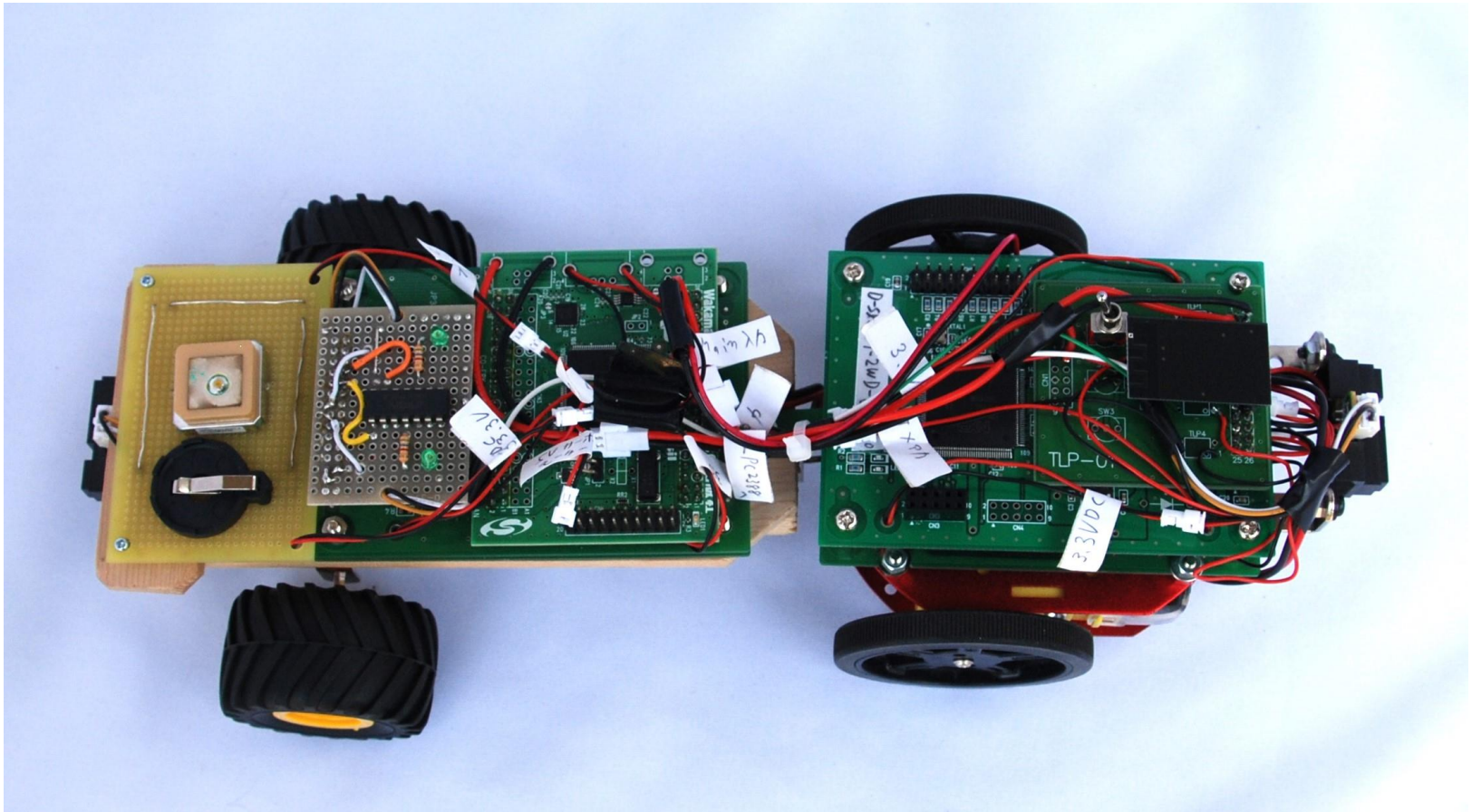
モーター用
DC3.7V

DC 3.3V

Texas DRV8835 DC motor x 2 drive test



Texas DRV8835 DC motor x 2 利用例



Texas DRV8835 DC motor x 2 利用例

まとめ

テスト項目

1. AE-STEP-KIT
2. LPC1788でブラシレスモーターを駆動する。
3. SSRキットを使ってモーターを直接駆動する。
4. NXP AE-PCA9685 Micro Servo test
5. Texas DRV8835 Step motor ST-42-BYH1004-5013 uno test
6. Texas DRV8835 Step motor LPC1788 test
7. TLP基板
8. Texas DRV8835 DC motor x 2 drive test