TRƯỜNG ĐẠI HỌC NÔNG LÂM THÀNH PHỐ HỒ CHÍ MINH

KHOA CƠ KHÍ CÔNG NGHỆ

BỘ MÔN CÔNG NGHỆ KĨ THUẬT Ô TÔ

BÁO CÁO THÍ NGHIỆM Ô TÔ

Giáo viên hướng dẫn: **ThS.NGUYỄN TRỊNH NGUYÊN**

Sinh viên thực hiện: **HUỲNH TRỌNG NGHĨA – 18154075**

TP.HCM, ngày 19 tháng 02 năm 2022

PHẦN A: TÍNH TOÁN CÁC THÔNG SỐ CỦA 6 BÀI THÍ NGHIỆM

I.Tính hệ số cản lăn theo phương pháp chạy theo quán tính

$$f = \frac{\delta . v^2}{2 . g . S}$$

Trong đó:

f: hệ số cản lăn

δ: Hệ số tính đến các khối lượng quay của ô tô khi hộp số đã bị ngắt, chủ yếu là các bánh xe.

$\delta = 1{,}04 \pm 0{,}05 i_h^2 = 1{,}04 \pm 0{,}05.1^2 = 0{,}99 – 1{,}09$. Ta lấy $\delta = 1{,}04$ (với $i_h = 1$ là tỉ số truyền hộp số ở tay số 4)

g: gia tốc trọng trường, $g = 9{,}81\ m/s^2$

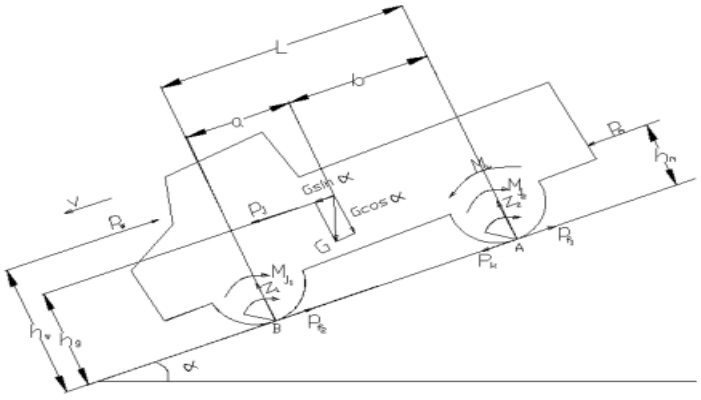
v: vận tốc của xe khi bắt đầu chạy theo quán tính (m/s), v = 20 km/h

S: quãng đường chạy theo quán tính của xe (m)

Bảng số liệu:

Thông số	Lần 1	Lần 2	Lần 3
Vận tốc (v)	5,55	5,55	5,55
Quãng đường (S)	36,9	37	37,2
Hệ số cản lăn (f)	0,044	0,044	0,044

II.Tính hệ số cản không khí theo phương pháp cho xe máy xuống dốc dưới tác dụng của lực trọng trường:



Ta có phương trình cân bằng lực kéo sau:

$$P_t = P_w + P_f$$

Trong đó:

P_i - Thành phần của trọng lượng, tác dụng song song với mặt đường dốc

Từ biểu thức 4.5 ta xác định được hệ số cản không khí K như sau:

$$K = \frac{G(\sin\alpha - f)}{f.v^2}\left(\frac{Ns^2}{m^2}\right)$$

Trong đó:

K: hệ số cản không khí

G: Trọng lực của xe, G = m.g = 103.9,81 = 1010,43 (N)

f: hệ số cản lăn

v: vận tốc của xe (m/s)

$\alpha = 10^o$

Bảng số liệu:

Thông số	Lần 1	Lần 2	Lần 3
Vận tốc (v)	5	4,7	5
Hệ số cản lăn (f)	0,044	0,044	0,044
Hệ số cản không khí (K)	119,09	134,78	119,09

III. Tính hệ số bám bằng phương pháp phanh:

Phương trình cân bằng động năng:

$$P_p.S = \frac{SG}{g}.\frac{v^2}{2}$$

Ta có hệ số bám φ:

$$\varphi = \frac{\delta.v^2}{2g.S_p}$$

Trong đó:

$\delta = 1,04 \pm 0,05i_h^2 = 1,04 \pm 0,05.1^2 = 0,99 – 1,09$. Ta lấy $\delta = 1,04$ (với i_h= 1 là tỉ số truyền hộp số ở tay số 4)

S_p: Quãng đường phanh (m)

v: vận tốc xe (m/s), v = 28 km/h

Bảng số liệu:

Thông số	Lần 1	Lần 2	Lần 3
Vận tốc (v)	7,78	7,78	7,78
Quãng đường phanh (S_p)	6,2	6,5	6
Hệ số bám (δ)	0,517	0,493	0,534

IV. Đo tiêu hao nhiên liệu bằng phương pháp cân trực tiếp:

Suất tiêu hao nhiên liệu:

$$g_e = \frac{G_{nl}}{N_e}\left(\frac{kg}{kW}.h\right)$$

Trong đó:

G_{nl}: Lượng tiêu hao nhiên liệu (kg/h)

N_e: Công suất động cơ (kW), N_e= 6,83 kW

Bảng số liệu:

Thông số	Lần 1	Lần 2	Lần 3
Lượng tiêu hao nhiên liệu (G_{nl})	0,112	0,110	0,12
Suất tiêu hao nhiên liệu	0,016	0,016	0,018

V. Tính quãng đường phanh:

Bảng số liệu:

Thông số	Lần 1	Lần 2	Lần 3
Vận tốc (v)	7,78	7,78	7,78
Quãng đường phanh (S_p)	6,2 m	6,5 m	6 m

VI. Tốc độ cực đại ở tay số 1:

Bảng số liệu:

Thông số	Lần 1	Lần 2	Lần 3
Vận tốc (v)	40 km/h	41 km/h	40 km/h
Thời gian (t)	5,4 s	5,5 s	6 s

PHẦN B: ĐỒ ÁN KHẢO SÁT GIA TỐC PHANH

Hiện nay hệ thống phanh ô tô không ngừng được cải tiến nhằm tăng cường tính năng an toàn cho xe. Lái xe có thể nâng cao hiệu suất phanh nếu hiểu cấu tạo, nguyên lý hoạt động của hệ thống phanh xe hơi.

Phanh là thiết bị cơ học có chức năng hạn chế chuyển động của bánh xe bằng cách tạo ra ma sát. Theo đó, hệ thống phanh khi hoạt động sẽ giúp kiểm soát việc giảm tốc độ hoặc dừng hẳn xe theo chủ ý của lái xe. Vấn đề đặt ra cho các nhà sản xuất xe hơi những yêu cầu cần phải cải tiến hệ thống phanh đảm bảo an toàn, có tính thẩm mỹ cao và mang lại sự thoải mái cho lái xe. Đến nay, các dòng ô tô hiện đại được trang bị hệ thống phanh đĩa thủy lực giúp tăng cường việc đảm bảo an toàn cho sử dụng.

Hệ thống phanh ô tô đạt chuẩn cần đáp ứng những tiêu chí như sau:

- Quãng đường phanh ngắn nhất trong điều kiện phanh đột ngột.
- Thời gian phanh nhỏ nhất thích ứng các tình huống bất ngờ.
- Gia tốc phanh chậm dần càng lớn mang lại hiệu quả phanh càng cao.
- Phanh êm dịu, đảm bảo tính ổn định trong mọi trường hợp.
- Điều khiển nhẹ nhàng, người lái không tốn nhiều sức khi sử dụng.
- Phân bố mô men đều trên các bánh xe phù hợp với tải trọng lực bám.
- Không bị hiện tượng bó phanh.
- Thoát nhiệt tốt, nâng cao tuổi thọ của linh kiện trong hệ thống phanh.
- Kết cấu gọn nhẹ, dễ chẩn đoán hư hỏng trong mọi điều kiện.

Mục đích thí nghiệm: khảo sát gia tốc phanh nhằm biết được tình trang phanh hiện tại mà từ đó đề ra biện pháp cải tiến tăng tính an toàn cho con người trong quá trình sử dụng ô tô.

Phương pháp thí nghiệm: Cảm biến gia tốc được gắn cố định trên mặt phẳng ngang trên xe. Giá trị gia tốc được vi điều khiển Atmega16 chuyển đổi thành giá trị ADC. Giá trị ADC được vi điều khiển truyền lên máy tính qua bộ UART được mã hóa dưới dạng mã ASCII. Mã ASCII được truyền liên tục và lưu giữ trong bộ nhớ đệm trong Labview và được chương trình Labview xử lí thông qua API Ni-Visa để chuyển ngược lại từ giá trị ADC về giá trị gia tốc và xuất ra đồ thị gia tốc.

I. CÁC PHẦN CỨNG TIẾN HÀNH THÍ NGHIỆM

1. ATMEGA16:

Atmega16 là vi điều khiển 8 bits do hãng Atmel sản xuất với cấu trúc tập lệnh đơn giản hóa RISC. Với khả năng thực hiện mỗi lệnh trong vòng một chu kỳ xung clock, ATmega16 có thể đạt được tốc độ 1MIPS trên mỗi MHz (1 triệu lệnh/s/MHz). Ngoài ra, ATmega16 có các đặc điểm sau: 16KB bộ nhớ Flash với khả năng đọc trong khi ghi, 512 byte bộ nhớ EEPROM, 1KB bộ nhớ SRAM, 32 thanh ghi chức năng chung, 32 đường vào ra chung, 3 bộ định thời/bộ đếm, ngắt nội và ngắt ngoại, USART, giao tiếp nối tiếp 2 dây, 8 kênh ADC 10 bit, ... ATmega 16 hỗ trợ đầy đủ các chương trình và công cụ phát triển hệ thống như: trình dịch C, macro assemblers, chương trình mô phỏng/sửa lỗi, kit thử nghiêm, …

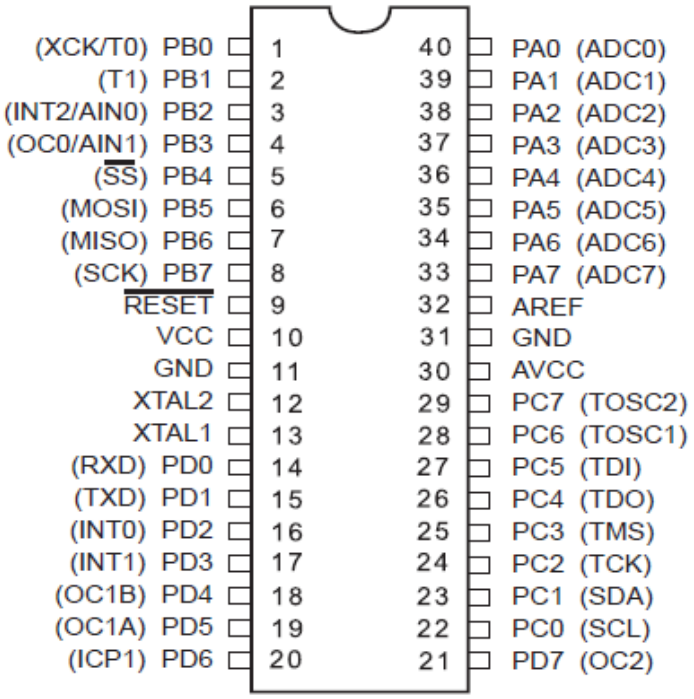
ATmega16 là một loại Vi điều khiển có nhiều tính năng đặc biệt thích hợp cho việc giải quyết những bài tóan điều khiển trên nền vi xử lý.

