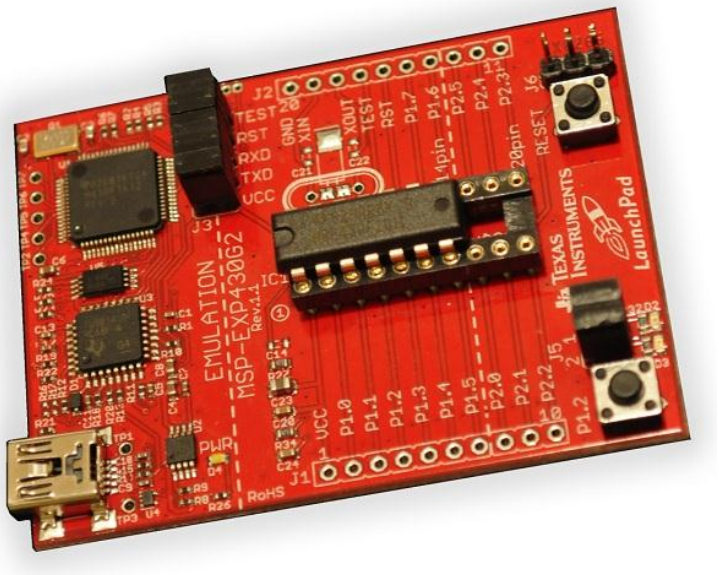
ĐO KHOẢNG CÁCH BẰNG SRF05 + MSP430 LAUNCHPAD KIT HIỆN THỊ LÊN MÁY TÍNH



Phần cứng

- MSP430 Launchpad Kit (Giá 4.3$)
- 1 cảm biến siêu âm SRF05
- Vài sợi Bus hoặc dây nối

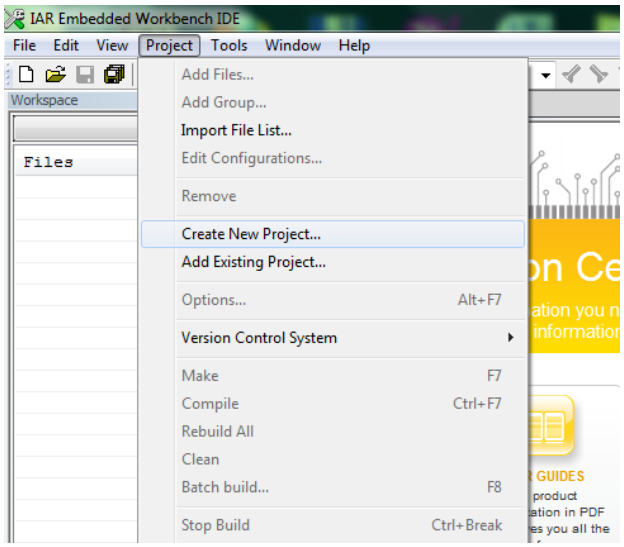
Phần mềm:

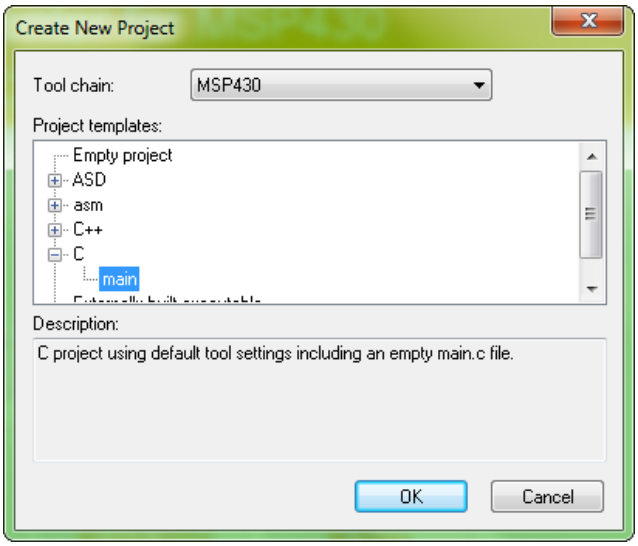
- Phần mềm **IAR Embedded**
- Phần mềm **PuTTy** (nhận dữ liệu UART)
  Link: http://www.putty.org/

