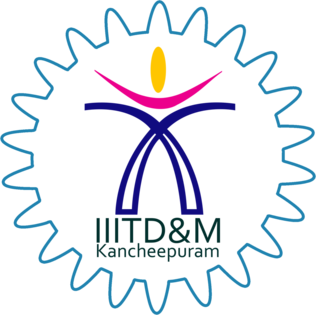
**Embedded Systems**

**Indian Institute of Information Technology Design and Manufacturing Kancheepuram**

****

**Project**

**Calculate the speed and distance covered by a wheel using wheel encoder sensor and LCD display**

**Submitted by :**

Arindam Sharma-CED17I022

Sreepathy Jayanand-CED17I038

Arjun Ramesh-COE17B032

Rahul James -CED17i050

Aditya Gopakumar-CED17I036

**Problem statement**:

Calculate the speed and distance covered by a wheel using wheel encoder sensor and LCD display.

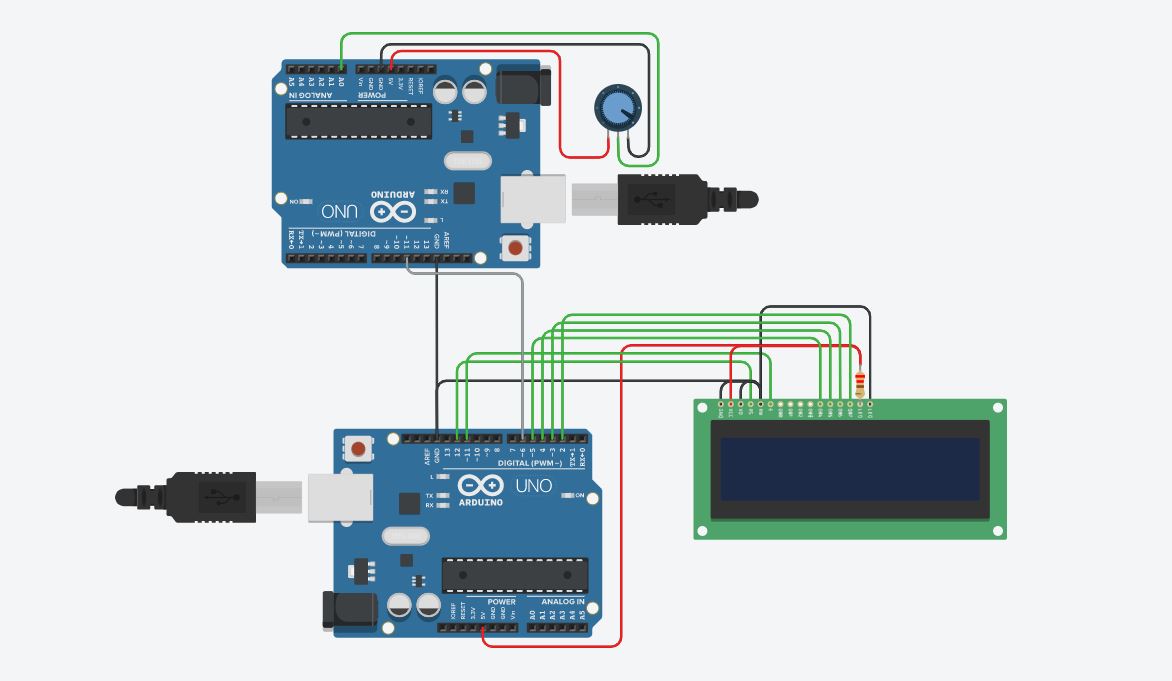
In the simulation, due to lack of resources we are using another Arduino for sending the input what the encoder would send.

