# Digital RPM Meter using Arduino

SudiptoDebnath (12.01.05.018),KaziBozleRabby (12.01.05.026),
Mohaimenul Islam (12.01.05.033),
SusmitaSaha (12.01.05.034), TarinSubha (12.01.05.032).
Department of Electrical & Electronic Engineering,
Ahsanullah University of Science & Technology, Dhaka, Bangladesh
E-mail: S.dnath91@gmail.com

**Abstract:** A Digital RPM Meter is a measuring instrument which can measure the rotational speed of a rotary machine digitally. The industrial name of this instrument is "Tachometer". It is an important measuring device in the field of electrical engineering & widely used in industries and laboratorial work.

**Keywords:**Arduino, IR- pair, lcd, attachinterrupt, op-amp.

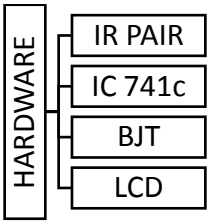
## 1. Introduction

In order to constructing a simple digital RPM meter, we need to work in three different fields. They are-

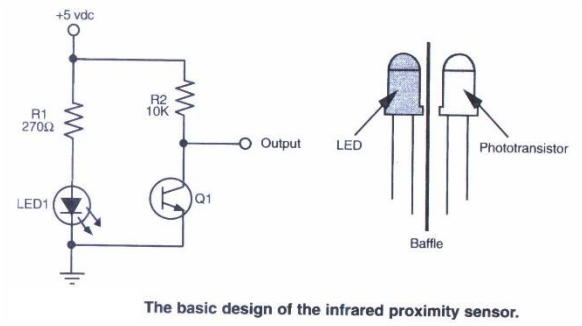

In hardware section we design a comparator circuit to take the inputs using arduino& through the software section we control the arduino, calculations and outputs.

## 2. Hardware

In Hardware section we thoroughly discuss about the comparator circuit and the following parts.




**IR-PAIR:** An IR-Pair is consisted of a infrared LED & a phototransistor. Here infrared LED works as transmitter & phototransistor works as a receiver. When this pair is accordingly biased with 5V dc infrared LED begin to transmit the infrared beam. If the beam is received through the phototransistor the output voltage across it is increased.




The basic design of the infrared proximity sensor.

Fig.1

**IC 741c:**An IC 741c is single op-amp device. An operational amplifier (op-amp) is a DC-coupled high-gain electronic voltage amplifier with a differential input and, usually, a single-ended output. In this configuration, an op-amp produces an output potential (relative to circuit ground) that is typically hundreds of thousands of times larger than the potential difference between its input terminals.

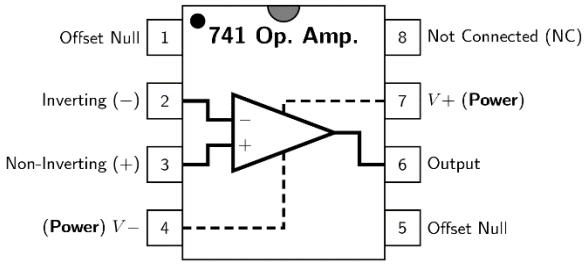



Fig.2: Schematic Diagram of 741c.

Here in our circuit we use this device to convert the analog voltage signal into digital voltage signal which comes from the output of IR-pair.

**BJT:**A bipolar junction transistor (BJT or bipolar transistor) is a type of transistor that relies on the contact of two types of semiconductor for its operation. BJTs can be used as amplifiers, switches, or in oscillators. BJTs can be found either as individual discrete components, or in large numbers as parts of integrated circuits.

We use BD135 transistor to improve the output signal of op-amp.

