

Contactless digital tachometer using microcontroller

R. Palanisamy, S. Vidyasagar, V. Kalyanasundaram, R. Sridhar

Department of Electrical and Electronics Engineering, SRM Institute of Science and Technology, India

Article Info

Article history:

Received Nov 28, 2019

Revised Jun 17, 2020

Accepted Jun 29, 2020

Keywords:

Digital tachometer

Memory unit

Microcontroller

Transmitter and receiver

ABSTRACT

Tachometer is a device that used for counting or for the measuring purpose of the number of revolutions (that is the total number rotations made by the device in unit of measuring time) of an object in unit time. It is expressed in the unit of RPS or RPM, the model uses a set of infrared transducer receiver to count the RPM pulses, and the Arduino microcontroller is used for the implementation of the project. The individual pulses are counted by the microcontroller to give the final output of the RPM.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

S. Vidyasagar,

Department of Electrical and Electronics Engineering,

SRM Institute of Science and Technology,

Kattankulathur, Chennai, India.

Email: vidyasas@srmist.edu.in

1. INTRODUCTION

The basic operating principle of a mechanical tachometer is a movable magnet driven by the rotating input shaft that interacts closely with an eddy current sensor that generates the reading [1-5]. The rotating magnet within the sensor forces the indicator needle to show the proportional reading on the given output method as per the engine speed, whereas a countering spring acts against the sensors force to bring it to a stable stopping point (output reading on the dial) [6]. As compared to a digital tachometer, the traditional kind faces many issues like the probability of mechanical failure, wear of parts with age that might give errors [7-11].

- The basic characteristics of an 8051 microcontroller are as mentioned, Flash Memory=8 kB, RAM=256 Bytes, Clock speed=12 Mhz, one operating cycle=12 clock cycles, 8051 operating voltage is generally between 5 V to 6.6 V and minimum current required is 25 m Amps [12-14].
- Single instruction executed in 2 machine cycles General IDE and burner can used to write and compile code in assembly language for 8051. Programmer needs to know about the registers and pins and their specific operations by compiler along with the use of specific SFR,s and interrupt routines to write code for 8051 microcontroller. To code a 8051 controller we need a burner also known as an external programmer as it does not directly interact directly with a modern computer [15-17].
- It is tough to program in Assembly language.
- No EEPROM.
- Arduino boards need a minimum of 5 V to operate and work upto a rage of 20V. It can b directly powered by the device being used to code it testing as it is connected using a basic USB cable that is easily available. They can also be powered using independent sources using an external adapter [18].

- Arduino is present in logic 5 V and 3.3 V. That lets us use arduino more adaptably as compared to an 8051 and can interfaced with a 5 V logic input device and lower voltage sensors with minimum requirements that can be taken directly from the arduino board. That eliminates the need of a level converter [19-21].
- Minimum required powering current is 25mAmps that is like 8051.
- Its GPIO pins be used alongside 40mAmps of current which is very advantageous when compared to an 8051's on board gpio [22].
- Arduino uno Flash Memory is fixed at 32 kB, Arduino uno's RAM is pre-set to 2 kB, Arduino uno EEPROM at 1 kB, clock speed at 16 Mhz that can be increase upto 20 Mhz, here we consider it as: one operating cycle = 1 operating clocks cycle.
- An instruction run for a single operation cycle [23].
- It is easier to code in c programming language.
- PIC needs more than a single clock cycle for every instruction. AVRC programming lets us execute most of the instructions in a single clock cycle [24].
- PIC has a fixed memory stack that has very limited space, so it limits us from stacking them very deeply and hence subroutines are limited to a certain extent and C compiler cannot make big stack frames. Stack pointers are used to address all registers of RAM that are available in the given memory space [25, 26].
- PIC microcontroller can be used to directly address 256 bytes of memory that has to be used by changing the bank after every instruction cycle using extra instructions to change. AVR has an advantage over it as it can directly access 64k of the memory.
- Crippleware is one of the few free pic compilers. AVR free compiler being developed and updated by a larger team [27, 28].
- PIC has a 'W' register whereas AVR has 32 registers that can be used for general purpose as well and includes multiple pairs which can be used as pointers.

2. PROPOSED SYSTEM CONCEPT

The basic concept includes an Arduino with the background of c programming from embedded systems that employs an IR sensor, an Arduino microcontroller, an LCD output screen and a source from generating the output to be measured, which is shown in Figure 1. The IR sensor module involves an IR transmitter LED and an IR photoreceiver, the microcontroller used is loaded with a C programmed code and is stored in its memory until it is reset, and the actuator is an LCD display device, displaying the real time speed being measured from the motor surface.