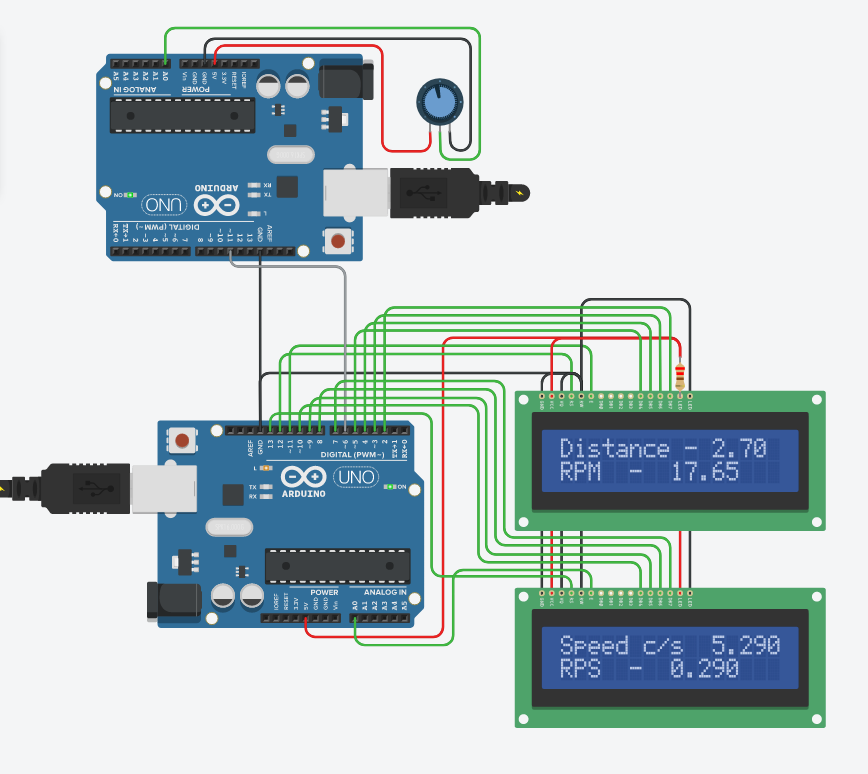
**Part 1 : Arduino**

**Components required:**

1. Arduino Uno R3
2. 16 x 2 lcd display
3. Battery
4. Jumper wires
5. Rotary encoder ( / Arduino for Simulating encoder)

**Diagram:**





**Working Principle:**

When the rotary encoder rotates it sends pulses to the Arduino. Arduino identifies the rotation and direction(clockwise/anticlockwise) of the wheel. Arduino counts the pulses and computes the speed and distance using mathematical formulas(calibrated) and displays to the lcd display.

In the rotary encoder conductors are placed in a disc and connected to a common pin in a circular fashion. Encoders have 2 output pins and when it rotates it gives zero and one. The pattern of zeros and ones gives us information about the speed, distance and direction of the rotation.

**Calibration:**

The rotary encoder gives information about the rotations but we have to use the below mentioned mathematical formulas to calculate the distances.



Distance travelled in one rotation = 2 x π x r

Distance travelled in one step = 2 x π x r / N,

where N is the number of spikes in the rotary encoder.

Since we don’t have the wheel encoder we have decided to go with an arduino to generate the signals which the wheel encoder would produce.

**Code:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Code** | **Explanation** |
| **1** | **#include <LiquidCrystal.h>**  **LiquidCrystal lcd(12, 11, 5, 4, 3, 2);**  **LiquidCrystal lcd1(13, A0, 10, 9, 8, 7);**  **const int encoder= 6;**  **const int hole=20;**  **const int pi =3.14;**  **const int wheel\_radius = 3; //in cm**  **float dist;** | Include LiquidCrystal header  Declare the 2 lcd displays  Encoder-value from Arduino1  Hole - number of holes in wheel encoder  Declare wheel radius and pi  Dist - Distance travelled |
| **2** | **float Distance(int var){**  **float cir=2\*pi\*wheel\_radius; //in cm**  **dist=dist+(RPS(var)\*cir\*var)/1000;**    **Serial.print(var);**  **Serial.print(",");**  **Serial.println(dist);**      **return dist;**  **}** | Update distance using the formula  Circumference = 2πr  Distance incremented with  ∂ distance = rotations\_per\_second \* circumference \* time  Print values |
| **3** | **float RPS(int var){**  **float rps=1000/float(hole\*var);**  **return rps;**  **}** | Formula for rotations per second: (time to 1 hole to pass) \* (total holes) \*(10e-3)  To convert ms to sec |
| **4** | **float RPM(int var){**  **return RPS(var)\*60;**  **}** | Rotations per minute = rotations per second \* 60 |
| **5** | **float Speed(int var){**  **float cir=2\*pi\*wheel\_radius; //in cm**  **return RPS(var)\*cir;**  **}** | Speed in cm/s= circumference \* rotations per second |
| **6** | **void setup() {**  **lcd.begin(16, 2);**  **lcd1.begin(16, 2);**  **Serial.begin(9600);**    **lcd.print("Distance - ");**  **lcd1.print("Speed c/s - ");**    **lcd.setCursor(0,1);**  **lcd1.setCursor(0,1);**    **lcd.print("RPM - ");**  **lcd1.print("RPS - ");**  **pinMode(encoder,INPUT);**    **dist=0;**  **}** | Initialize the 2 lcds  Print the distance in lcd 1 row 1 and speed in lcd2 row 1  Print rpm in lcd 1 row 2 and rps in lcd 2 row 2  Set pin mode to input for encoder so that it receives signal from arduino 1  Initialize distance to 0 |
|  | **void loop() {** |  |
|  | **int temp = digitalRead(encoder);**  **float diff=0;**  **if(temp==0){**  **int curr\_time=millis();**  **while(digitalRead(encoder)==0);**  **while(digitalRead(encoder)==1);**  **diff=millis()-curr\_time;**  **}**    **if(diff==0){diff=1;}** | Read the data coming from the encoder pin  Find current time in milliseconds using millis()  Wait until we change from 0 to 1 to 0, i.e 1 cycle.  Diff = 1 cycle time = current system time - time measured earlier. |
|  | **// LCD output**  **lcd.setCursor(11, 0);**  **lcd1.setCursor(11, 0);**  **lcd.print(Distance(diff));lcd.print(" ");**  **lcd1.print(Speed(diff));lcd.print(" ");**  **lcd.setCursor(8,1);**  **lcd1.setCursor(8,1);**  **lcd.print(RPM(diff));lcd.print(" ");**  **lcd1.print(RPS(diff));lcd.print(" ");** | Set the display cursor to 11 th row and 0th column  Print speed  Set display cursor to 8th row and 1st column in both lcd  Display the calculated rpm and rps |
|  | **}** |  |