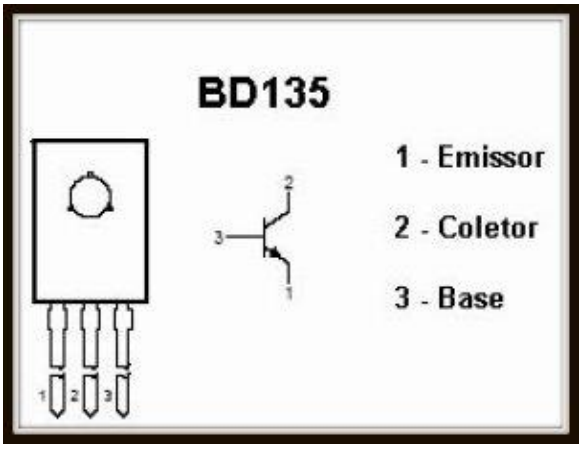



Fig.3 : BD 135

**LCD:**The LCDs have a parallel interface, meaning that the microcontroller has to manipulate several interface pins at once to control the display. The interface consists of the following pins:

A register select (RS) pin that controls where in the LCD's memory you're writing data to. You can select either the data register, which holds what goes on the screen, or an instruction register, which is where the LCD's controller looks for instructions on what to do next.

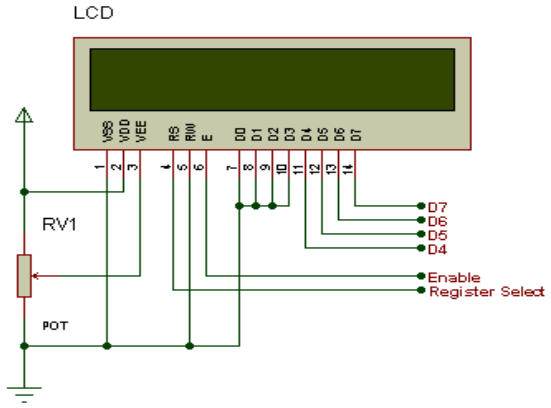



Fig.4(a): Schematic of LCD

A Read/Write (R/W) pin that selects reading mode or writing mode.An Enable pin that enables writing to the registers.8 data pins (D0 -D7). The states of these pins (high or low) are the bits that you're writing to a register when you write, or the values you're reading when you read.

There's also a display constrast pin (Vo), power supply pins (+5V and Gnd) and LED Backlight (Bklt+ and BKlt-) pins that you can use to power the LCD, control the display contrast, and turn on and off the LED backlight, respectively.

| Pin | Symbols and functions |
|---|---|
| 1 | GND |
| 2 | VCC (+5v) |
| 3 | Contrast adjust |
| 4 | ( RS ) ==>> 0 = Instruction input / 1 = Data input |
| 5 | ( R/W ) ==>> 0 = Write to LCD Module / 1 = Read from LCD module |
| 6 | ( E ) ==>> Enable signal |
| 7 | ( DB0 ) ==>> Data Pin 0 |
| 8 | ( DB1 ) ==>> Data Pin 1 |
| 9 | ( DB2 ) ==>> Data Pin 2 |
| 10 | ( DB3 ) ==>> Data Pin 3 |
| 11 | ( DB4 ) ==>> Data Pin 4 |
| 12 | ( DB5 ) ==>> Data Pin 5 |
| 13 | ( DB6 ) ==>> Data Pin 6 |
| 14 | ( DB7 ) ==>> Data Pin 7 |
| 15 | ( VB+ ) ==>> back light (+5V) |
| 16 | ( VB- ) ==>> back light (GND) |

Fig.4(b): Function of LCD's pins.

## 3. Arduino

To take the outputs from comparator circuit & compute the data for calculation and to control the LCD, here we use a microcontroller atmega 238p which comes with aarduino which is an open-source electronics prototyping platform.Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software running on a computer (e.g. Flash, Processing, MaxMSP).We use arduinouno to make our task easy. The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started."Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0.