Nguyễn Phước Lộc, K09 – Automation Control - ĐHBK Tp.HCM
Email: loc.plsoft@gmail.com

Viết chương trình cho KIT launchpad

Mở IAR Embedded Workbench  và làm theo các bước sau:

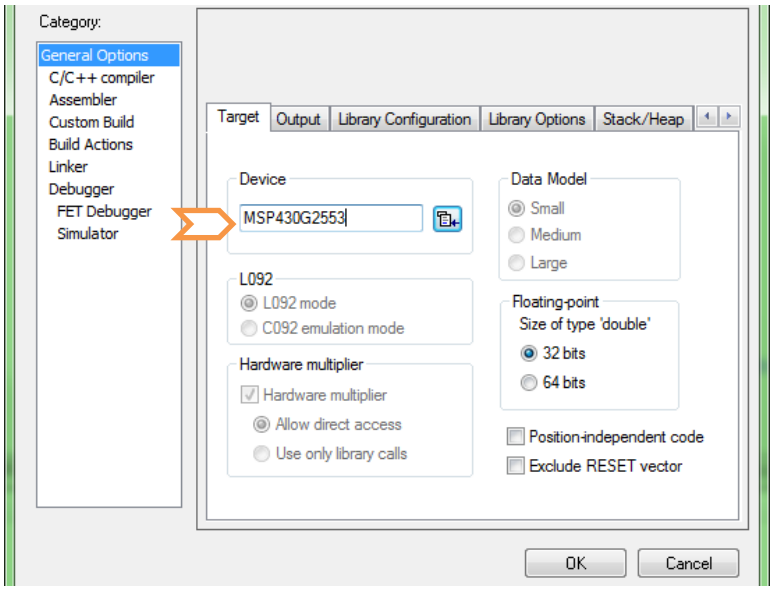


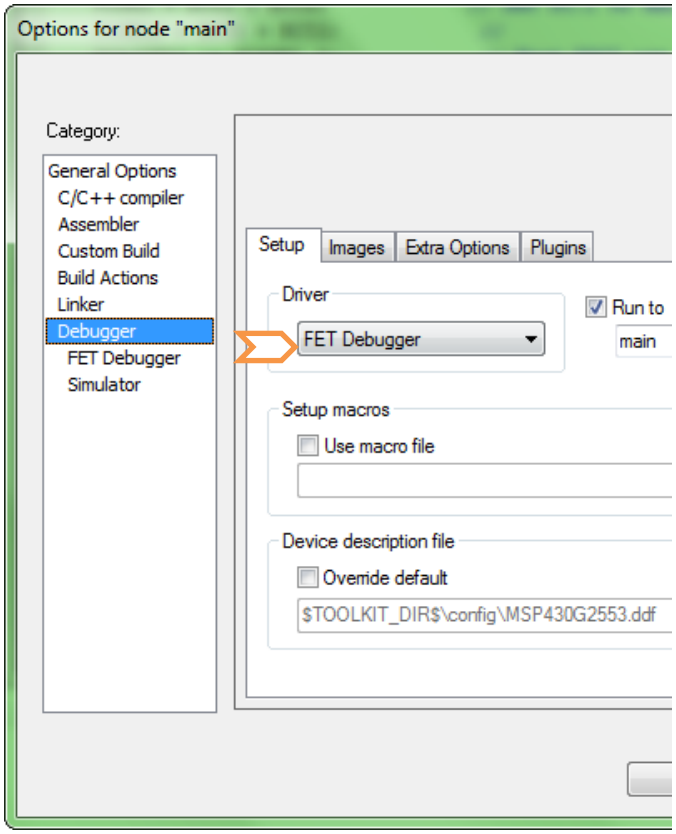


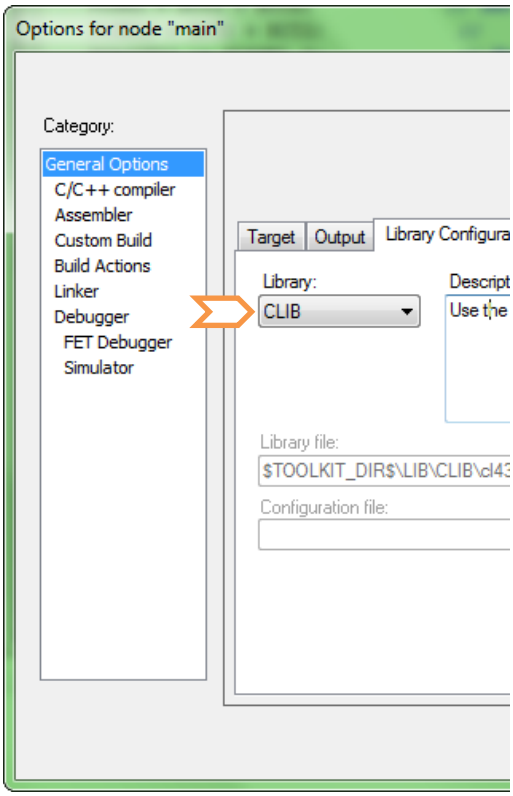
Đặt tên Project …

Nguyễn Phước Lộc, K09 – Automation Control - ĐHBK Tp.HCM
Email: loc.plsoft@gmail.com

Cài đặt các thông số sau:







Nguyễn Phước Lộc, K09 – Automation Control - ĐHBK Tp.HCM
Email: loc.plsoft@gmail.com

Viết chương trình cho "main.c" như sau:

```
#include <msp430g2553.h>
#include <stdbool.h>
///////////////Defines///////////////
#define LED1        BIT6
#define LED0        BIT0
#define DAT         BIT0 //P2.0   //input signal port
#define VCC         BIT5 //P1.5
#define GND         BIT4 //P1.4
#define KICH        BIT1
char charbuffer[8];
int i=0;
int j=0;
unsigned int capture_array[51];            // RAM array for captures
int tick=0;
int cap=0;
int pre_cap=0;
int first_pulse=0;

///////////////Function Protos///////////////
void TX(char *tx_message);
void DO_KHOANG_CACH(void);
static char *i2a(unsigned i, char *a, unsigned r);
char *itoa(int i, char *a, int r);


static char *i2a(unsigned i, char *a, unsigned r)
{
   if (i/r > 0) a = i2a(i/r,a,r);
   *a = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"[i%r];
   return a+1;
}

char *itoa(int i, char *a, int r)
{
   if ((r < 2) || (r > 36)) r = 10;
   if (i < 0)
   {
      *a = '-';
      *i2a(-(unsigned)i,a+1,r) = 0;
   }
   else *i2a(i,a,r) = 0;
   return a;
}

void TX(char *tx_message)
{
   unsigned int i=0; //Define end of string loop int
   char *message; // message variable
   unsigned int message_num; // define ascii int version variable
   message = tx_message; // move tx_message into message
   while(1)
   {
      if(message[i]==0) // If end of input string is reached, break loop.
      {break;}
      message_num = (int)message[i]; //Cast string char into a int variable
      UCA0TXBUF = message_num; // write INT to TX buffer
      i++; // increase string index
      __delay_cycles(10000); //transmission delay
```