- Các loại vi điều khiển AVR rất phổ biến trên thị trương Việt Nam nên không khó khăn trong việc thay thế và sửa chữa hệ thống lúc cần.
- Giá thành của dòng Vi Điều Khiển này khá phải chăng
- Các phần mềm lập trình và mã nguồn mở có thể tìm kiếm khá dễ dàng trên mạng. Các thiết kế demo nhiều nên có nhiều gợi ý tốt cho người thiết kế hệ thống.

ATmega16 là vi điều khiển 8bit dựa trên kiến trúc RISC. Với khả năng thực hiện mỗi lệnh trong vong một chu kỳ xung clock, Atmega16 có thể đạt

được tốc độ 1MIPS trên mỗi MHz (1triệu lệnh/s/MHz),các lệnh được xử lý nhanh hơn,tiêu thụ năng lượng thấp.

Sơ đồ vi điều khiển ATMEGA16



Ý nghĩa các chân:

- Chân Vcc: Chân số 10 là VCC cấp điện áp nguồn cho Vi điều khiển. Nguồn điện cấp là +5V±0.5.
- Chân GND: Chân số11 và chân số 31 nối GND(hay nối Mass). Khi thiết kế cần sử dụng một mạch ổn áp để bảo vệ cho Vi điều khiển, cách đơn giản là sử dụng IC ổn áp 7805.
- Port A (PA): Port A gồm 8 chân (từ chân 33 đến 40) có chức năng: đầu vào cho chuyển đổi ADC.
- Port B (PB): Port PB gồm 8 chân (từ chân 1 đến chân 8), ngoài có chức năng làm các đường xuất/nhập thì còn có nhiều chức năng phụ khác.

- Port C (PC): Port C gồm 8 chân (từ chân 22 đến chân 29): Nếu giao tiếp JTAG được kích hoạt điện trở trên các PC5(TDI), PC3 (TMS), PC2 (TCK) sẽ được kích hoạt ngay cả khi khởi động lại (reset).
- Port D (PD): Cổng nhập xuất dữ liệu song song D (PORTD) nó có thể đc sử dụng các chức năng đặc biệt thay vì nhập xuất dữ liệu.

(Chip ATMEGA 16)

2. MẠCH CHUYỂN USB UART TTL FT232RL:

Mạch chuyển USB UART TTL FT232RL là module mạch chuyển đổi tín hiệu USB sang UART (serial) sử dụng IC FT232RL của FTDI.
Mạch có sẵn ổn áp và mạch dao động tích hợp bên trong, hoạt động ổn định hơn so với các dòng USB to serial khác. Mạch tích hợp LED Status TX và RX giúp theo dõi trực tiếp trạng thái tín hiệu.
Mạch chuyển USB UART TTL FT232RL có chất lượng tốt, thiết kế nhỏ gọn dễ dàng tích hợp vào hệ thống. Không sử dụng thạch anh ngoài nên có độ bền và độ ổn định cao, thích hợp cho các ứng dụng cần chuyển USB sang UART TTL hoặc giao tiếp Vi điều khiển với máy tính.

Ứng dụng thường dùng

- Làm mạch giao tiếp UART với vi điều khiển, mạch nạp cho các bản Arduino không tích hợp mạch nạp onboard như Arduino Pro Mini, ChipiPro Lite, ...
- Làm mạch trung gian giao tiếp giữa board mạch với máy tính, hữu ích khi truyền dữ liệu từ board mạch lên máy tính để kiểm tra, phân tích.

- Làm mạch nạp cho các dòng vi điều khiển như ARM, AVR, PIC, ... có hỗ trợ nạp thông qua giao tiếp UART.

Thông số kĩ thuật

- IC nạp và giao tiếp: FT23RL (hãng FTDI).
- Nguồn: 5 VDC từ cổng USB.
- Điện áp ngõ ra: 5 VDC và 3.3 VDC.
- Dòng điện tối đa ngõ ra: 500 mA.
- Tốc độ baudrate tùy chỉnh.
- Driver hỗ trợ MacOS, Windows, Linux.

Kích thước: 17 x 32.5mm.

Sơ đồ ra chân

- GND (Power): Chân điện áp âm (0V).
- CTS (Input): Clear to Send, chân này không sử dụng, đã được nối 0V trên mạch.
- 5V (Power): Chân nguồn dương.
- TXD (Output): Truyền tín hiệu UART.
- RXI (Input): Nhận tín hiệu UART.
- DTR (Output): Data Terminal Ready, reset Arduino khi nạp chương trình.

(*Mạch chuyển USB UART TTL FT232RL)*

3. CẢM BIẾN GIA TỐC GY-61 ADXL335:

*Cảm Biến GY-61 Analog Accelerometer ADXL335 là modun cảm biến độ nghiêng 3 trục, tiêu thụ năng lượng thấp, độ phân giải cao. Modun ADXL 335 thường dùng trong các thiết bị di động, có chức năng đo gia tốc trọng trườ*ng

tĩnh, *trong các ứng dụ*ng đo góc nghiêng. *Ngoài ra nó còn đo gia tốc động từ các chuyển động hoặc rung động của vật thể.*

Thông số kĩ thuật:

- *Nguồn sử dụng: 3-5VDC*

- *Chuẩn giao tiếp: Điện áp Analog* trên 3 trục x, y, z

- *3 Axis sensing*

- *Kích thước: 15.7x20.3mm*

- Chip cảm biến: ADXL335
- Dòng điện: 400uA
- Full scale range: +/-3g
- Nhiệt độ hoạt động: -40'C~ +85'C
- Độ nhạy: 300mV/g
- Độ chính xác: ±10%
- Phù hợp khi kết nối với các hệ thống 5V hoặc 3.3V
- Điện áp đầu ra analog ở mức giữa: 1.65V

4. CÁC THIẾT BỊ KHÁC:

- *Đế nạ*p ATMEGA 16 (AVR T49*)*

- *Cap nối USB UART*

- *Mạch USBISP nạp chương trình cho ATMEGA16*

- *Cáp dữ liệu USB Mini B*

- *Dây nố*i, *dây cấp nguồn 5V*

II. PHẦN MỀM LẬP TRÌNH CHO THÍ NGHIỆM

1. CODEVISION AVR 2.05:

a. Giới thiệu phần mềm:

CodeVisionAVR - là một môi trường phát triển tích hợp phần mềm cho vi điều khiển Atmel AVR. Nó cung cấp sự hỗ trợ rộng rãi cho các thiết bị AVR và tạo ra một đoạn mã nhỏ gọn và hiệu quả.

CodeVisionAVR bao gồm các thành phần sau:

- Trình biên dịch ngôn ngữ C cho AVR;
- Trình biên dịch hợp ngữ cho AVR;
- Các máy phát điện của mã chương trình ban đầu cho phép khởi tạo thiết bị ngoại vi;
- Module giao tiếp với debug board STK-500;
- Module tương tác với các lập trình viên;
- Terminal.

CodeVisionAVR cho tập tin đầu ra là:

- HEX, BIN hoặc tập tin ROM để nạp vào thiết bị thông qua lập trình;
- COFF - file có chứa thông tin cho trình gỡ lỗi;
- OBJ - file.

Hiện nay, CodeVisionAVR bao gồm các thư viện và các ví dụ sau đây:

- Alphanumeric LCD modules for up to 4x40 characters;
- Philips I²C Bus;
- National Semiconductor LM75 Temperature Sensor;
- Maxim/Dallas Semiconductor DS1621 Thermometer/Thermostat;
- Philips PCF8563 and PCF8583 Real Time Clocks;
- Maxim/Dallas Semiconductor DS1302 and DS1307 Real Time Clocks;
- Maxim/Dallas Semiconductor 1 Wire protocol;
- Maxim/Dallas Semiconductor DS1820/DS18B20/DS1822 1 Wire Temperature - Sensors;
- Maxim/Dallas Semiconductor DS2430/DS2433 1 Wire EEPROMs;
- SPI;

- MMC/SD/SD HC FLASH Memory Card drivers and FAT12, FAT16, FAT32 access libraries;
- Power management;
- Delays;
- BCD and Gray code conversion.

Nó hỗ trợ hầu hết các vi điều khiển Atmel AVR. Phiên bản mới thêm hỗ trợ cho vi điều khiển với một kernel ATxmega.

