

```

/*****
Tidal Engineering © 1998
www.gti.net/tidaleng
1-800-877-0510
RCALECRY.C
Infrared (IR) Remote Controlled Oscilloscope
Lecroy 9310
Scope GPIB address should be set to 6
*****/
#include MT488A.LIB

main() {
    int i;
    int count;
    char stbuf[TBUFSIZE]; // transmit buffer
    char srbuf[RBUFSIZE]; // receive buffer
    char buf[RBUFSIZE+1]; // dummy buffer for receiving a complete command
    char s, t,data,key[1];
    char tbuf[ieee_out_len], rbuf[ieee_in_len];
    int osc;
    unsigned int chan;
    int ii,n,j,z,v,PAD_;
    char temp[25];

    /* The following character strings set the Lecroy delay time */
    char *delay[]={ "TRDL 0\0xa",
                    "TRDL 20\0xa",
                    "TRDL 40\0xa",
                    "TRDL 60\0xa",
                    "TRDL 80\0xa",
                    "TRDL 100\0xa"};

    /* The following character strings set the Lecroy's channel 1 Volts/Division */
    char *ch1[]={ "C1:VDIV 50mV\0xa",
                 "C1:VDIV 100mV\0xa",
                 "C1:VDIV 200mV\0xa",
                 "C1:VDIV 500mV\0xa",
                 "C1:VDIV 1V\0xa",
                 "C1:VDIV 2V\0xa",
                 "C1:VDIV 5V\0xa"};

    /* The following character strings set the Lecroy's channel 2 Volts/Division */
    char *ch2[]={ "C2:VDIV 50mV\0xa",
                 "C2:VDIV 100mV\0xa",
                 "C2:VDIV 200mV\0xa",
                 "C2:VDIV 500mV\0xa",
                 "C2:VDIV 1V\0xa",
                 "C2:VDIV 2V\0xa",
                 "C2:VDIV 5V\0xa"};

    /* The following character strings set the Lecroy's channel 1 offset voltage */
    char *choff1[]={ "C1:OFST -5V\0xa",
                    "C1:OFST -2V\0xa",
                    "C1:OFST -1V\0xa",
                    "C1:OFST 0V\0xa",
                    "C1:OFST 1V\0xa",
                    "C1:OFST 2V\0xa",
                    "C1:OFST 5V\0xa"};

    /* The following character strings set the Lecroy's channel 2 offset voltage */
    char *choff2[]={ "C2:OFST -5V\0xa",
                    "C2:OFST -2V\0xa",
                    "C2:OFST -1V\0xa",
                    "C2:OFST 0V\0xa",
                    "C2:OFST 1V\0xa",
                    "C2:OFST 2V\0xa",
                    "C2:OFST 5V\0xa"};

    /* The following character strings set the Lecroy's sweep */
    char *time[] =