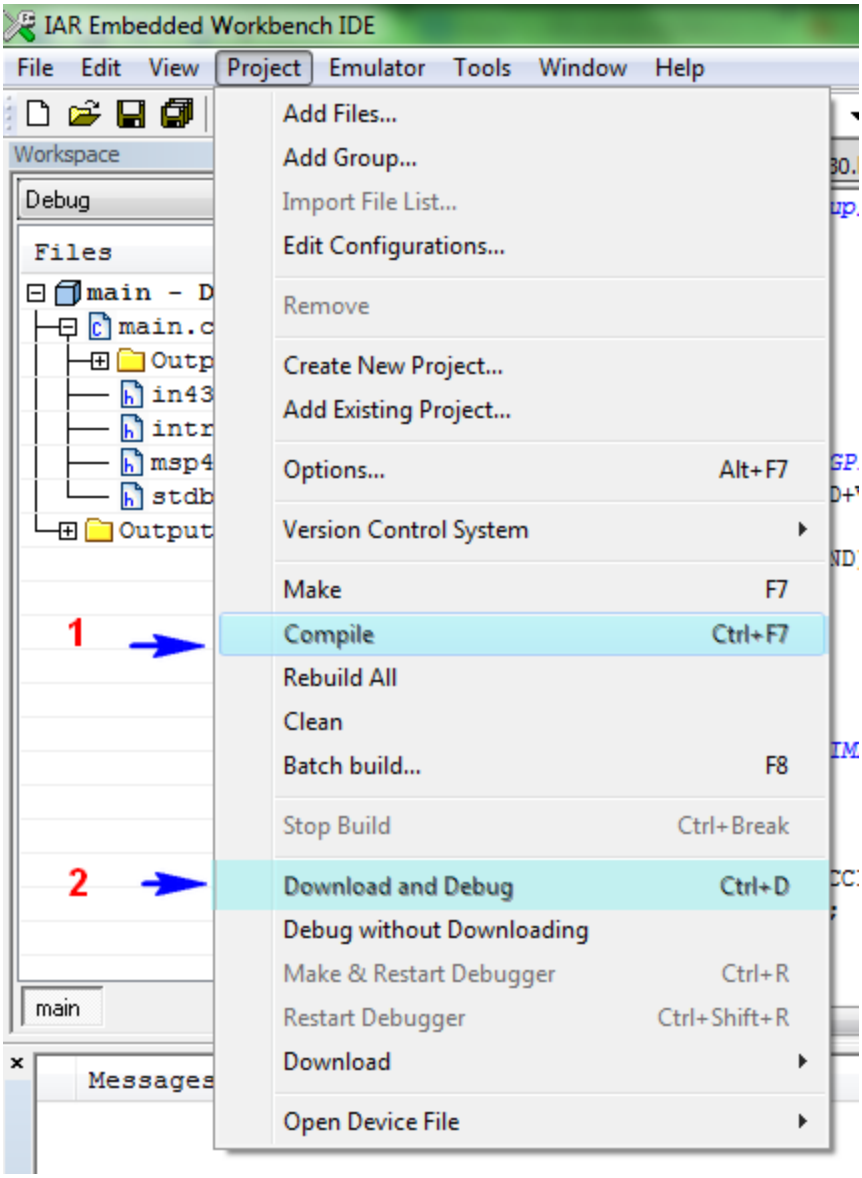
Nguyễn Phước Lộc, K09 – Automation Control - ĐHBK Tp.HCM
Email: loc.plsoft@gmail.com

```
      if(i>50) //prevent infinite transmit
      {
       //  P1OUT |= (LED1+LED0);
         break;
      }
   } // End TX Main While Loop
} // End TX Function

int main(void)
{ WDTCTL = WDTPW + WDTHOLD; // Stop watchdog timer
   //setup clock to 1MHZ
   BCSCTL1 = CALBC1_1MHZ;          // Set DCO to 1MHz
   DCOCTL = CALDCO_1MHZ;
   ////////////////USCI setup////////////////
   P1SEL = BIT1 + BIT2;          // Set P1.1 to RXD and P1.2 to TXD
   P1SEL2 = BIT1 + BIT2;          //
   UCA0CTL1 |= UCSSEL_2;          // Have USCI use SMCLK AKA 1MHz main CLK
   UCA0BR0 = 104;               // Baud: 9600, N= CLK/Baud, N= 10^6 / 9600
   UCA0BR1 = 0;              // Set upper half of baud select to 0
   UCA0MCTL = UCBRS_1;            // Modulation UCBRSx = 1
   UCA0CTL1 &= ~UCSWRST;          // Start USCI
   ////////////////General GPIO Defines////////////////
   P1DIR |= (LED0 + LED1+GND+VCC); //define output ports

   P1OUT &= ~(LED0 + LED1+GND); //turn ports low

   P2DIR |= KICH;
   P2IE |= DAT;
   P2IFG &= ~DAT;
   P2SEL = DAT;                        // Set P2.0 to TA0
   ////////////////SETUP TIMER

   TA1CCTL0 = CM_3 + SCS + CCIS_0 + CAP + CCIE; // falling edge + CCI0A (P2.0)// + Capture Mode +
Interrupt
   TA1CTL = TASSEL_2 + MC_2;              // SMCLK + Continuous Mode

   __enable_interrupt();

   while(1)
   {
     __delay_cycles(100000);
    DO_KHOANG_CACH();
   }
}

// Timer1 interrupt service routine
#pragma vector=TIMER1_A0_VECTOR
__interrupt void TIMER1(void)
{
   if ( (first_pulse==0) & (DAT == 1) )
      {
         P1OUT |= LED1;
       pre_cap=TA1CCR0;
       first_pulse=1;

      }
   else
      {
         P1OUT &= ~LED1;
         tick = TA1CCR0;
         cap = (tick- pre_cap)/58;
```