**Summary:**

| | |
|---|---|
| Microcontroller: | ATmega328 |
| Operating Voltage: | 5V |
| Input Voltage (recommended): | 7-12V |
| Input Voltage (limits): | 6-20V |
| Digital I/O Pins: | 14 (of which 6 provide PWM output) |
| Analog Input Pins: | 6 |
| DC Current per I/O Pin | 40 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 32 KB (ATmega328) of which 0.5 KB used by bootloader |
| SRAM | 2 KB (ATmega328) |
| EEPROM | 1 KB (ATmega328) |
| Clock Speed | 16 MHz |

**Power:** The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.The power pins are as follows:**VIN**: The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

**5V:**This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.

**3V:**A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

**GND:** Ground pins.

**IOREF**: This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

**Memory:** The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

**Input and Output:**Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:


Fig.5 : Arduino Uno Schematic

**Serial:** 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

**External Interrupts:** 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.

**PWM:** 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.

**SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.

**LED:** 13. There is a built-in LED connected to digital **pin 13:** When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the

analogReference() function. Additionally, some pins have specialized functionality:

**TWI:** A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library. There are a couple of other pins on the board:

**AREF:** Reference voltage for the analog inputs. Used with analogReference().

**Reset:**Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

## 4. SOFTWARE

A program needs to link between arduino and the hardware. This program is written on C++ language at arduino IDE platform. The program is given below:

```
#include<LiquidCrystal.h>
LiquidCrystal lcd(12,11,10,9,8,7,6,5,4,3);
int rpm;
int rpml=0;
volatile int count =0;
void setup() {
  // put your setup code here, to run once:
  lcd.begin(16,2);
  lcd.setCursor(2,0);//(column,row)
  lcd.print("Current RPM");
  attachInterrupt(0,techo,RISING);
}

void loop() {
  // put your main code here, to run repeatedly
if(millis()/1000<=6)rpml=count;
rpm=rpml*10;
 lcd.setCursor(5,1);
  lcd.print(rpm);
}
void techo()
{
  count++;
}
```

Here we discussed the built in function that were used to complete the program:

**begin():** Initializes the interface to the LCD screen, and specifies the dimensions (width and height) of the display. begin() needs to be called before any other LCD library commands.Syntax: lcd.begin(cols, rows)**.** Parameters- lcd: a variable of type LiquidCrystal.cols: the number of columns that the display hasrows: the number of rows that the display has.

**setCursor():** Position the LCD cursor; that is, set the location at which subsequent text written to the LCD will be displayed. Syntax: lcd.setCursor(col, row). Parameters-lcd: a variable of type LiquidCrystal. col: the column at which to position the cursor (with 0 being the first column).row: the row at which to position the cursor (with 0 being the first row).

**print():** Prints text to the LCD.Syntax: lcd.print(data).Parameters-lcd: a variable of type LiquidCrystal.data: the data to print (char, byte, int, long, or string).

**attachInterrupt():** Specifies a named Interrupt Service Routine (ISR) to call when an interrupt occurs. Replaces any previous function that was attached to the interrupt. Most Arduino boards have two external interrupts: numbers 0 (on digital pin 2) and 1 (on digital pin 3).Interrupts are useful for making things happen automatically in microcontroller programs, and can help solve timing problems. Good tasks for using an interrupt may include reading a rotary encoder, or monitoring user input. Syntax: attachInterrupt(interrupt, ISR, mode). Parameters-interrupt: the number of the interrupt (int). ISR: the ISR to call when the interrupt occurs; this function must take no parameters and return nothing. This function is sometimes referred to as an interrupt service routine. mode: defines when the interrupt should be triggered. Four contstantsare predefined as valid values:LOW to trigger the interrupt whenever the pin is low. CHANGE to trigger the interrupt whenever the pin changes value. RISING to trigger when the pin goes from low to high. FALLING for when the pin goes from high to low.

