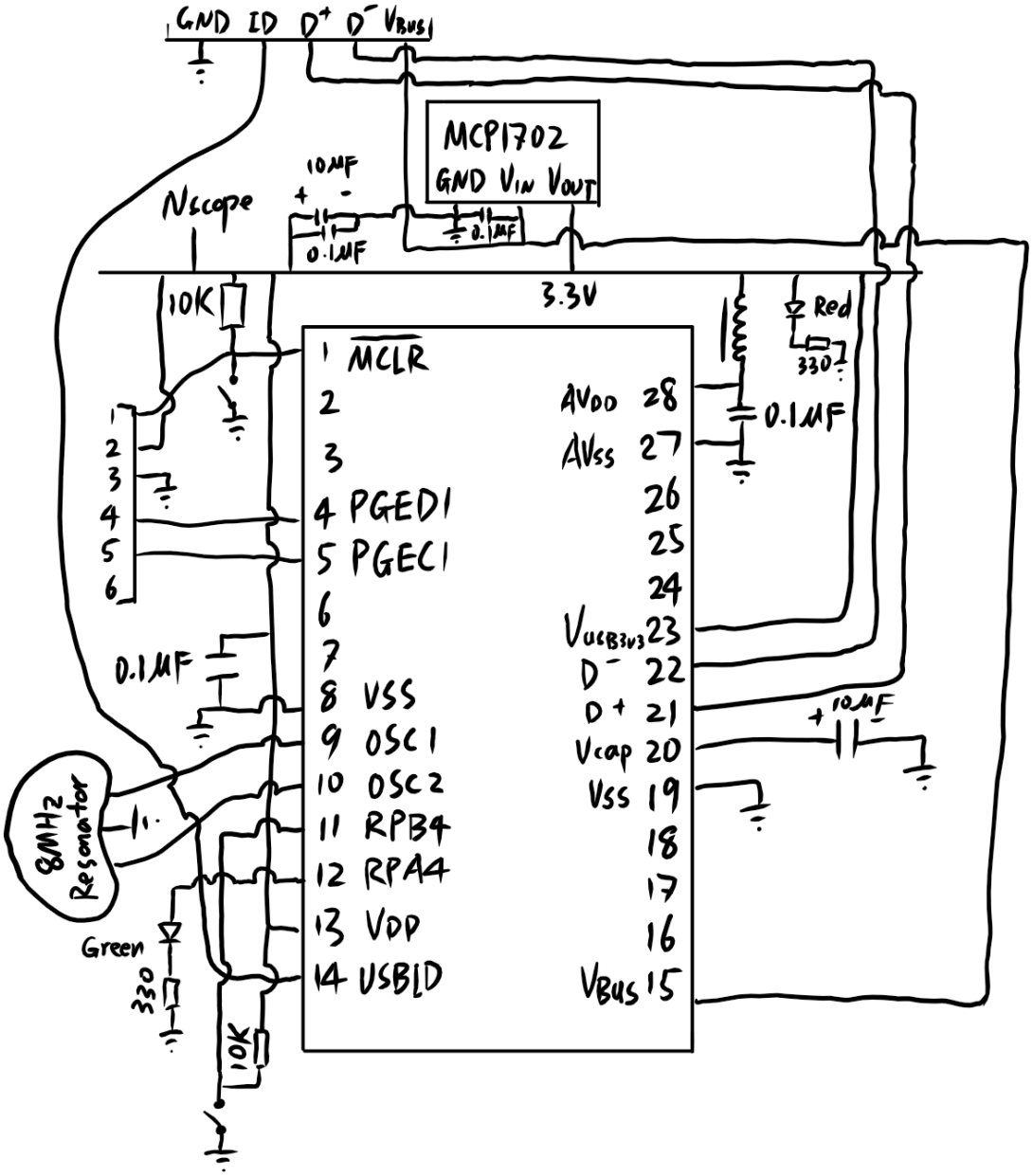
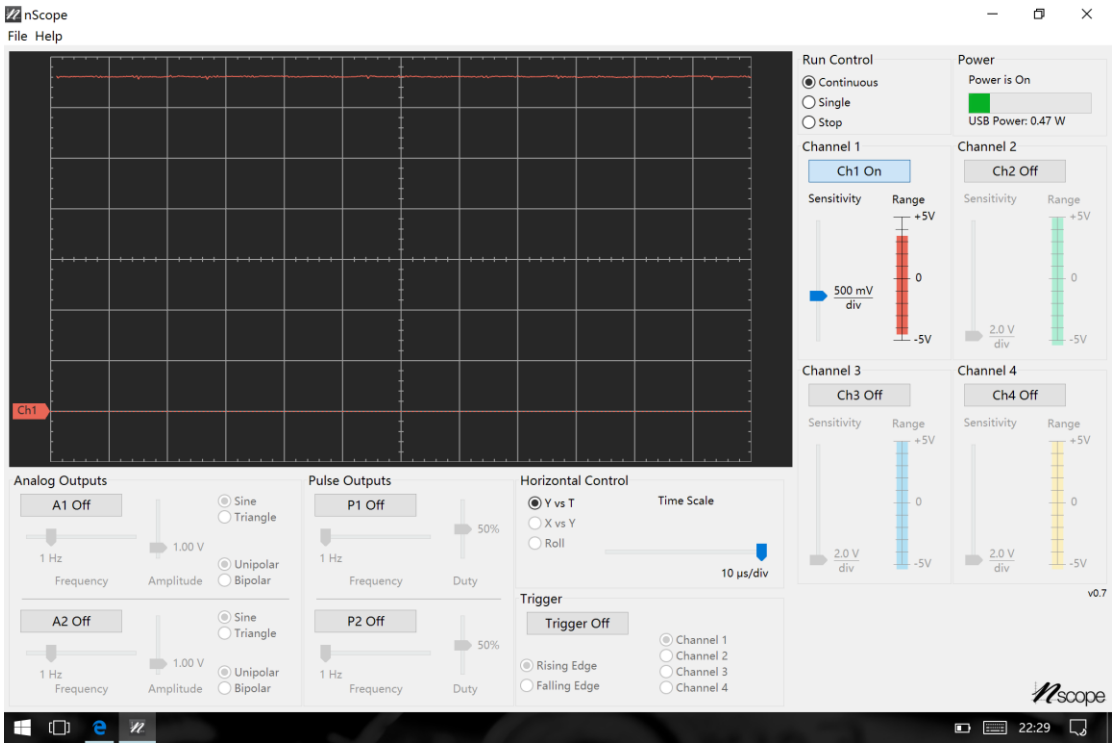