Nguyễn Phước Lộc, K09 – Automation Control - ĐHBK Tp.HCM
Email: loc.plsoft@gmail.com

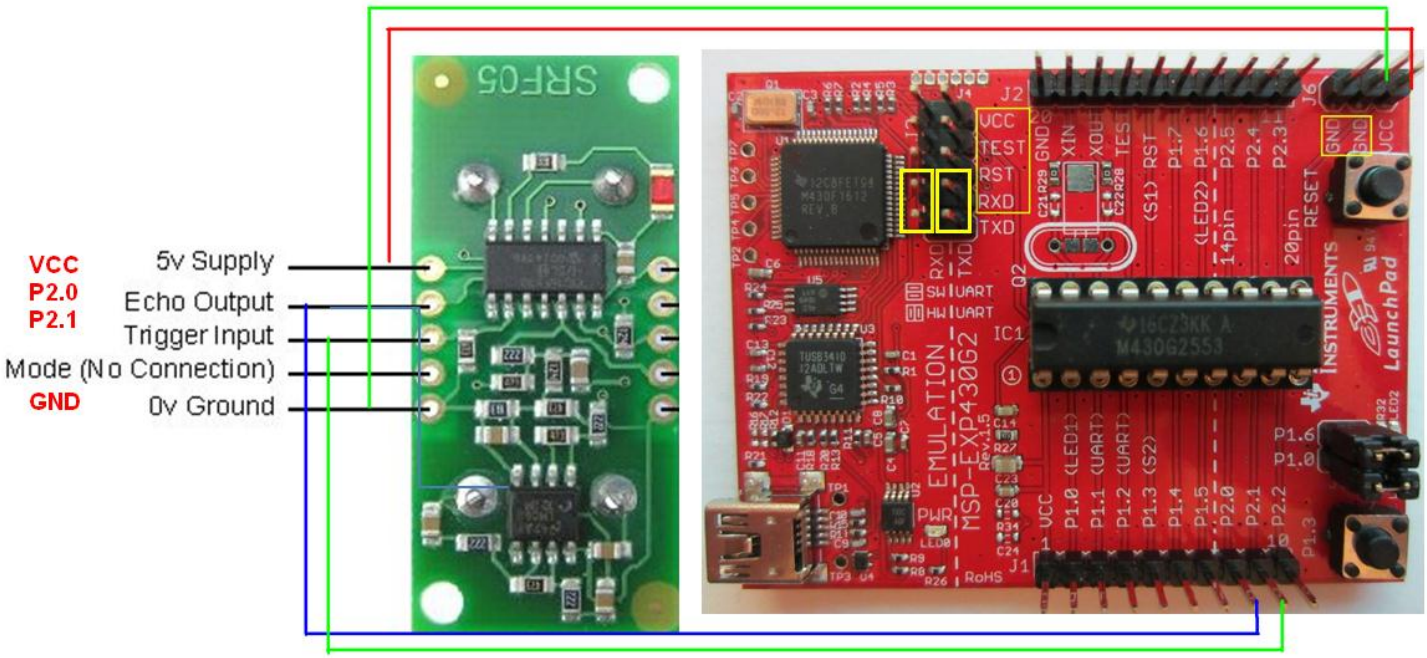
```
            first_pulse =0;
          TA1CCR0=0;

            itoa(cap, charbuffer, 10);
            TX(charbuffer);
            TX("\r\n");
        }

}
void DO_KHOANG_CACH(void)
{
  P2OUT |= KICH;
  __delay_cycles(20);
  P2OUT &= ~KICH;
}
```