## 5. Troubleshooting

**Circuit Diagram:**

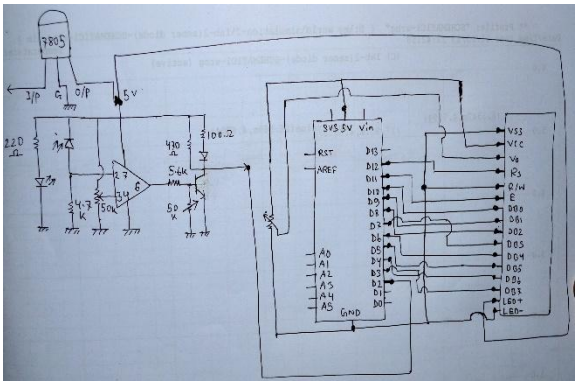


Fig. 6: The project Diagram

Here we use IC 7805 for voltage regulation so that we can get 5V constant dc supply from its output. Then we setup the comparator circuit. When the IR-pair is disconnected we set the voltage of pin 3 of IC 741 at 0.34V & voltage of pin 2 is 0.15V. As a result we get high output voltage at pin 6 about 4.23V. Here we use reflection method to link between Infrared LED & phototransistor. When this pair is connected the pin 2 voltage goes high about 4.6V. As a result we get the lower output voltage about 1.13V. To improve these outputs we use a BJT BD135. We connect a resistor valued 470 ohm with the collector to pull-up the output voltage 4.23V to 5V. We also connect a led parallel with that resistor to indicate when the link is created between the infrared led & phototransistor. The output from the collector is then connected to the digital pin 2 of arduino. To control the lcd we use the following connection:

| Arduino Pin out | LCD Pin out |
|---|---|
| 12 | RS |
| 11 | E |
| 10 | DB 0 |
| 09 | DB 1 |
| 08 | DB 2 |
| 07 | DB 3 |
| 06 | DB 4 |
| 05 | DB 5 |
| 04 | DB 6 |
| 03 | DB 7 |

The rest pins are connected as following:

| LCD pin out | Connected to |
|---|---|
| VSS | Ground |
| R/W | Ground |
| Vcc | 5V |
| $V_o$ | Mid- point of a 50 K trimmer |
| LED + | 5V |
| LED- | Ground |

## 6. Development

We use our rpm meter to control the speed of a low volt DC motor. For this we only need to develop our coding section not the hardware section.

## 7. Modification

Instead of reflection of beam method cutting of beam method can be applied to measure the speed of a fan.

Reflecting distance can be increased by varying the voltage of pin 3 of IC 741.

By change the coding we can make a people counter.

## 8. Future Development

- People counter.
- Controlling the switching system of a room.

## 9. Costing &Equipments

| Name | Specification | Quantity | Price(BDT) |
|---|---|---|---|
| IR- pair | - | 1 pair | 10/- |
| Voltage Regulator | 7805 | 1 pc | 20/- |
| Resistance | 100Ω, 220Ω, 470Ω, 4.7K Ω, 5.6K Ω. | 1 pc | 5/- |
| Trimmer | 50K Ω | 3 pc | 9/- |
| LED | - | 1 pc | 3/- |
| Op-amp | IC 741 | 1 pc | 15/- |
| BJT | BD 135 | 1 pc | 15/- |
| LCD | 16*2 | 1 pc | 170/- |
| Arduino | Uno R3 | 1 pc | 1200/- |
| Conneting wire | - | - | 20/- |
| Bread Board | - | 1 pc | 180/- |
| **TOTAL** | | = | 1624/- |

## 10. Limitations

- A good reflector is needed.
- Reflection distance should be minimum for good accuracy.
- Direct sunlight may hamper the accuracy.

## 11. Reference


- Coughling& Driscoll, Operational amplifier & Linear integrated Circuits, 6th edition, Prentice- Hall India.
- Arduino.cc (http://arduino.cc/)
- ZulkarnimeSifat, ex-student of AUST, Project Engineer, Bangladesh Army.
- Jeremy Blum, Video lectures, (http://www.jeremyblum.com/category/arduino-tutorials/).