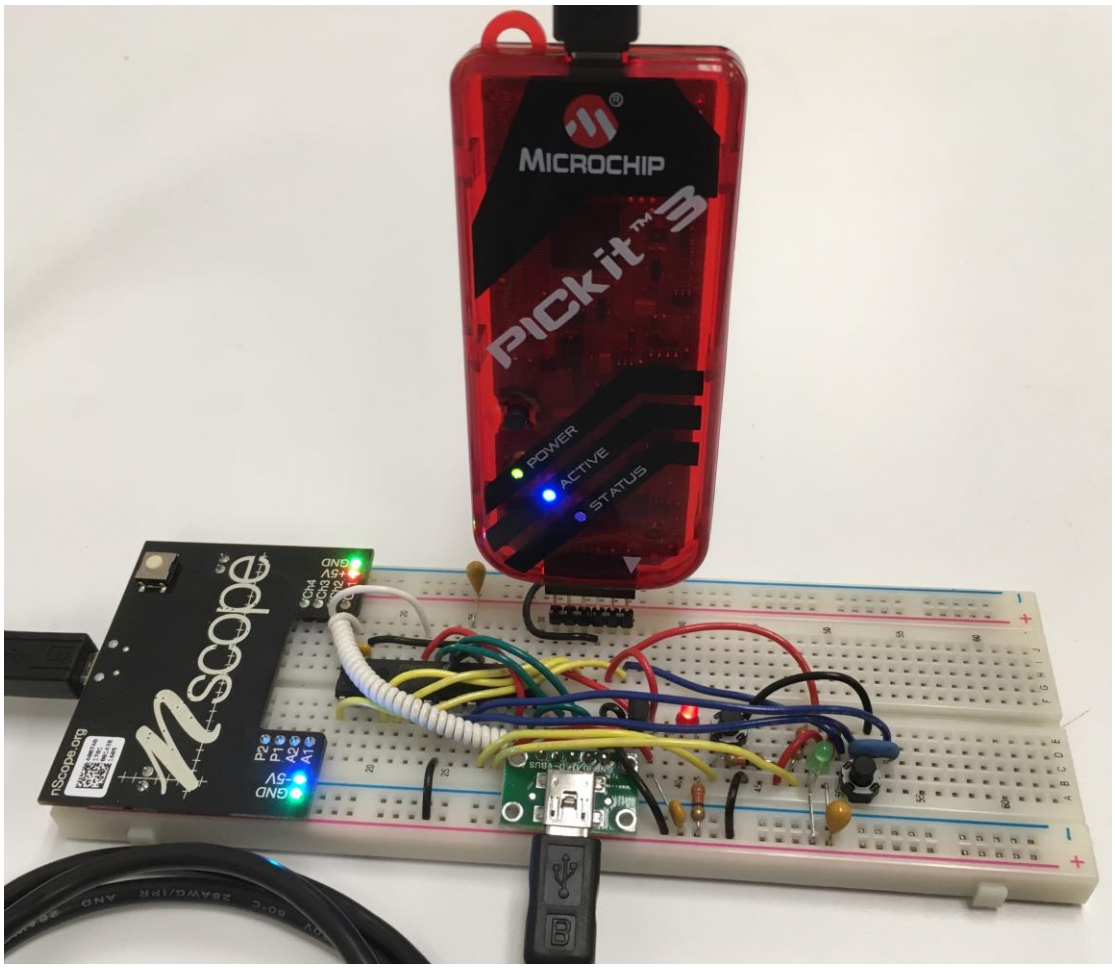