For the simulation we are using 2 arduino boards. One for giving simulated values to the other Arduino acting as if it was a rotating wheel with an encoder embedded in it. The second Ardiono would receive values from the first and compute the distance travelled, rotations per second and speed and display it in the led.

For the LCD display we have used the ***liquidcrystal.h*** header file

The ***main loop()*** function runs again and again. Inside that we are computing the current time in milliseconds using ***millis()*** function and storing it in ***current\_time***.

Temp is the value that is received from the first arduino as the encoder output using the ***digitalRead(encoder)*** function . The encoder variable comes from the first arduino.

**Calculation of the time difference between successive pulses:**

This is done using the if(temp == 0) condition check. For every consecutive pulse we enter this if block.

We compute the time difference using diff = current\_time - prev\_time

Difference of time = Current time - previous recorded time.

This would update the time difference after each pulse received. Then we update the prev\_time, using the formula prev\_time = current\_time.

After updating the diff variable, we compute rotations per second.

We know the time it takes for 1 pulse. 20 pulses make 1 rotation.

Time for 1 pulse = diff

Number of rotations per second = 10e3 / (20 \* diff)

This is stored in the rps variable.

The speed would be modified using the formula:

Speed = rps \* 2 \* π \* r cm/s,

Where r is radius of wheel in cm

The distance would be modified using the formula:

distance = distance + diff \* rps \* 2 \* π \* r cm

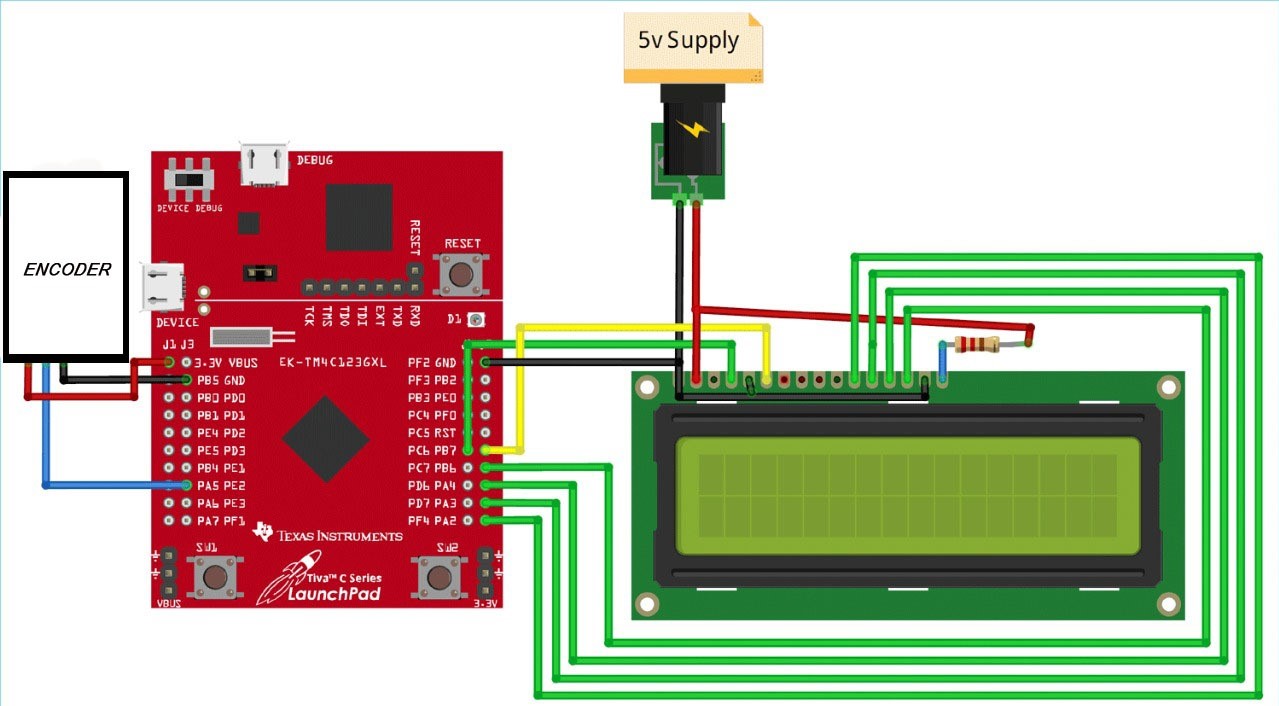
Then we print the computed values.