(CodeVisionAVR V2.05.0)

b. Tiến hành lập trình:

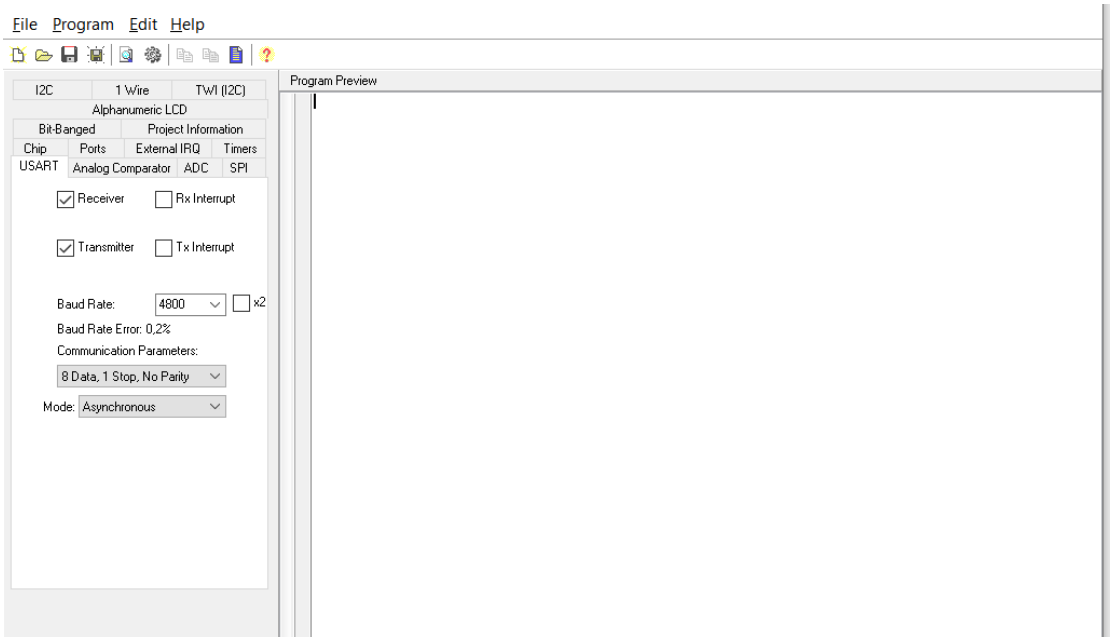
Bước 1: Mở phần mềm CV AVR tạo chương trình mới → Project → Atmega.

Bước 2: Khai báo

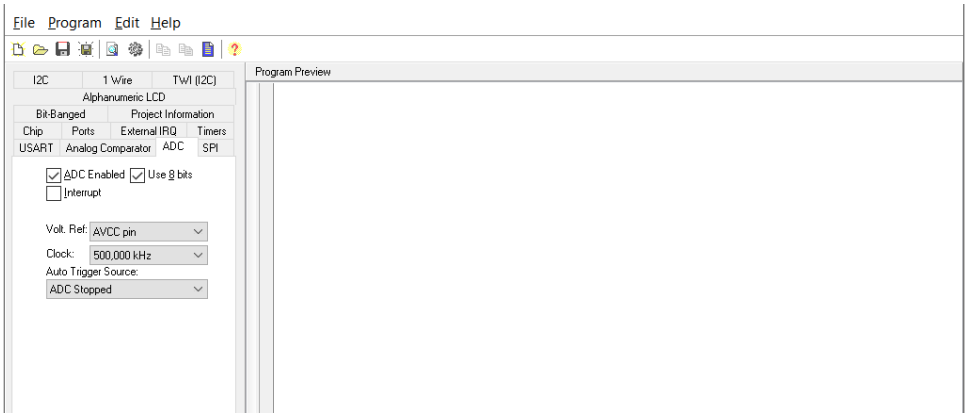
- Chọn chip ATMEGA 16

- USART

Bật Receiver → Transmister → Baud rate 4800



- ADC mở : chọn 8 bit → Vol ref : AVCC Pin .



Bước 3: Viết chương trình:

```
#include <mega16.h>
#include <stdio.h>
#include <delay.h>

#define ADC_VREF_TYPE 0x60

// Declare your global variables here
unsigned char AdcValue;

// Read the 8 most significant bits
// of the AD conversion result
```

```
unsigned char read_adc (unsigned char adc_input)
{
ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
// Delay needed for the stabilization of the ADC input voltage
delay_us (10);
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCH;
}

 void main(void)
{
// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 4800
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x0C;

// ADC initialization
// ADC Clock frequency: 500,000 kHz
// ADC Voltage Reference: AVCC pin
// ADC Auto Trigger Source: ADC Stopped
```

```
// Only the 8 most significant bits of
// the AD conversion result are used
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x81;

while (1)
    {
    AdcValue  = read_adc(0);
    putchar(AdcValue);
    delay_ms(100);
    }
}
```

Bước 4 : Biên dịch chương trình, debug lỗi, tạo file hex

Giải thích chương trình :

Khai báo thư viện <mega16.h> để sử dụng các hàm hỗ trợ chip atmega16

Khai báo thư viện <delay.h> để sử dụng hàm tạo trễ delay_ms()

Khai báo thư viện <stdio.h> để sử dụng hàm putchar() truyền dữ liệu

```
#include <mega16.h>
#include <stdio.h>
#include <delay.h>
```

Khởi tạo giá trị cho các thanh ghi bộ ADC 8-bits

```
// ADC initialization
// ADC Clock frequency: 500,000 kHz
// ADC Voltage Reference: AVCC pin
// ADC Auto Trigger Source: ADC Stopped
// Only the 8 most significant bits of
// the AD conversion result are used
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x81;
```

Trình biên dịch hỗ trợ tự động tạo hàm đọc và chuyển đổi giá trị ADC:

```
unsigned char read_adc(unsigned char adc_input)
{
ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
// Delay needed for the stabilization of the ADC input voltage
delay_us(10);
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCH;
}
```

Khởi tạo giá trị cho các thanh ghi của bộ giao tiếp UART

```
// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 4800
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x0C;
```

Khai báo một biến AdcValue để lưu trữ giá trị ADC mà vi điều khiển đọc được

```
// Declare your global variables here
```

unsigned char AdcValue ;

Biến AdcValue sử dụng kiểu dữ liệu unsigned char vì giá trị ADC không có giá trị âm và bộ ADC-8bits chỉ có giá trị từ 0-255

```
while (1)
    {
    AdcValue  = read_adc(0);
    putchar(AdcValue);
    delay_ms(100);
    }
```

Dùng hàm read_adc() để đọc giá trị ADC chân A.0 rồi lưu trữ giá trị đọc được vào biến toàn cục AdcValue đã khai báo trước đó.

Dùng hàm putchar() của thư viện <stdio.h> để truyền giá trị ADC đã đọc được lên máy tính thông qua bộ giao tiếp truyền thông UART với tốc độ truyền baudrate 4800 bps đã được thiết lập trước đó.

Hàm delay_ms() tạo độ trễ giúp đọc và truyền dữ liệu được ổn định hơn.

Cả 3 hàm đó được đặt trong một vòng lặp while(1) để vi điều khiển luôn luôn đọc giá trị và truyền giá trị, vì là vòng lặp while này luôn đúng nên vi điều khiển chỉ dừng lại khi tác động ngắt vật lí bên ngoài từ người dùng.

2. CHƯƠNG TRÌNH NẠP PROGISP:

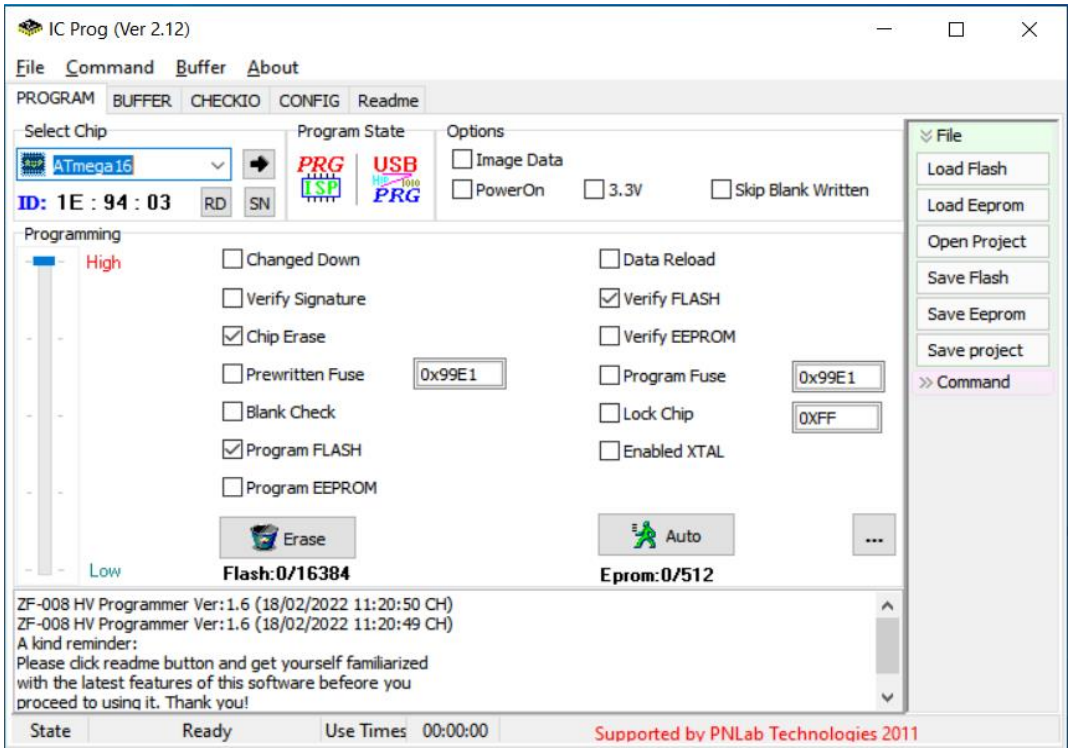
a. Giới thiệu:

Progisp là một phần mềm thông dụng giúp chúng ta nạp code từ phần mềm codevision AVR vào chip vi điều khiển .