Sau khi hoàn tất thì Download chương trình xuống KIT



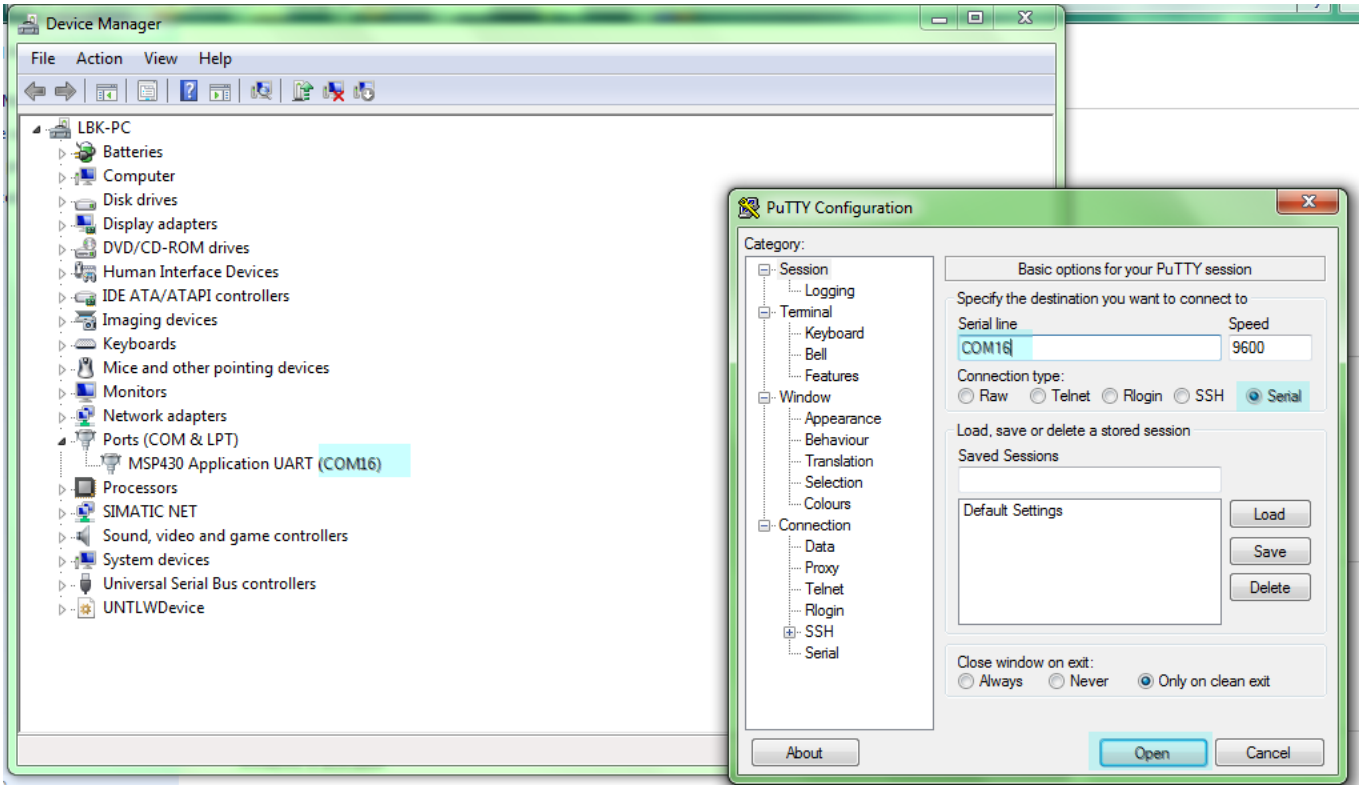
Nguyễn Phước Lộc, K09 – Automation Control - ĐHBK Tp.HCM
Email: loc.plsoft@gmail.com

Kết nối với SRF05



Chú ý: Đổi sang mode Hardware Uart bằng cách thay đổi các Switch kết nối như hình khoanh màu vàng.

Mở chương trình PUTTY (hi vọng bạn sẽ hiểu hình sau nói gì) hi



Nguyễn Phước Lộc, K09 – Automation Control - ĐHBK Tp.HCM
Email: loc.plsoft@gmail.com

Kết quả đo:

Giải thích sơ lược về chương trình.

Chương trình sử dụng Timer1 ở chế độ Capture để bắt xung từ chân P2.0. Cứ mỗi 100ms chương trình sẽ phát xung (20us) ra chân P1.1 để kích cho SRF05 hoạt động. SRF05 trả về giá trị đo được ở chân P2.0.