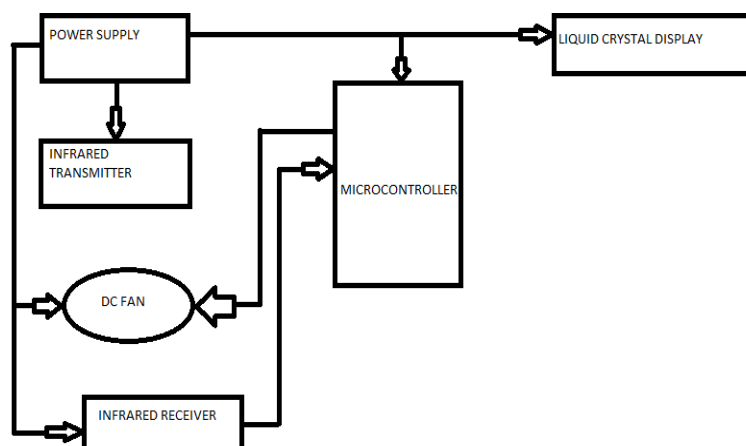


Figure 1. Block diagram of the proposed system

The IR sensor detects the high pulses caused by reflection from the motors surface and the controller is used to calculate the speed of the motor without actually being connected to it, by the principle of reflection a result is generated from the signals, which is shown in Figure 2. This calculated output signal is sent as an electrical signal and fed into the actuator that is the LCD panel, which is programmed to show the speed in terms revolutions in one minute. This is the final output.

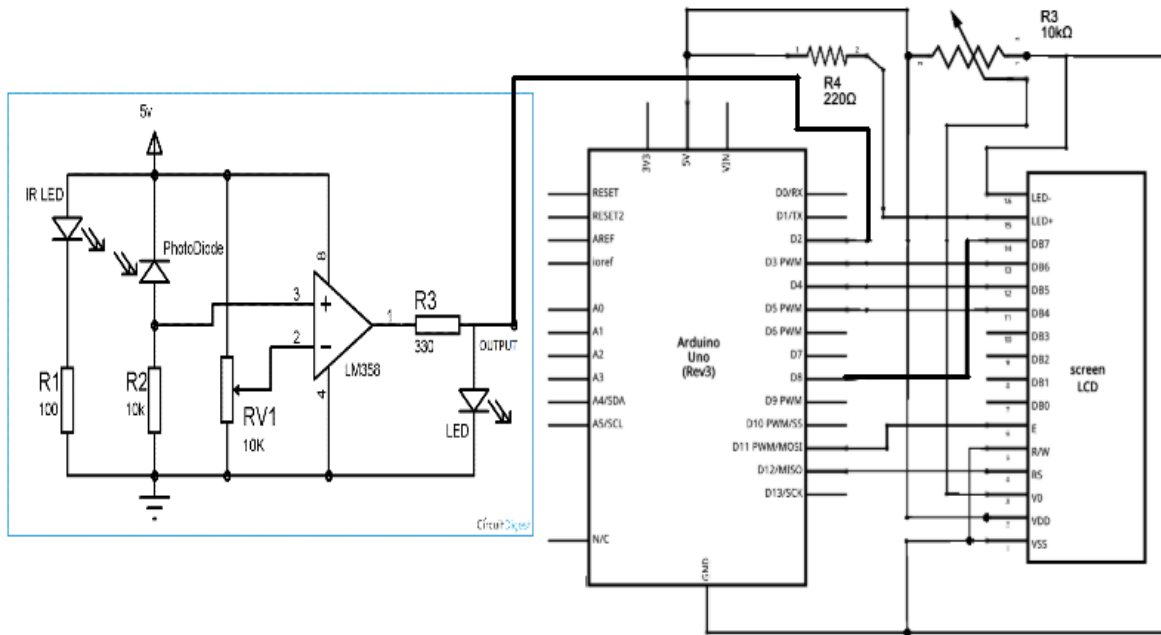


Figure 2. Control using Arduino microcontroller

3. CONTROL STRATEGIES

3.1. Arduino

The Arduino Uno is a logic board based on the open source hardware provided by multiple manufacturers. It has an Atmega328 microcontroller embedded onto the board itself. The Arduino uno board in particular is part of the compact Arduino series that contains the Arduino nano and the Arduino mini. The Arduino board has a total of 28 pins, which is shown in Figure 3.