**Part 2: Tivac**

**Components required:**

1. Tivac microcontroller (TM4C123GH6PM)
2. 16 x 2 LCD display
3. Battery
4. Jumper wire
5. Rotary encoder

**Diagram:**



Working principle:When the rotary encoder rotates it sends pulses to the Tivac. The Tivac identifies the rotation and direction(clockwise/anticlockwise) of the wheel. The Tivac counts the pulses and computes the speed and distance using mathematical formulas(calibrated) and displays to the LCD display.

In the rotary encoder conductors are placed in a disc and connected to a common pin in a circular fashion. Encoders have 2 output pins and when it rotates it gives zero and one. The pattern of zeros and ones gives us information about the speed, distance and direction of the rotation.

**Calibration:**

The rotary encoder gives information about the rotations but we have to use the below mentioned mathematical formulas to calculate the distances.

Distance travelled in one rotation = 2 x π x r

Distance travelled in one step = 2 x π x r / N,

where N is the number of spikes in the rotary encoder.

**Code:** [[tivac code]](https://drive.google.com/file/d/1D7LqI33oGzmjR_R17kVwJKyIrBw6UUN1/view?usp=sharing)

For the LCD display we have used the liquidcrystal.h header file

Initialize the registers.

For the simulation we are using 1 Tivac board and 1 wheel encoder. The encoder would send pulses to the tivac board and based on the pattern of the pulses sent by the encoder the Tivac would compute the distance, speed, rotation per second using mathematical formulas and display in the 16 x 2 LCD display.

For the LCD display we have used the liquidcrystal.h header file

Temp is the value that is received from the encoder using the digitalRead(encoder) function .

**Calculation of the time difference between successive pulses:**

This is done using the if(temp == 0) condition check. For every consecutive pulse we enter this if block.

We compute the time difference using diff = current\_time - prev\_time

Difference of time = Current time - previous recorded time.

This would update the time difference after each pulse received. Then we update the prev\_time, using the formula prev\_time = current\_time.

After updating the diff variable, we compute rotations per second.

We know the time it takes for 1 pulse. 20 pulses make 1 rotation.

Time for 1 pulse = diff

Number of rotations per second = 10e3 / (20 \* diff)

This is stored in the rps variable.

The speed would be modified using the formula:

Speed = rps \* 2 \* π \* r cm/s,

Where r is radius of wheel in cm

The distance would be modified using the formula:

distance = distance + diff \* rps \* 2 \* π \* r cm

Then we print the computed values.

**Reference :-** [**[arduino code]**](https://drive.google.com/file/d/1migxHzNkH4qHBHTAfhE3MS8teeE-NPo2/view?usp=sharing)**,** [**[video]**](https://drive.google.com/file/d/1v5MaeQkBLtBFdSW-PYwKNsOOXwKQ3_X0/view?usp=sharing)**,** [**[tivac code]**](https://drive.google.com/file/d/1D7LqI33oGzmjR_R17kVwJKyIrBw6UUN1/view?usp=sharing)