GND ID D+ D- VBUS
MCP1702
GND VIN VOUT
10MF
0.1MF
0.1MF
Nscope
3.3V
10K
Red
330
0.1MF
1 MCLR
2
3
4 PGED1
5 PGEC1
6
7
8 VSS
9 OSC1
10 OSC2
11 RPB4
12 RPA4
13 VDD
14 USBLD
AVDD 28
AVSS 27
26
25
24
VUSB3V3 23
D- 22
D+ 21
VCAP 20
VSS 19
18
17
16
VBUS 15
+ 10MF
0.1MF
8MHz Resonator
Green
330
10K


nScope
File Help
Ch1
Run Control
Continuous
Single
Stop
Power
Power is On
USB Power: 0.47 W
Channel 1
Ch1 On
Sensitivity
Range
+5V
0
-5V
500 mV
div
Channel 2
Ch2 Off
2.0 V
div
Channel 3
Ch3 Off
Channel 4
Ch4 Off
v0.7
Analog Outputs
A1 Off
Sine
Triangle
1 Hz
Frequency
1.00 V
Amplitude
Unipolar
Bipolar
A2 Off
Pulse Outputs
P1 Off
50%
Duty
P2 Off
Horizontal Control
Y vs T
X vs Y
Roll
Time Scale
10 µs/div
Trigger
Trigger Off
Rising Edge
Falling Edge
Channel 1
Channel 2
Channel 3
Channel 4
nscope
22:29


MICROCHIP
PICkit 3
POWER
ACTIVE
STATUS
nScope
nScope.org

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

// DEVCFG0
#pragma config DEBUG     = 0b10     // no debugging
#pragma config JTAGEN    = 0        // no jtag
#pragma config ICESEL    = 0b11     // use PGED1 and PGEC1
#pragma config PWP       = 0x1FF    // no write protect
#pragma config BWP       = 1        // no boot write protect
#pragma config CP        = 1        // no code protect

// DEVCFG1
#pragma config FNOSC     = 0b011    // use primary oscillator with pll
#pragma config FSOSCEN   = 0        // turn off secondary oscillator
#pragma config IESO      = 0b0      // no switching clocks
#pragma config POSCMOD   = 0b10     // high speed crystal mode
#pragma config OSCIOFNC  = 1        // free up secondary osc pins
#pragma config FPBDIV    = 0          // divide CPU freq by 1 for
peripheral bus clock
#pragma config FCKSM     = 0b10     // do not enable clock switch
#pragma config WDTPS     = 0x14     // slowest wdt
#pragma config WINDIS    = 1        // no wdt window
#pragma config FWDTEN    = 0        // wdt off by default
#pragma config FWDTWINSZ = 0b11     // wdt window at 25%

// DEVCFG2 - get the CPU clock to 48MHz
#pragma config FPLLIDIV  = 1         // divide input clock to be in
range 4-5MHz
#pragma config FPLLMUL   = 0b111    // multiply clock after FPLLIDIV
#pragma config FPLLODIV  = 1         // divide clock after FPLLMUL to
get 48MHz
#pragma config UPLLIDIV  = 1          // divider for the 8MHz input
clock, then multiply by 12 to get 48MHz for USB
#pragma config UPLLEN    = 0        // USB clock on

// DEVCFG3
#pragma config USERID    = 0          // some 16bit userid, doesn't
matter what
#pragma config PMDL1WAY  = 0        // allow multiple reconfigurations
#pragma config IOL1WAY   = 0        // allow multiple reconfigurations
#pragma config FUSBIDIO  = 1          // USB pins controlled by USB
module
#pragma config FVBUSONIO = 1         // USB BUSON controlled by USB
module
```

```
int main() {

    __builtin_disable_interrupts();

    // 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;
    LATAbits.LATA4       = 1;
    __builtin_enable_interrupts();

    while (1) {
       _CP0_SET_COUNT(0);
       while (_CP0_GET_COUNT()<12000) {;}
       LATAbits.LATA4 = !LATAbits.LATA4;
       while (PORTBbits.RB4 == 1){;}
    }
}
```