(Phần mềm nạp Progisp)

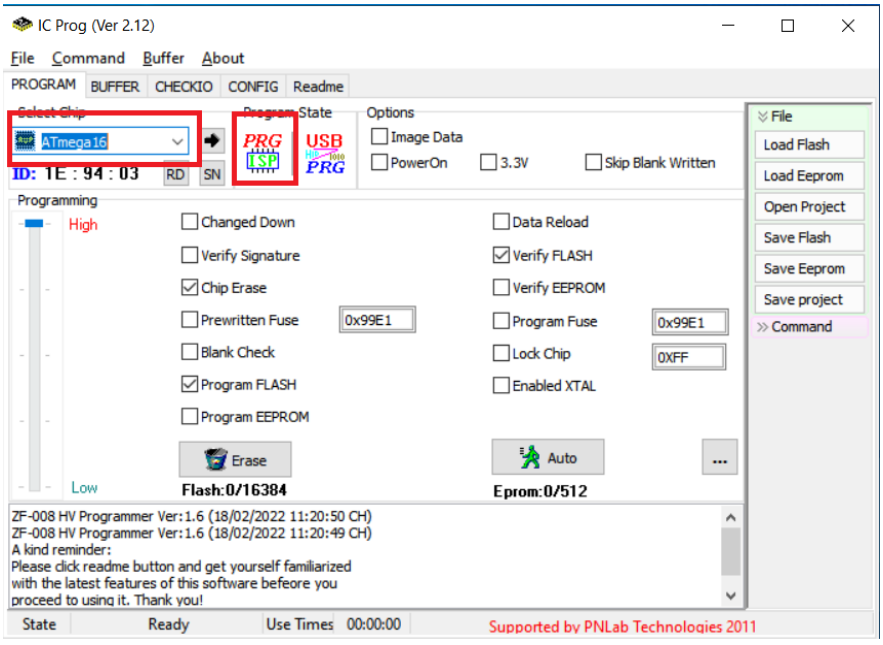
b. Tiến hành nạp chương trình:

Bước 1: Chạy chương trình nạp Progisp



Bước 2 :

- Chương trình nhận mạch nạp USBISP
- Chọn loại chip ATMEGA16



Bước 3:

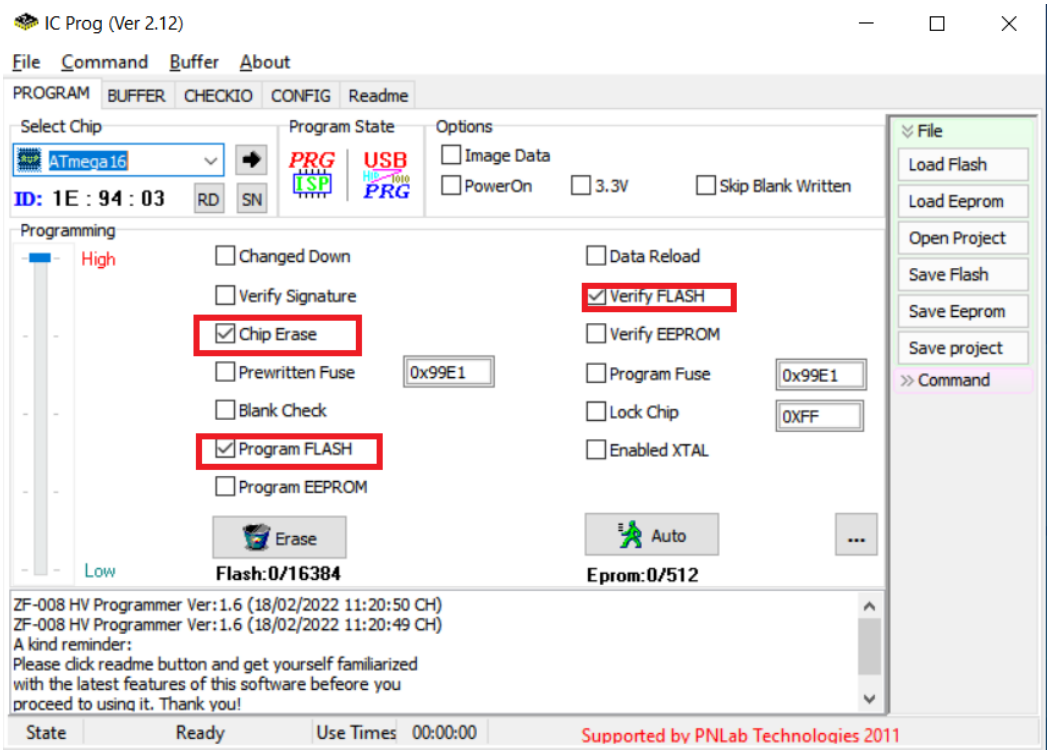
Các thông số quan trọng khi nạp chương trình cho ATMEGA16 (thông số quan trọng chương trình sẽ tự setup tương ứng , bạn không cần thay đổi gì cả)

Chip Erase: Cho phép xóa chip

Programe Flash: Nạp file Hex cho chip

Verify Flash: Kiểm tra lỗi của chương trình Flash

Data Reload: Tự động cập nhật dữ liệu khi thay đổi file Hex



Bước 4:

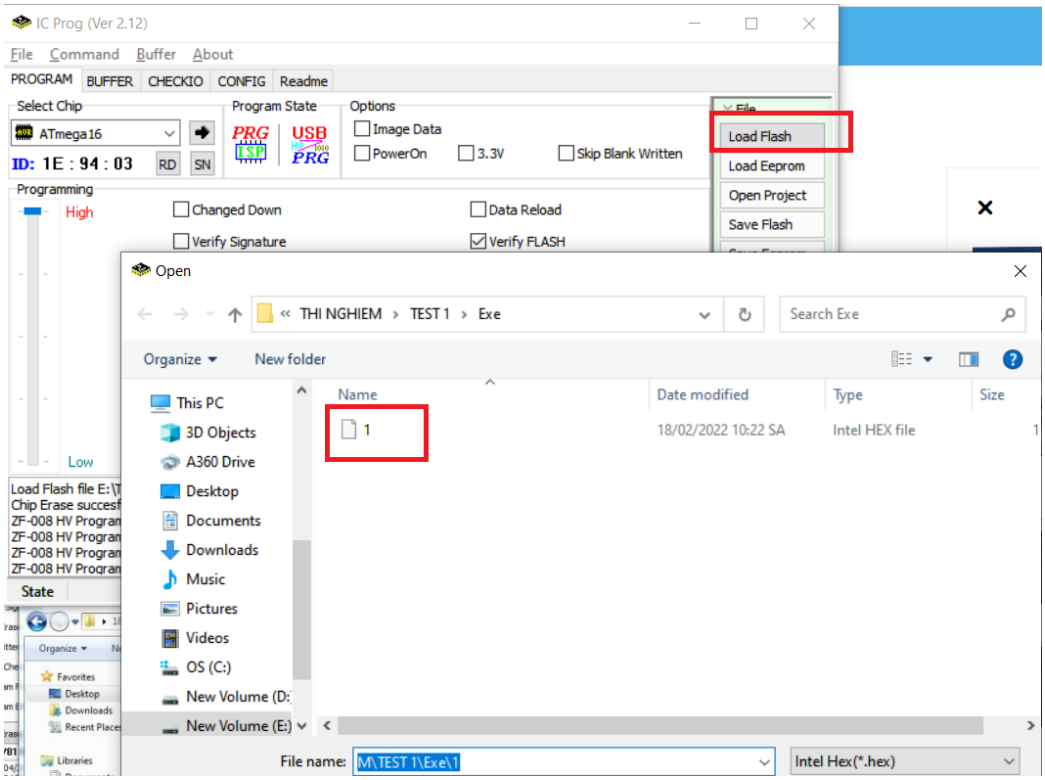
Xóa chip (Xóa chương trình cũ có trong chip)

Bước 5: Nạp chương trình Flash

- 1: Click **Load Flash**

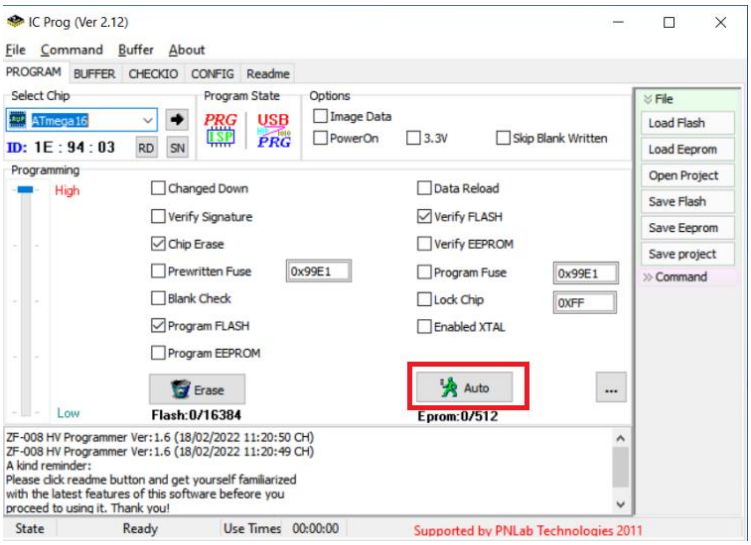
- 2: Tìm **file hex trong chương trình của bạn**

- 3: **Open** thôi



Bước 6 : Load chương trình

Ấn Auto để load chương trình



Bước 7: Hoàn Thành

3. PHẦN MỀM MÔ PHỎNG PROTEUS:

a. Giới thiệu:

Proteus là phần mềm cho phép mô phỏng hoạt động của mạch điện tử bao gồm phần thiết kế mạch và viết chương trình điều khiển cho các họ vi điều khiển như MCS-51, PIC, AVR, …

Proteus là phần mềm mô phỏng mạch điện tử của Lancenter Electronics, mô phỏng cho hầu hết các linh kiện điện tử thông dụng, đặc biệt hỗ trợ cho cả các MCU như PIC, 8051, AVR, Motorola.

Phần mềm bao gồm 2 chương trình: ISIS cho phép mô phỏng mạch và ARES dùng để vẽ mạch in. Proteus là công cụ mô phỏng cho các loại Vi Điều Khiển khá tốt, nó hỗ trợ các dòng VĐK PIC, 8051, PIC, dsPIC, AVR, HC11, MSP430, ARM7/LPC2000 ... các giao tiếp I2C, SPI, CAN, USB, Ethenet, ... ngoài ra còn mô phỏng các mạch số, mạch tương tự một cách hiệu quả. Proteus là bộ công cụ chuyên về mô phỏng mạch điện tử.

Những khả năng khác của ISIS là:

• Tự động sắp xếp đường mạch và vẽ điểm giao đường mạch.

• Chọn đối tượng và thiết lập thông số cho đối tượng dễ dàng

• Xuất file thống kê linh kiện cho mạch

• Xuất ra file Netlist tương thích với các chương trình làm mạch in thông dụng.

• Đối với người thiết kế mạch chuyên nghiệp, ISIS tích hợp nhiều công cụ giúp cho việc quản lý mạch điện lớn, mạch điện có thể lên đến hàng ngàn linh kiện.

