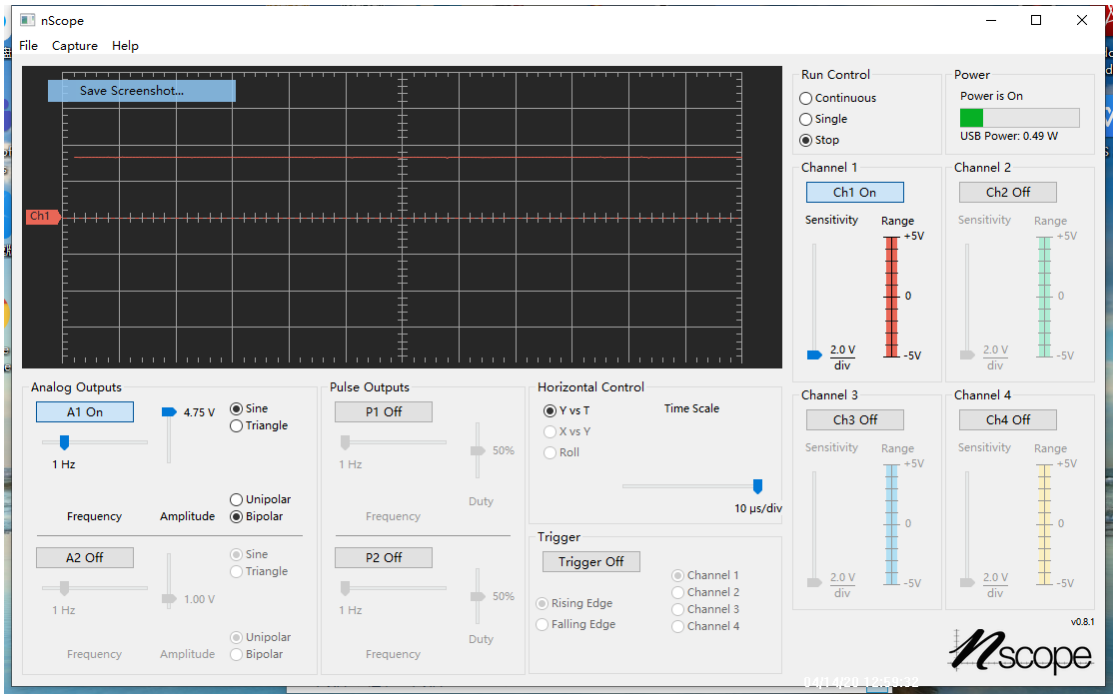
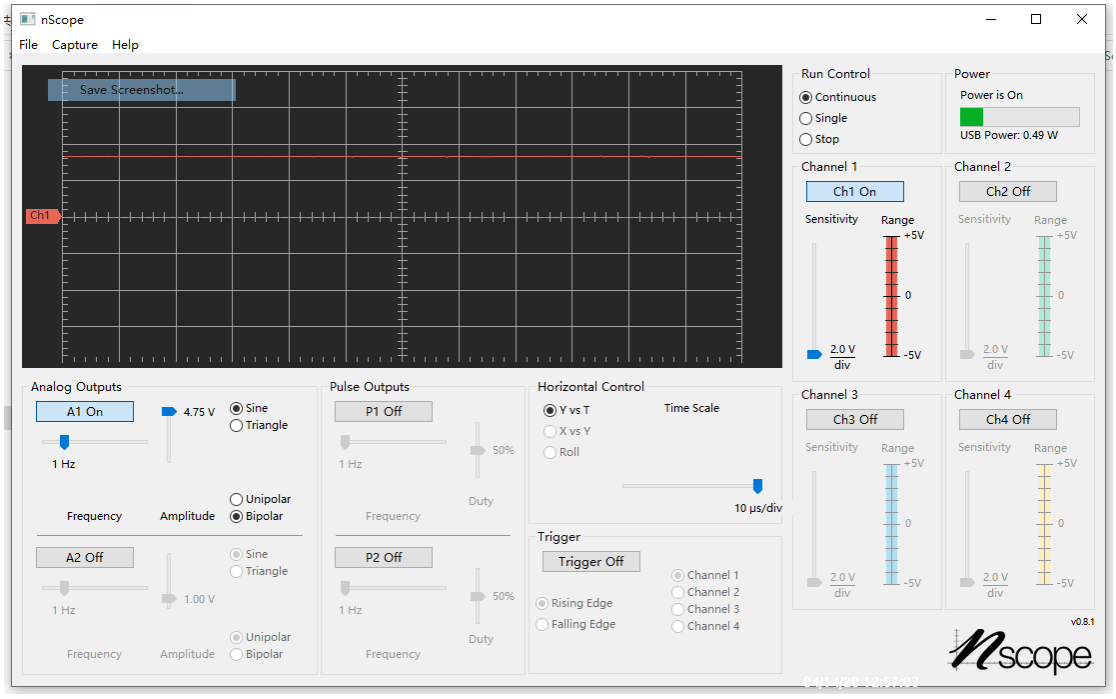
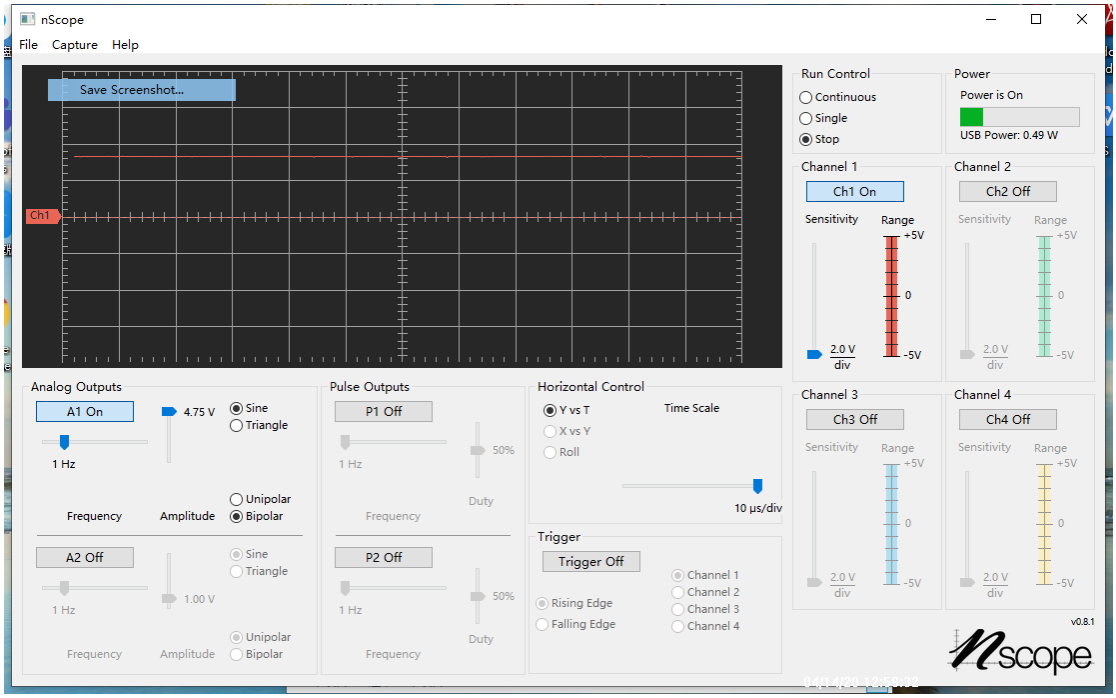
HW1-Dawei Liu



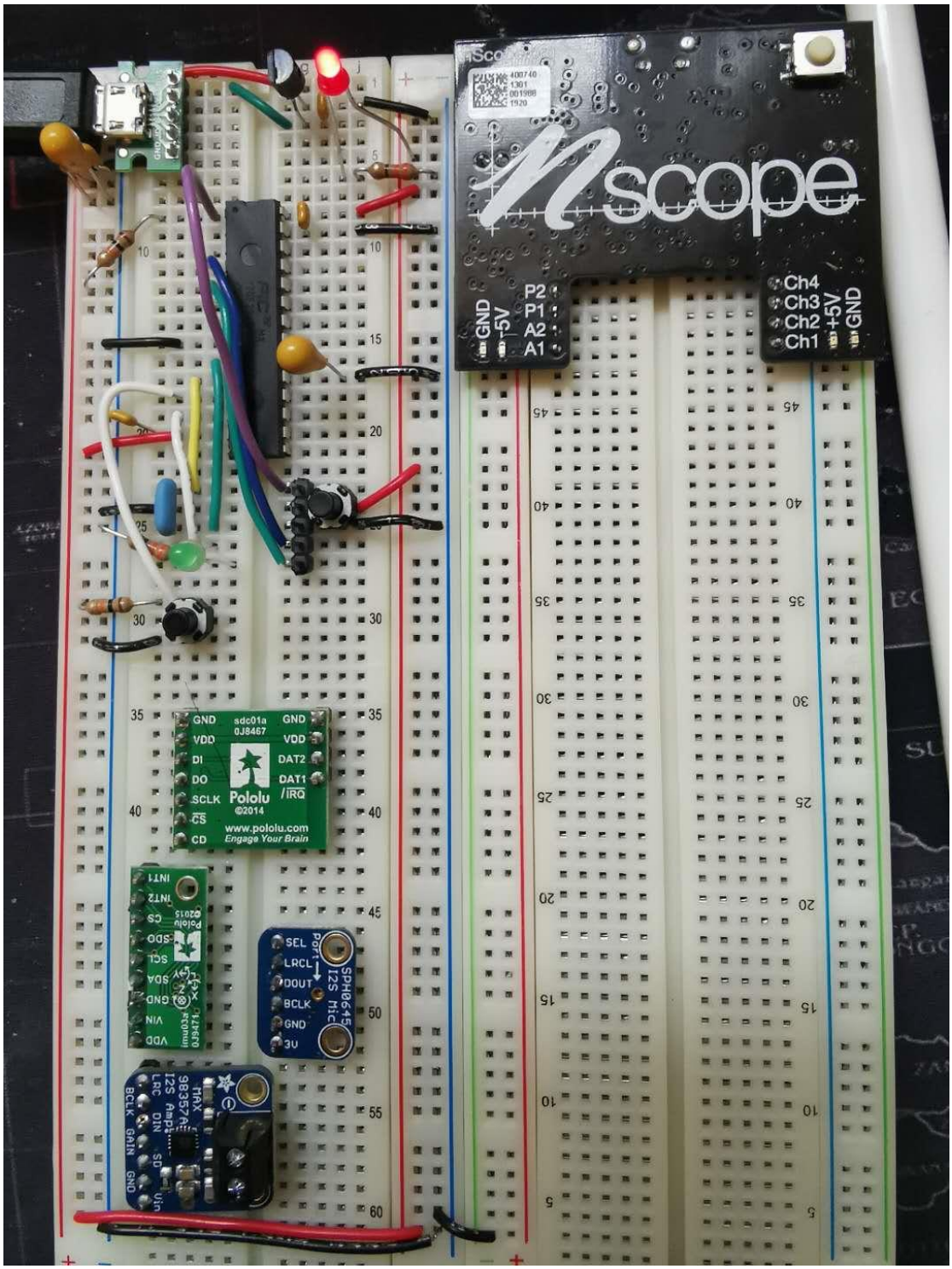
Screenshot without led



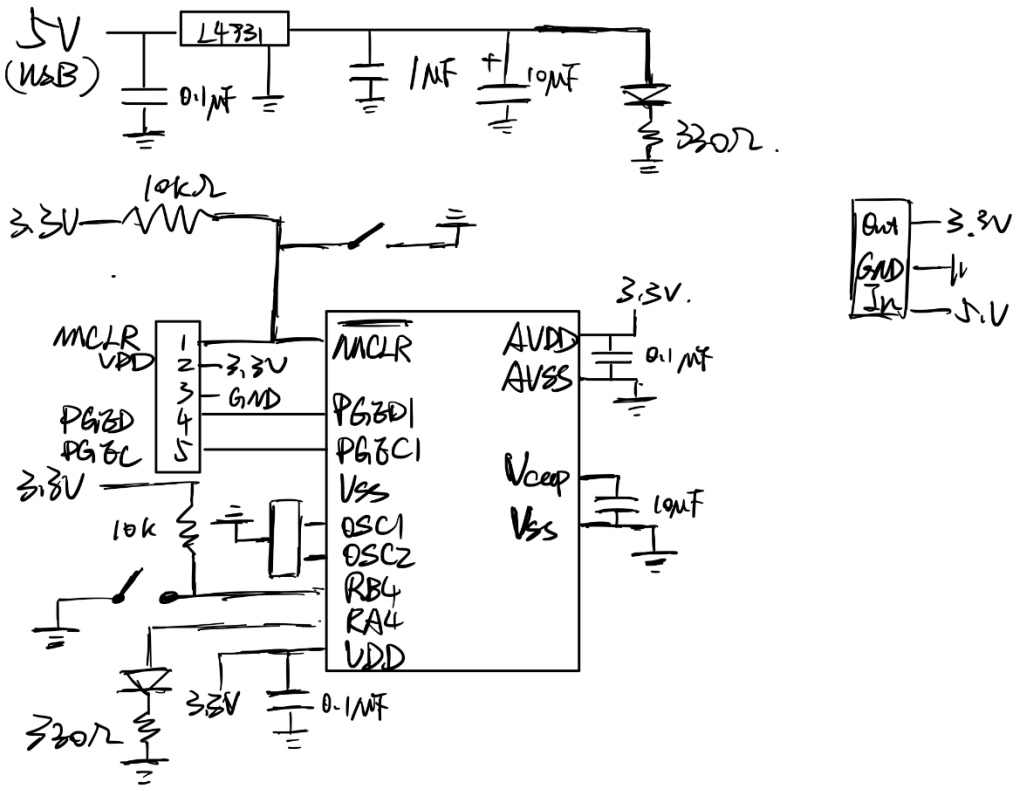
Screenshot with LED



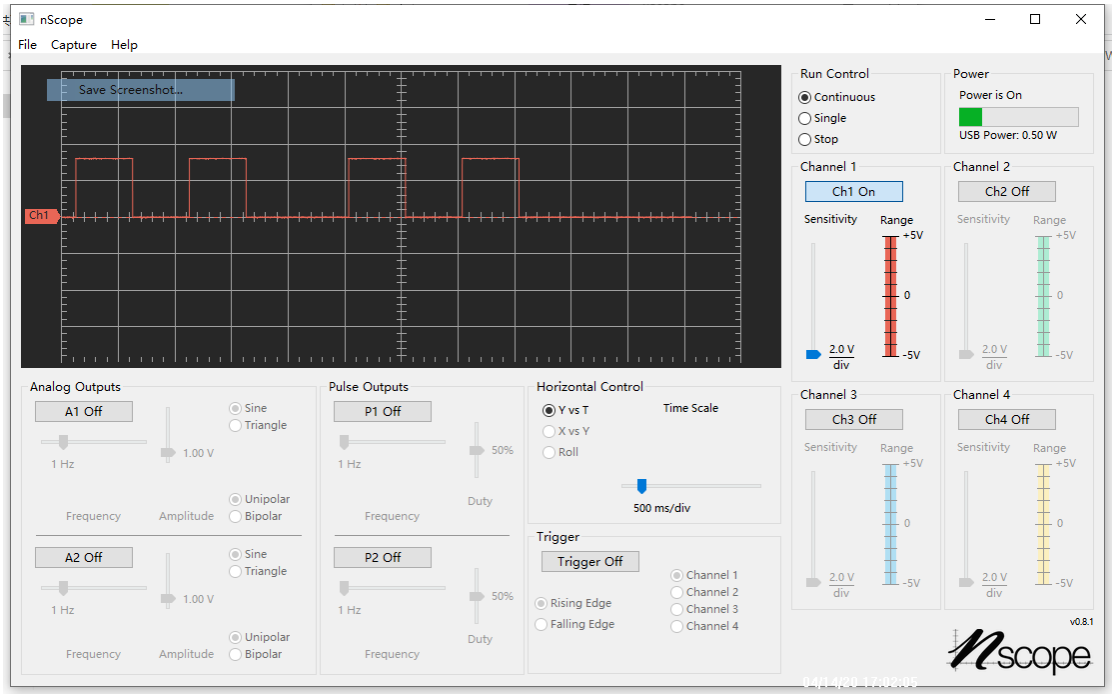
Screenshot with capacitor


nscope
GND
-5V
P2
P1
A2
A1
Ch4
Ch3
Ch2
Ch1
+5V
GND
Pololu
www.pololu.com
Engage Your Brain

Circuit



Circuit diagram



Screenshot for flashing LED

Code

```
#include<xc.h>
// processor
SFR
definitions
                #include<sys/attribs.h>  // __ISR macro

                // DEVCFG0
                #pragma config DEBUG = OFF // disable debugging
                #pragma config JTAGEN = OFF // disable jtag