```

```

        {"TDIV .01ms\0xa",
        "TDIV .02ms\0xa",
        "TDIV .05ms\0xa",
        "TDIV .1ms\0xa",
        "TDIV .2ms\0xa",
        "TDIV .5ms\0xa",
        "TDIV 1ms\0xa",
        "TDIV 2ms\0xa",
        "TDIV 5ms\0xa",
        "TDIV 10ms\0xa",
        "TDIV 20ms\0xa",
        "TDIV 50ms\0xa",
        "TDIV 100ms\0xa",
        "TDIV 200ms\0xa",
        "TDIV 500ms\0xa",
        "TDIV 1s\0xa",
        "TDIV 2s\0xa"};

int chli,ch2i,choffli,choff2i,timei,donel,delayi;

VdInit();          // initialize the Dynamic C virtual driver
hitwd();
init_io();

while (1) {
    donel=false;
    ui=klcd;        //dt=dumb terminal, dc=dyanamic c stdio,klcd=keyboard and LCD
    Dinit_z0(srbuf,stbuf,RBUFSIZE,TBUFSIZE, MODE, IBAUD, NO_MODEM, ECHO );
    add488=17;
    /***** Initialize the National Instruments GPIB ASIC *****/
    if (init_488()==-1) {
        sprintf(buf,"\fCould not initialize Natational Instruments IEEE 488 ASIC");
        //Dwrite_z0 (buf,strlen(buf));
        tprintf (buf);
    }
    /***** Set the scope to GPIB address 6 *****/
    PAD_=6;
    /***** Initialize the driver for the scope *****/
    if ((osc = ibdev(0, PAD_, 0, T10s, 1, 0)) < 0) {
        sprintf(buf,"\fCould not open OSC_01 device");
        tprintf (buf);
    }
    /***** Put the scope in Remote REN *****/
    ibren(osc); //
    chli=5;ch2i=5;choffli=5;choff2i=5;delayi=3;timei=0;
    ibwrt(osc, chl[chli], strlen(chl[chli]));
    if (ibsta & ERR) tprintf("    Could not write to device\n\r");
    ibwrt(osc, ch2[ch2i], strlen(ch2[ch2i]));
    if (ibsta & ERR) tprintf("    Could not write to device\n\r");
    ibwrt(osc, time[timei], strlen(time[timei]));
    if (ibsta & ERR) tprintf("    Could not write to device\n\r");
    Dreset_z0rbuf();
    while(1) {
        /***** Watch for a key press on the IR Remote *****/
        n=in_key(key);
        if (n==1){
            switch(*key) {
                case 'a': // Step up through Channel 1, V/Div table
                    if(++chli<7) ibwrt(osc, chl[chli], strlen(chl[chli]));
                    else chli=6;
                    break;
                case 'd': // Step down through Channel 1, V/Div table
                    if(--chli>=0) ibwrt(osc, chl[chli], strlen(chl[chli]));
                    else chli=0;
                    break;
                case 'b': // Step up through Channel 2, V/Div table
                    if(++ch2i<7) ibwrt(osc, ch2[ch2i], strlen(ch2[ch2i]));
                    else ch2i=6;
                    break;
            }
        }
    }
}

```

```

case 'e': // Step down through Channel 2, V/Div table
    if(--ch2i>=0) ibwrt(osc, ch2[ch2i], strlen(ch2[ch2i]));
    else ch2i=0;
    break;
case 'c': // Step up through sweep table
    if(++timei!=17) ibwrt(osc, time[timei], strlen(time[timei]));
    else timei=16;
    break;
case 'f': // Step down through sweep table
    if(--timei>=0) ibwrt(osc, time[timei], strlen(time[timei]));
    else timei=0;
    break;
case 'g': // Step up through Channel 1, offset table
    if(++choff1i<7) ibwrt(osc, choff1[choff1i], strlen(ch1[choff1i]));
    else choff1i=6;
    break;
case 'j': // Step down through Channel 1, offset table
    if(--choff1i>=0) ibwrt(osc, choff1[choff1i], strlen(ch1[choff1i]));
    else choff1i=0;
    break;
case 'h': // Step up through Channel 2, V offset table
    if(++choff2i<7) ibwrt(osc, choff2[choff2i], strlen(choff2[choff2i]));
    else choff2i=6;
    break;
case 'k': // Step down through Channel 2, V offset table
    if(--choff2i>=0) ibwrt(osc, choff2[choff2i], strlen(ch2[choff2i]));
    else choff2i=0;
    break;
case 'i': // Step up through delay table
    if(++delayi<6) ibwrt(osc, delay[delayi], strlen(delay[delayi]));
    else delayi=5;
    break;
case 'l': // Step down through delay table
    if(--delayi>=0) ibwrt(osc, delay[delayi], strlen(delay[delayi]));
    else delayi=0;
    break;
case 'm': // Auto sweep
    ibwrt(osc, "TRMD AUTO\0xa", 10);
    break;
case 'n': // Normal Sweep
    ibwrt(osc, "TRMD NORM\0xa", 10);
    break;
case 'o': // Single seep
    ibwrt(osc, "TRMD SINGLE\0xa", 12);
    break;
default:
    //tprintf ("\fUndefined key press\nPress enter key\nto continue\n");
    //input(buf);

    break;

}
delay_ms(250);
Dreset_z0rbuf();
}
if (done1) break;
}
}
}

```