• Thiết kế theo cấu trúc (hierachical design)

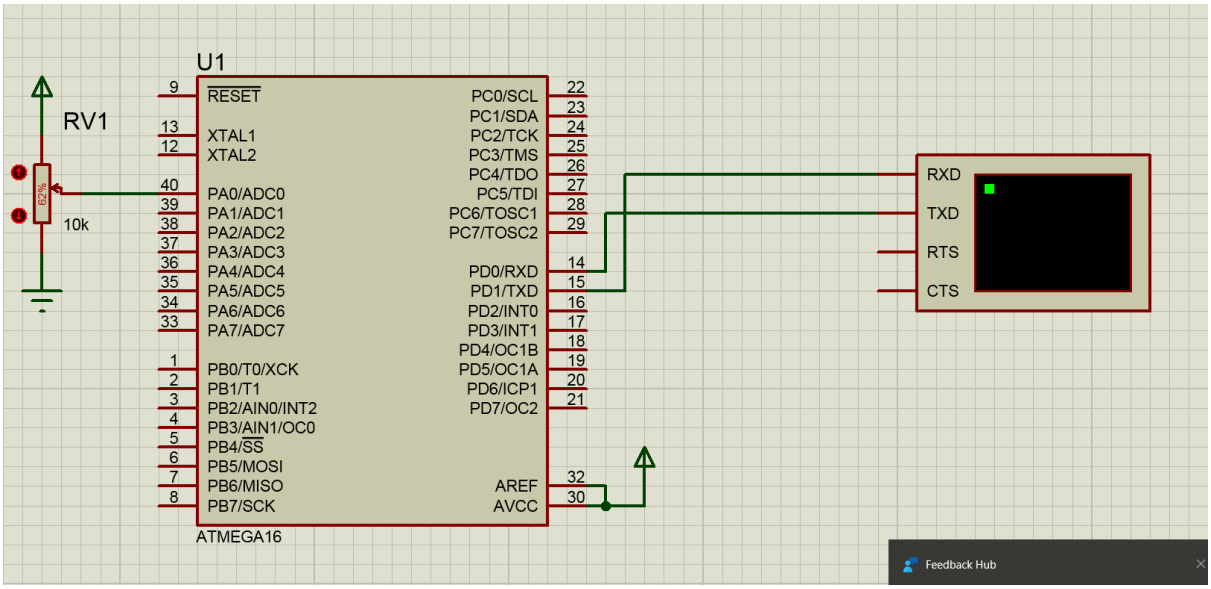
• Khả năng tự động đánh số linh kiện

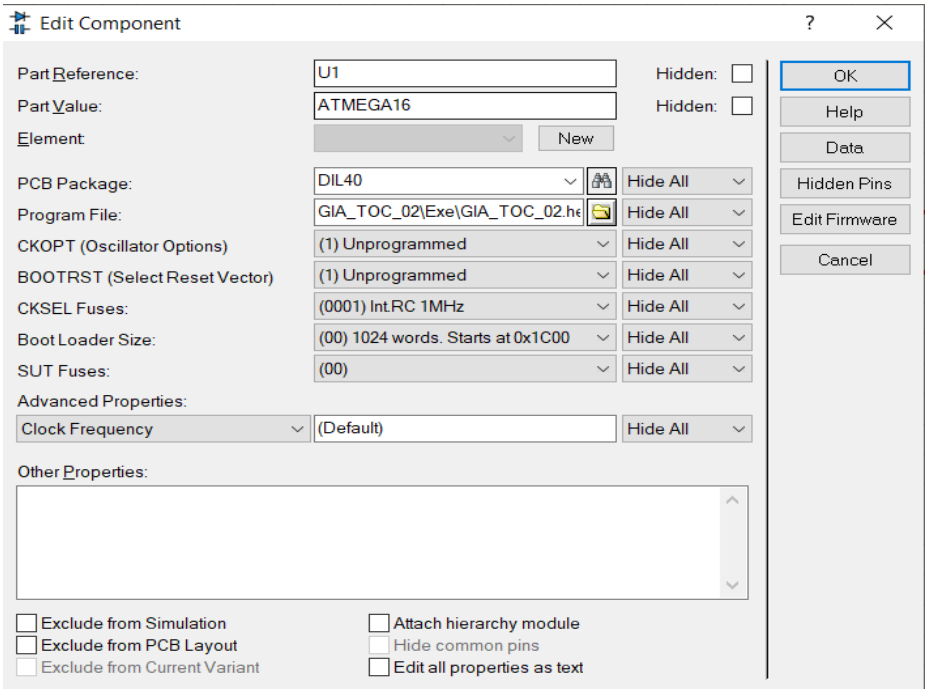
b. Mô phỏng:

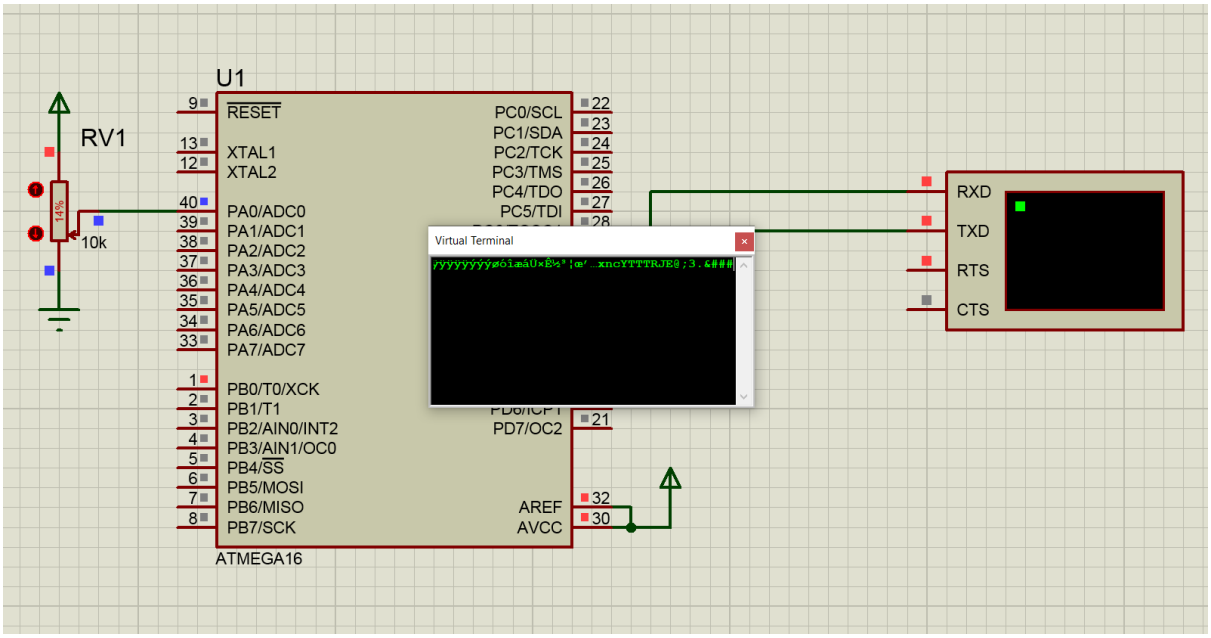
Vì Proteus không hỗ trợ module cảm biến gia tốc ADXL335 nên ta dùng biến trở để mô phỏng

Sử dụng Virtual Terminal để giả lập truyền dữ liệu từ vi điều khiển lên máy tính qua giao tiếp truyền thông UART, thiết lập tốc độ baudrate là 4800 bps đồng bộ tốc độ truyền nhận cho dữ liệu

Nạp file hex đã tạo vào chip và thiết lập tần số hoạt động là xung nội 1Mhz như đã khai báo bên source code.







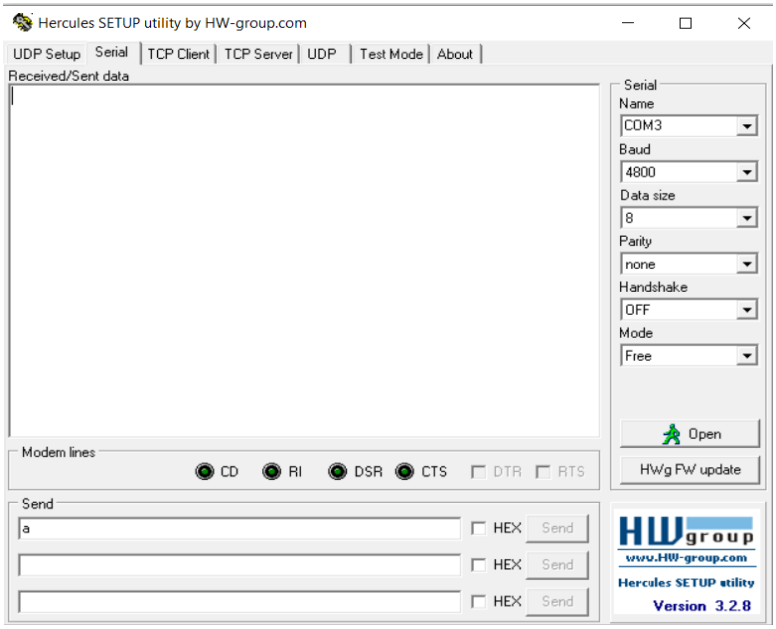
4. PHẦN MỀM TEST CỔNG COM (HERCULES) :

a. Giới thiệu:

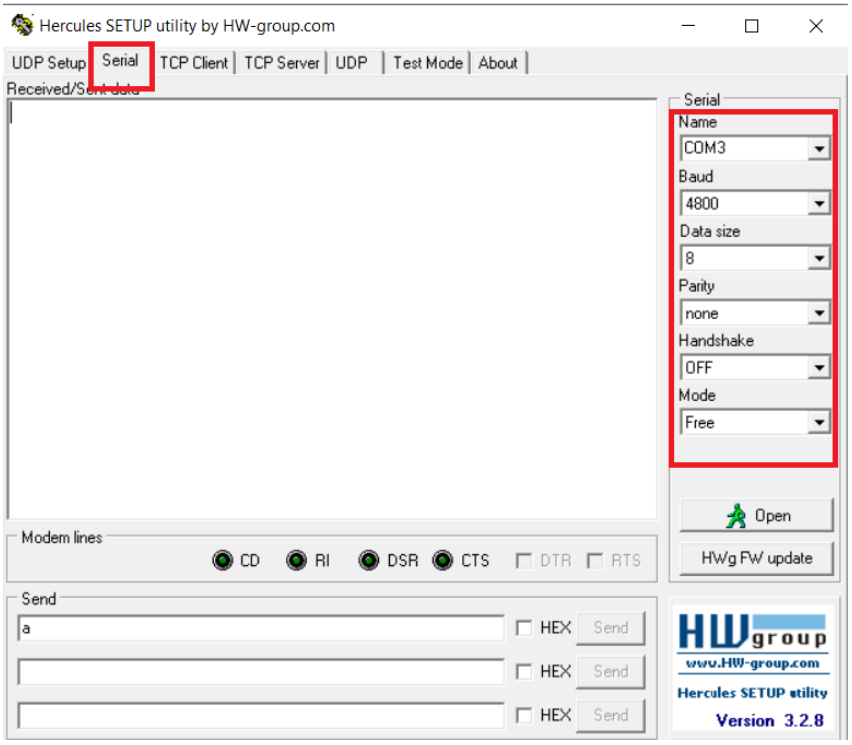
Hercules Terminal cũng như các phần mềm Terminal khác dùng để đọc chuỗi nhận được thông qua các cổng khác nhau trên máy tính.

b. Tiến hành kiểm tra

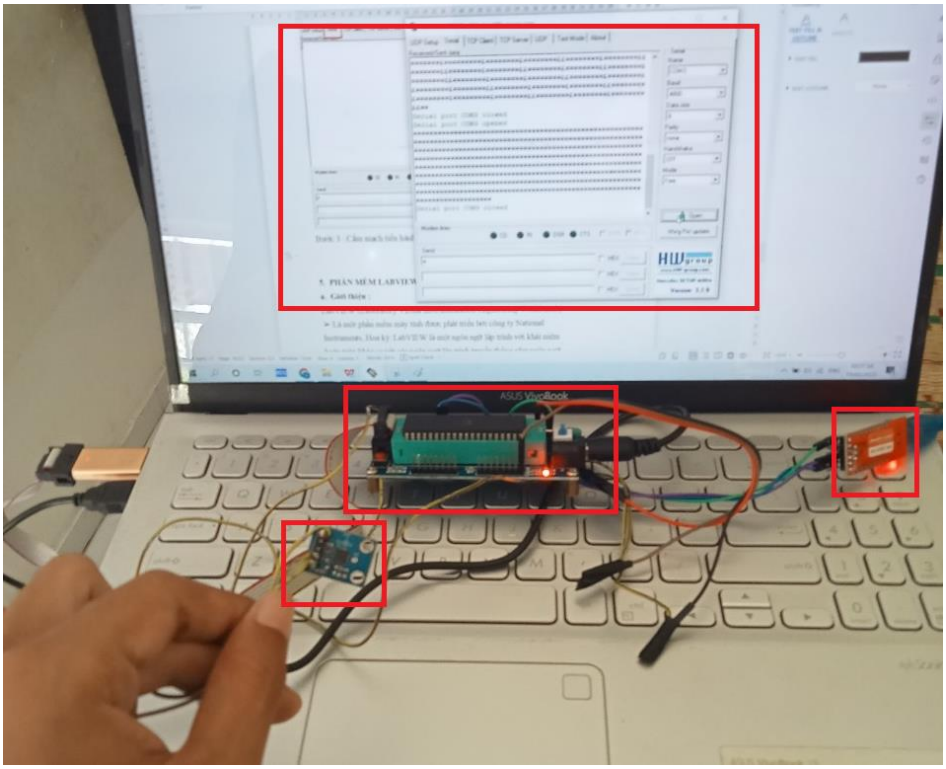
Bước 1 : Mở phần mềm Hercules

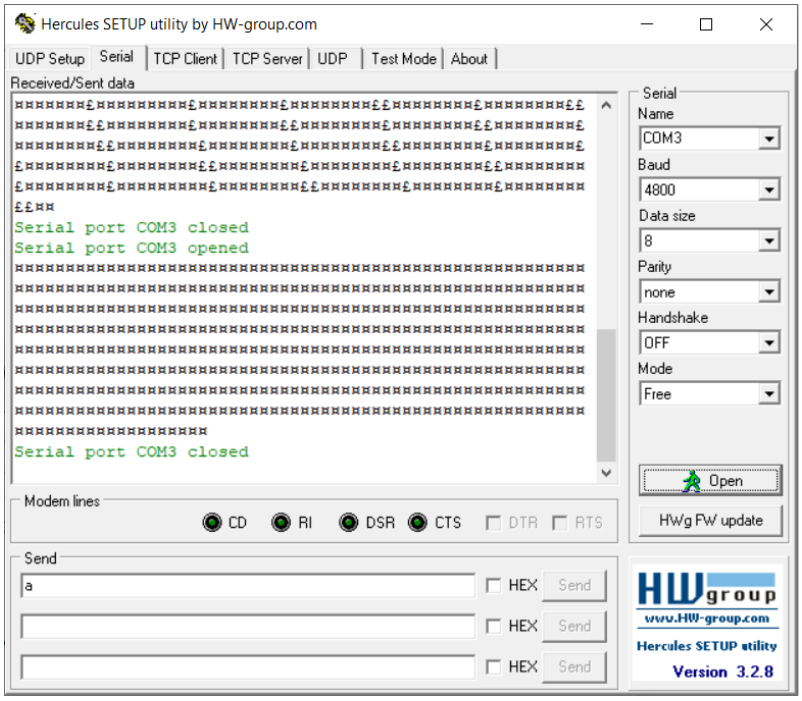


Bước 2: Setup giá trị



Bước 3 : Cắm mạch tiến hành kiểm tra





5. PHẦN MỀM LABVIEW:

a. Giới thiệu:

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench)

➢ Là một phần mềm máy tính được phát triển bởi công ty National Instruments, Hoa kỳ. LabVIEW là một ngôn ngữ lập trình với khái niệm hoàn toàn khác so với các ngôn ngữ lập trình truyền thống như ngôn ngữ C, Pascal ...

➢ LabVIEW diễn đạt cú pháp thông qua hình ảnh trực quan trong môi trường soạn thảo.

➢ LabVIEW được gọi tên khác là lập trình G (viết tắt của Graphical, nghĩa là đồ họa).

Chức năng:

❖ Thu thập các tín hiệu như: cảm biến dòng, cảm biến áp, cảm biến nhiệt độ, cảm biến lưu lượng, hình ảnh từ webcam, encoder, ...

❖ Giao tiếp với các thiết bị ngoại vi thông qua một số chuẩn giao tiếp: RS232, RS485, USB, PCI, Ethernet.

❖ Mô phỏng và xử lý các tín hiệu thu nhận được để phục vụ các mục đích nghiên cứu hay mục đích của hệ thống mà người lập trình mong muốn.

❖ Xây dựng các giao diện người dùng một cách nhanh chóng và thẩm mỹ hơn nhiều so với các ngôn ngữ khác như: Visual Basic, Matlab, …

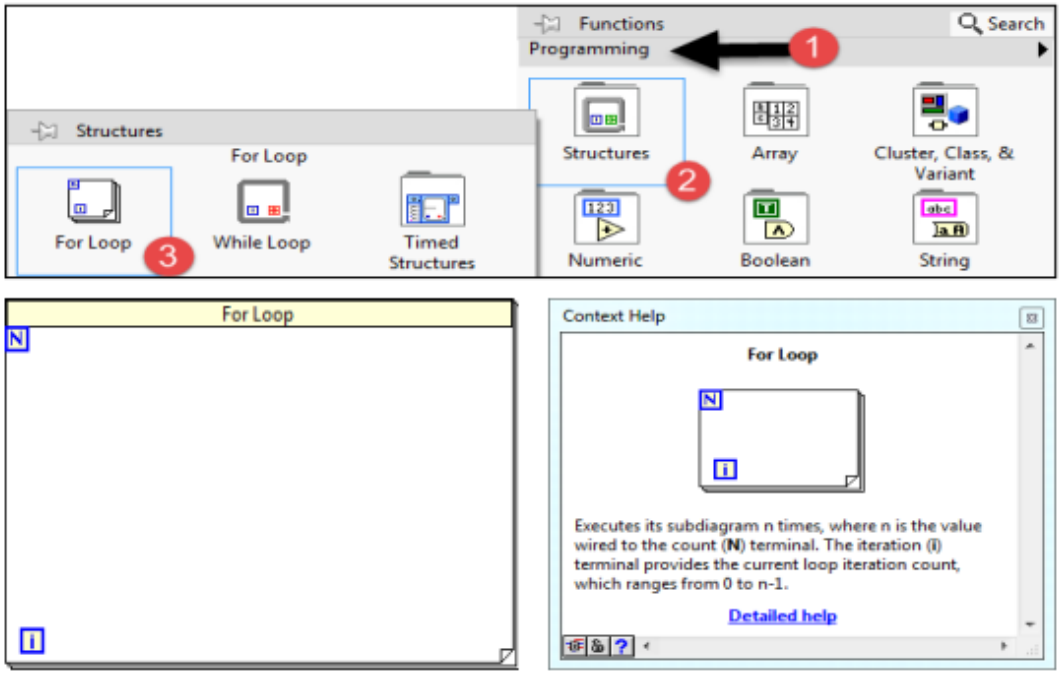
❖ Cho phép thực hiện các thuật toán điều khiển như: PID, Fuzzy, LQR, … một cách nhanh chóng thông qua các chức năng tích hợp sẵn trong LabVIEW.