                #pragma config ICESEL = ICS_PGx1 // use PGED1 and PGEC1
                #pragma config PWP = OFF // disable flash write protect
                #pragma config BWP = OFF // disable boot write protect
                #pragma config CP = OFF // disable code protect

                // DEVCFG1
                #pragma config FNOSC = PRIPLL // use primary oscillator with
                pll
                #pragma config FSOSCEN = OFF // disable secondary oscillator
                #pragma config IESO = OFF // disable switching clocks
                #pragma config POSCMOD = HS // high speed crystal mode
                #pragma config OSCIOFNC = OFF // disable clock output
                #pragma config FPBDIV = DIV_1 // divide sysclk freq by 1 for
                peripheral bus clock
                #pragma config FCKSM = CSDCMD // disable clock switch and FSCM
                #pragma config WDTPS = PS1048576 // use largest wdt
                #pragma config WINDIS = OFF // use non-window mode wdt
                #pragma config FWDTEN = OFF // wdt disabled
                #pragma config FWDTWINSZ = WINSZ_25 // wdt window at 25%

                // DEVCFG2 - get the sysclk clock to 48MHz from the 8MHz
                crystal
                #pragma config FPLLIDIV = DIV_2 // divide input clock to be in
                range 4-5MHz
                #pragma config FPLLMUL = MUL_24 // multiply clock after
                FPLLIDIV
                #pragma config FPLLODIV = DIV_2 // divide clock after FPLLMUL