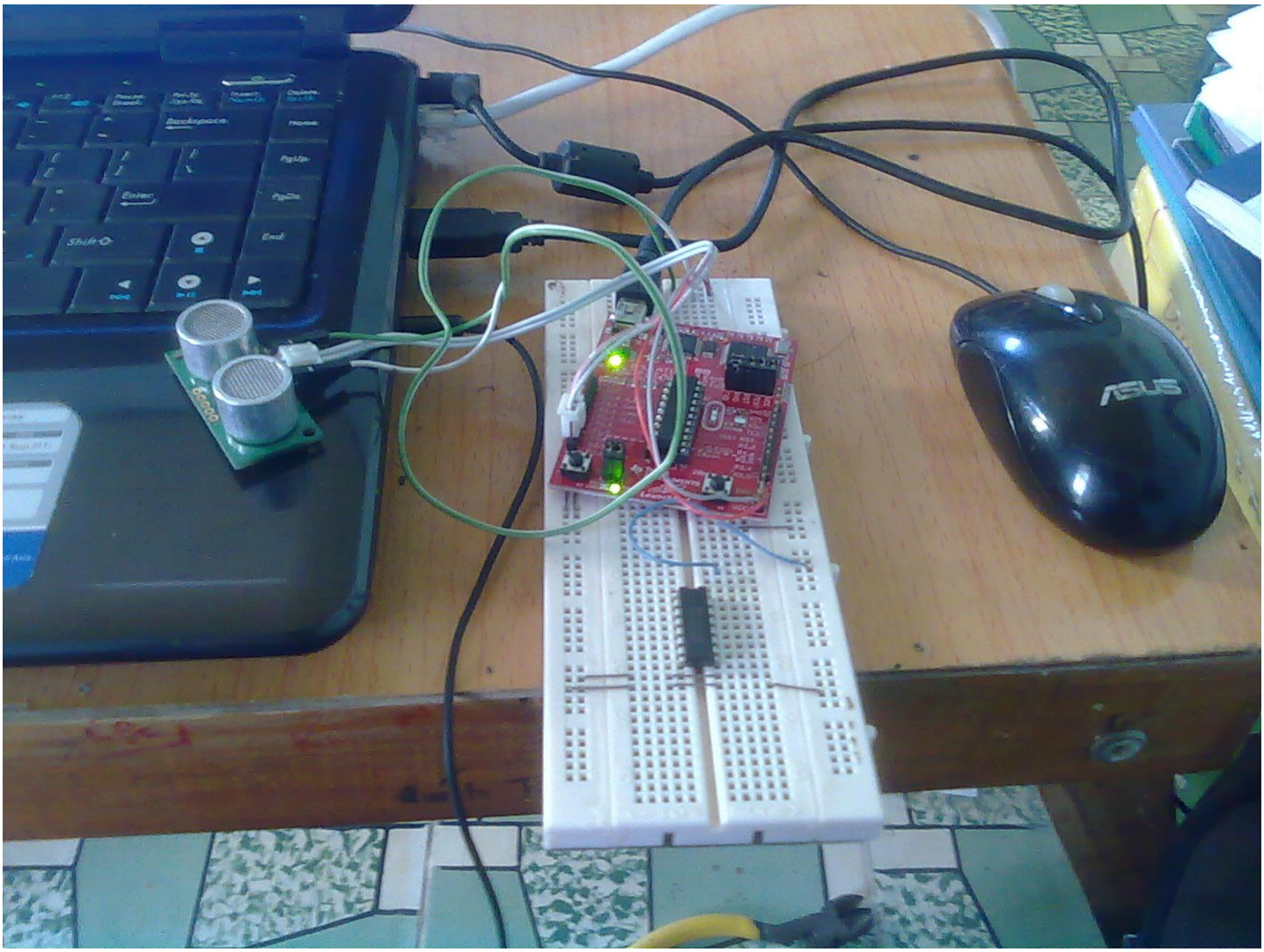
Giá trị đo được sẽ được chia cho 58 để ra khoảng cách tính bằng centimet. Sau đó giá trị được gửi lên máy tính.

Mọi việc đã xong rồi đó.

Good luck to you !

Nguyễn Phước Lộc, K09 – Automation Control - ĐHBK Tp.HCM
Email: loc.plsoft@gmail.com

Và đây là kết quả:



Nguyễn Phước Lộc, K09 – Automation Control - ĐHBK Tp.HCM
Email: loc.plsoft@gmail.com