❖ Cho phép kết hợp với nhiều ngôn ngữ lập trình truyền thống như: C, C++

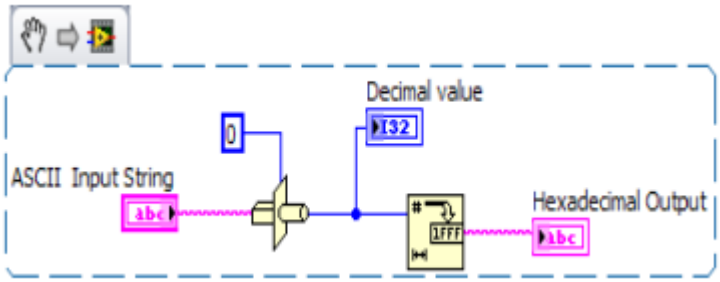
b. Lập trình:

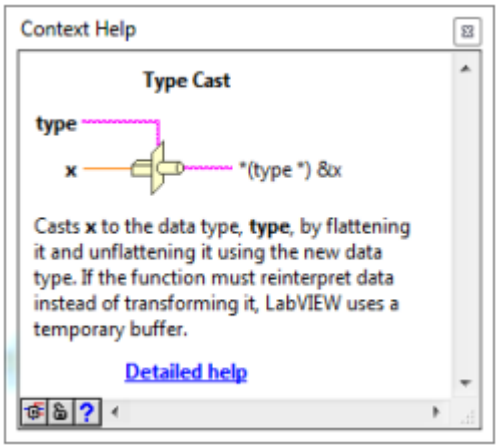
Bước 1: "For Loop" trong Labview:

Programming/Structures/While Loop

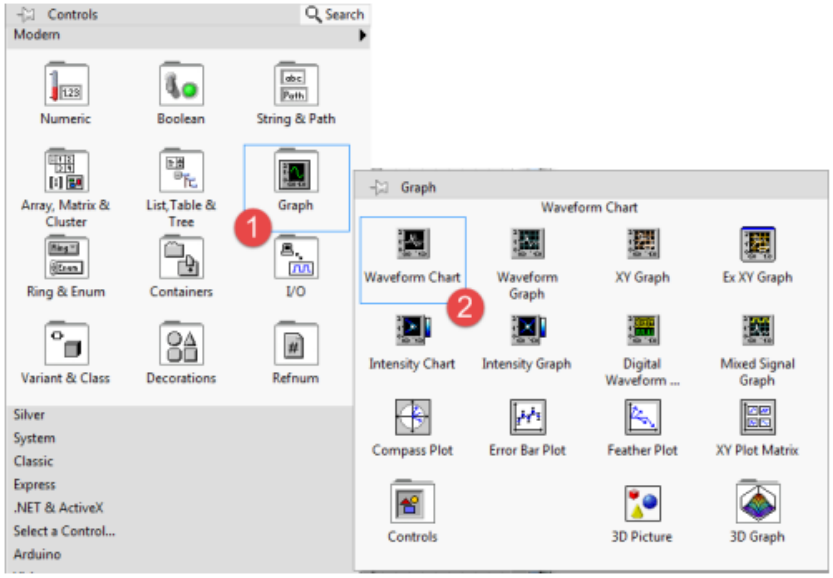


Bước 2: Ép kiểu từ mã ASCII sang số nguyên

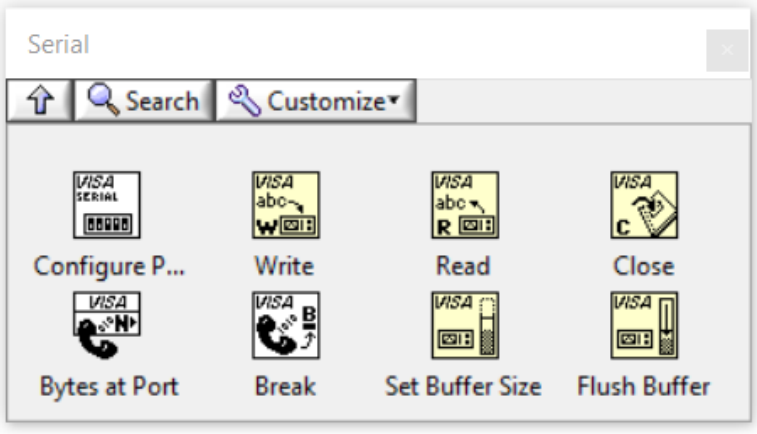




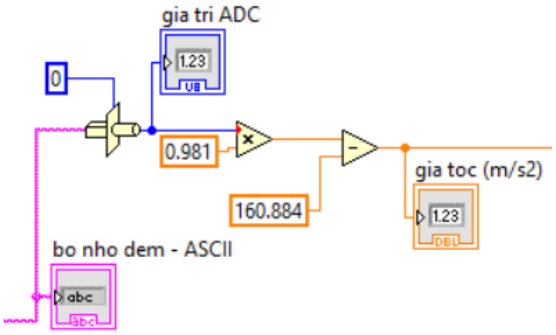
Bước 3: Đồ thị hiện thị tín hiệu dạng sóng



Serial-VISA là một cấp cao API được sử dụng để giao tiếp với các bus các thiết bị đo đạc. Nó là một nền tảng, bus và môi trường độc lập



Bước 4: Chuyển đổi từ mã ASCII về giá trị gia tốc



Theo datasheet của ADXL335 thì:

195 mV = 9.81 m/s2 = 1g

Vì ADC 8 bit nên giá trị từ 0 – 255 có 256 khoảng giá trị

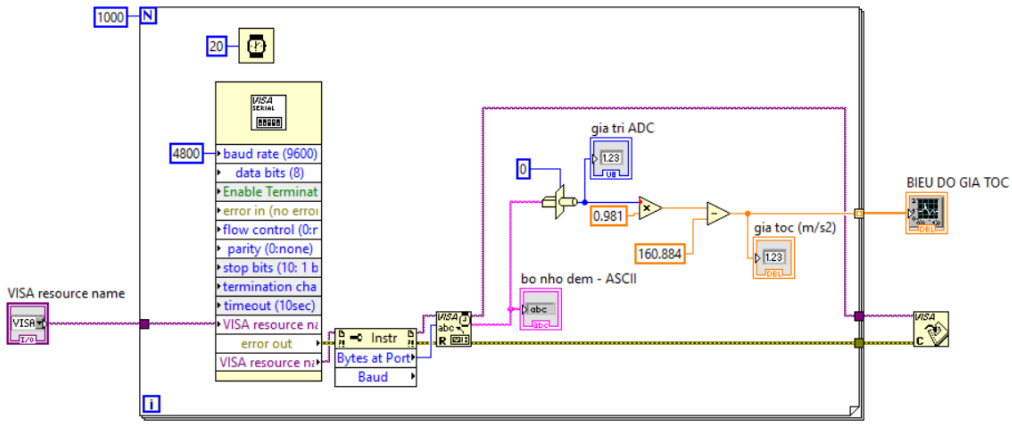
Vậy với điện áp tham chiếu 5V:

$$1\ ADC = 1000.\frac{5}{256} = 19.5\ mV$$

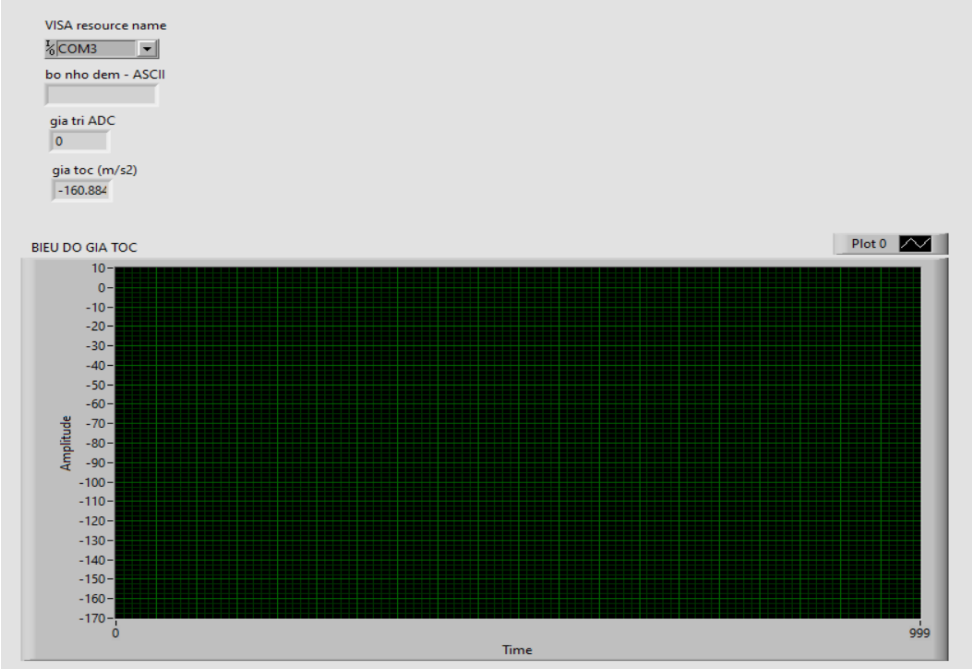
Nên

1 ADC = 0.981 m/s2 = 1/10 g

Block Diagram



Front panel

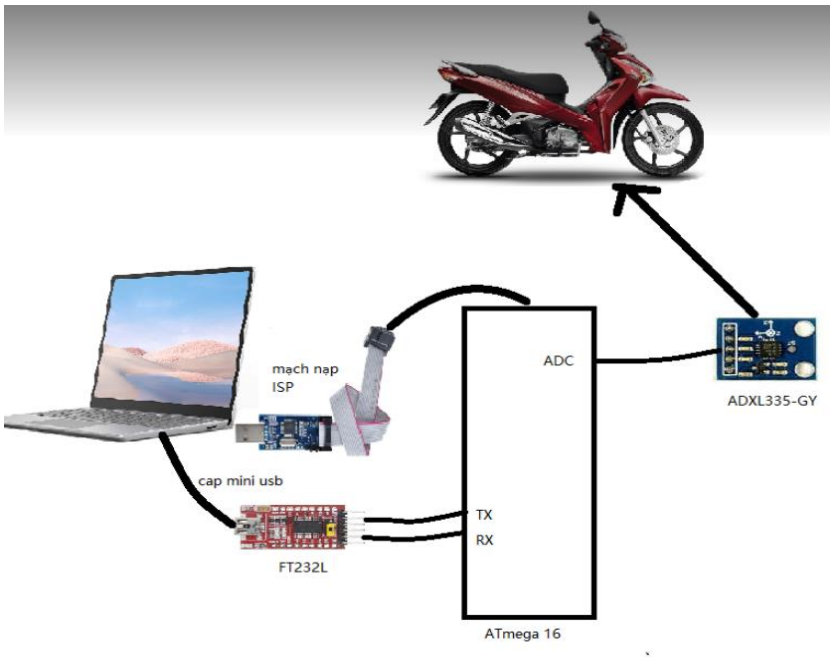


III. TIẾN HÀNH THÍ NGHIỆM

1. BỐ TRÍ THÍ NGHIỆM:

Tiến hành thí nghiệm trên xe FUTURE NEO

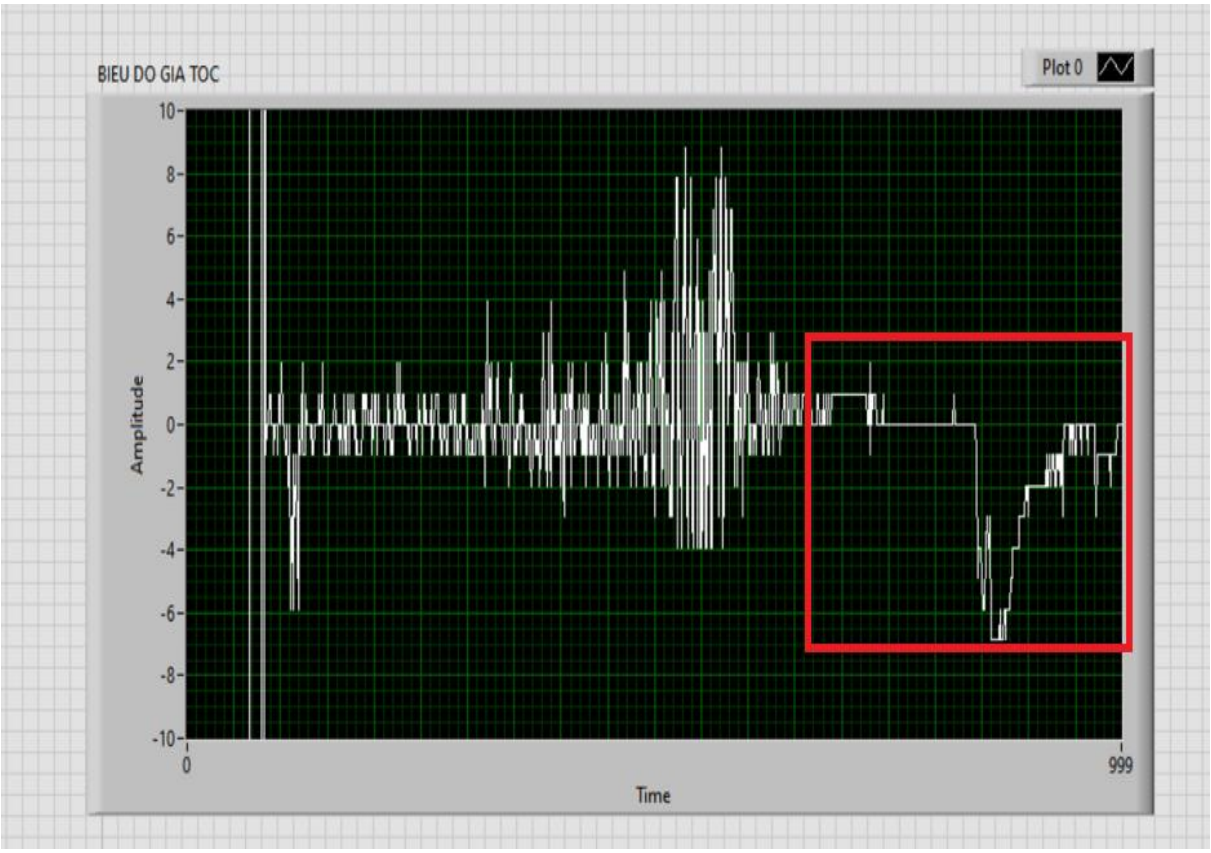
ẢNH BỐ TRÍ THÍ NGHIỆM



2. THỰC HIỆN THÍ NGHIỆM

https://drive.google.com/file/d/10R1CcQkPfyqIFIUq-MRV2GwGqSJ2gGI4/view?usp=sharing

3. KẾT QUẢ THÍ NGHIỆM:



KẾT QUẢ SỐ LIỆU:

A	B
822	0
823	0
824	0
825	0
826	0
827	0
828	0
829	0
830	0
831	0
832	0
833	0
834	0
835	0
836	0
837	0
838	0
839	0
840	0
841	0
842	0
843	0
844	-0,981
845	-4,905
846	-3,924
847	-3,924
848	-3,924
849	-3,924
850	-4,905
851	-5,886
852	-5,886
853	-5,886
854	-3,924
855	-2,943
856	-2,943
857	-3,924
858	-3,924
859	-2,943
860	-6,867
861	-6,867
862	-6,867
863	-6,867
864	-6,867
865	-6,867
866	-6,867
867	-6,867
868	-6,867
869	-5,886
870	-6,867
871	-6,867
872	-6,867
873	-6,867
874	-5,886
875	-6,867
876	-5,886
877	-5,886
878	-5,886
879	-5,886
880	-5,886
881	-4,905
882	-4,905
883	-3,924
884	-3,924
885	-3,924
886	-3,924
887	-3,924
888	-3,924
889	-3,924
890	-2,943
891	-2,943
892	-2,943
893	-2,943
894	-2,943
895	-2,943
896	-1,962
897	-1,962
898	-1,962
899	-2,943
900	-1,962
901	-1,962
902	-1,962
903	-1,962
904	-1,962
905	-1,962
906	-1,962
907	-1,962
908	-1,962
909	-1,962
910	-1,962
911	-1,962

Đối chiếu với tiêu chuẩn TCVN 6824:2001 về phanh của xe mô tô gắn máy, do trong thí nghiệm ta kết hợp giữa phanh trước và phanh sau:

TCVN **TIÊU CHUẨN VIỆT NAM**

TCVN 6824 : 2001

PHƯƠNG TIỆN GIAO THÔNG ĐƯỜNG BỘ - HỆ THỐNG PHANH CỦA MÔ TÔ, XE MÁY - YÊU CẦU VÀ PHƯƠNG PHÁP THỬ TRONG CÔNG NHẬN KIỂU

Road vehicles - Braking device of motor cycles and mopeds - Requirements and test methords in type approval

Loại L:

Phương tiện cơ giới đường bộ có ít hơn bốn bánh.

E.1 Loại L1:

Xe hai bánh có dung tích xi lanh động cơ không quá 50 cm^3 đối với động cơ nhiệt và vận tốc thiết kế lớn nhất không quá 50 km/h.

E.2 Loại L2:

Xe ba bánh được bố trí tuỳ ý, có dung tích xi lanh động cơ không quá 50 cm^3 đối với động cơ nhiệt và vận tốc thiết kế lớn nhất không quá 50 km/h.

E.3 Loại L3:

Xe hai bánh có dung tích xi lanh động cơ lớn hơn 50 cm^3 đối với động cơ nhiệt hoặc vận tốc thiết kế lớn nhất hơn 50 km/h.

E.4 Loại L4:

Xe ba bánh được bố trí không đối xứng theo mặt phẳng trung tuyến dọc của xe, có dung tích xi lanh động cơ lớn hơn 50 cm^3 đối với động cơ nhiệt hoặc vận tốc thiết kế lớn nhất hơn 50 km/h (xe có thùng bên).

E.5 Loại L5:

Xe ba bánh được bố trí đối xứng theo mặt phẳng trung tuyến dọc của xe, có dung tích xi lanh động cơ lớn hơn 50 cm^3 đối với động cơ nhiệt hoặc vận tốc thiết kế lớn nhất hơn 50 km/h.

Xe trong thí nghiệm sử dụng là xe Future dung tích 125 cm^3 nên tiêu chuẩn đối chiếu là loại L3.

C.2.2.2 Xe được thử đầy tải hoặc không tải.

C.2.2.2.1 Phanh xe chỉ với hệ thống phanh liên hợp.

Bảng C-3

Loại	Khoảng cách dừng, S m	Gia tốc phanh khai triển đầy đủ trung bình tương ứng m/s^2
L1, L2	$S \le 0,1V + V^2/115$	4,4
L3	$S \le 0,1V + V^2/132$	5,1
L4	$S \le 0,1V + V^2/140$	5,4
L5	$S \le 0,1V + V^2/130$	5,0

C.2.2.2.2 Phanh xe với hệ thống phanh chính thứ hai hoặc hệ thống phanh dự phòng (khẩn cấp), đối với tất cả các loại xe khoảng cách dừng phải:

$S \le 0,1V + V^2/65$ (gia tốc phanh khai triển đầy đủ trung bình tương ứng là 2,5 m/s^2)

Nhận xét:

Qua quá trình tiến hành khảo sát thí nghiệm kiểm tra gia tốc phanh ta thấy gia tốc phanh cực đại của xe trong thí nghiệm là 6,867 m/s^2 > 5,4 m/s^2 => xe đảm bảo điều kiện vận hành theo TCVN 6824:2001.

MỤC LỤC

PHẦN A: TÍNH TOÁN CÁC THÔNG SỐ CỦA 6 BÀI THÍ NGHIỆM..... 1
I.Tính hệ số cản lăn theo phương pháp chạy theo quán tính 1
II.Tính hệ số cản không khí theo phương pháp cho xe máy xuống dốc dưới tác dụng của lực trọng trường: 1
III. Tính hệ số bám bằng phương pháp phanh: 2
IV. Đo tiêu hao nhiên liệu bằng phương pháp cân trực tiếp: 3
V. Tính quãng đường phanh: 3
VI. Tốc độ cực đại ở tay số 1: 3
PHẦN B: ĐỒ ÁN KHẢO SÁT GIA TỐC PHANH 4
I. CÁC PHẦN CỨNG TIẾN HÀNH THÍ NGHIỆM 5
1. ATMEGA16: 5
2. MẠCH CHUYỂN USB UART TTL FT232RL: 7
3. CẢM BIẾN GIA TỐC GY-61 ADXL335: 8
4. CÁC THIẾT BỊ KHÁC: 9
II. PHẦN MỀM LẬP TRÌNH CHO THÍ NGHIỆM 10
1. CODEVISION AVR 2.05: 10
2. CHƯƠNG TRÌNH NẠP PROGISP: 16
3. PHẦN MỀM MÔ PHỎNG PROTEUS: 20
4. PHẦN MỀM TEST CỔNG COM (HERCULES) : 22
5. PHẦN MỀM LABVIEW: 24
III. TIẾN HÀNH THÍ NGHIỆM 28
1. BỐ TRÍ THÍ NGHIỆM: 28
2. THỰC HIỆN THÍ NGHIỆM 29
3. KẾT QUẢ THÍ NGHIỆM: 29