                to get 48MHz

                // DEVCFG3
                #pragma config USERID = 0 // some 16bit userid, doesn't matter
                what
                #pragma config PMDL1WAY = OFF // allow multiple
                reconfigurations
```

```
#pragma config IOL1WAY = OFF // allow multiple reconfigurations


int main() {

    __builtin_disable_interrupts(); // disable interrupts while
initializing things

    // set the CP0 CONFIG register to indicate that kseg0 is
cacheable (0x3)
    __builtin_mtc0(_CP0_CONFIG, _CP0_CONFIG_SELECT, 0xa4210583);

    // 0 data RAM access wait states
    BMXCONbits.BMXWSDRM = 0x0;

    // enable multi vector interrupts
    INTCONbits.MVEC = 0x1;

    // disable JTAG to get pins back
    DDPCONbits.JTAGEN = 0;

    // do your TRIS and LAT commands here
    TRISAbits.TRISA4 = 0;
    TRISBbits.TRISB4 = 1;



    __builtin_enable_interrupts();

    while (1) {
        // use _CP0_SET_COUNT(0) and _CP0_GET_COUNT() to test
the PIC timing
        // remember the core timer runs at half the sysclk
        if (PORTBbits.RB4 == 0){
        _CP0_SET_COUNT(0);
        while(_CP0_GET_COUNT() < 24000000/2){LATAbits.LATA4 =
1;}
        _CP0_SET_COUNT(0);
        while(_CP0_GET_COUNT() < 24000000/2){LATAbits.LATA4 =
0;}
        _CP0_SET_COUNT(0);
        while(_CP0_GET_COUNT() < 24000000/2){LATAbits.LATA4 =
1;}
        _CP0_SET_COUNT(0);
```

```
        while(_CP0_GET_COUNT() < 24000000/2){LATAbits.LATA4 =
0;}
        }
    }
}
```