Arduino Pin Mapping

www.arduino.cc

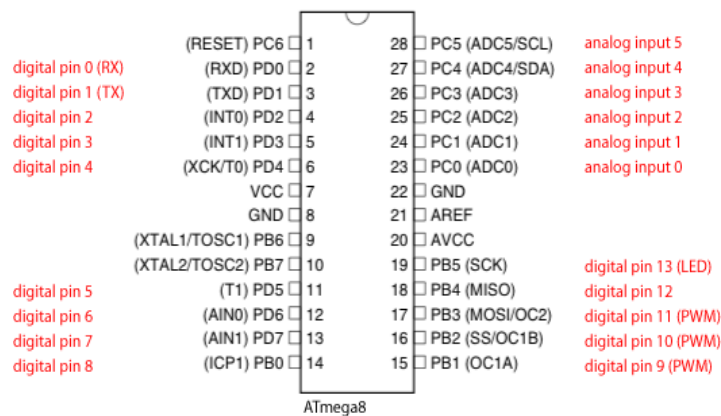


Figure 3. Pin diagram of Arduino microcontroller

The pins are of the following types.

- 13 data Pins
 - 7 Digital PWM Pins
 - 6 General purpose data Pins
- 6 Analog input pins
- 2 Ground Pins

- 5V output Pin
- 3.3V output Pin
- Vin Pin
- Reset Pin
- IOREF Pin

It has 3 zones of onboard memory

- SRAM-32 KB
- EEPROM-1 KB
- Flash Memory -32KB (31.5 Usable)

The clock speed of an Arduino Uno board is 16 MHz. It can be connected to a computer using the UART TTL serial communication link or via a USB port. The Arduino board can either draw power via USB or via an external supply which can be connected via the barrel plug connection provided on the board. It can take an input of 7V to 20V. The recommended operating voltage should be 5V. If there is power being drawn from the 5V output pin on the board, an input of 7V or greater must be provided to the board. However, the maximum permissible voltage for sustained use is 12V otherwise the voltage regulators overheat. There are two basic types of Pin functions:

- General Pin functions
 - LED Pin: Pin 13 has a built in LED pin which is edge triggered and so in on when the pin is HIGH
 - Vin Pin: It can take input from an external source if it is not being powered by the USB port or via the Barrel plug
 - IOREF Pin: This pin on the Arduino board is used to select a voltage reference based on which an appropriate power source is selected.
 - Reset Pin: It is generally used to add a reset button to the circuit
- Specialized Pin functions
 - RX Pin/ Pin 0: Used to receive serial data
 - TX Pin/Pin 1: Used to transmit serial data
 - Pins 2 and 3: Used to create hardware interrupts
 - PWM Pins: pins 3,5,6,9 and 20 can be used for 8 bit pulse width modulation. We utilise the `analogWrite()` function to achieve that result
 - SPI Pins: Pins 10, 11, 12 and 13 can be used for Serial Peripheral Interface. They use the SPI library to communicate with the Arduino IDE.
 - TWI Pins: A4 and A5 pins also known as the SDA and SCL pins respectively are used for two wire interface. This is enabled with the Wire library
 - AREF: This provided an analogue reference voltage to the input.

Each pin other than the 3.3 V pin can take a maximum of 20 mA of DC current. The 3.3 V pin can take up to a 50mA DC input current.

3.2. CODE

```
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 8;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
int pwm=9;
int pot=A0;
float value=0;
int percent;
float rev=0;
int rpm;
int oldtime=0;
int time;
void isr() //interrupt service routine
{
  rev++;
}
void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  attachInterrupt(2,isr,RISING);
}
```

```

void loop() {
  delay(1000);
  detach Interrupt(0); //detaches the interrupt
  time=millis() oldtime; //finds the time
  rpm=(rev/time)*60000; //calculates rpm
  oldtime=millis(); //saves the current time
  rev=0;
  value=analog Read(pot); //reads the speed control POT
  value=value/4;
  analog Write(pwm,value); //sets the desired speed
  percent=(value/255)*100; //finds the duty cycle %
  lcd clear();
  lcd.setCursor(0,0);
  lcd.pnt("___TACHOMETER___");
  lcd.Cursor(0,1);
  lcd print(rpm);
  lcd print(" RPM");
  attach Interrupt(0,isr,RISING);
}

```

4. CONTROLLING THE SPEED OF MOTOR

For regulating the speed of the motor, a potentiometer is added to the circuit. For driving the motor, transistor Q1 is used. Through the limiting resistance R1, the base is connected to PWM pin 9 of the arduino. Analog pin A0 of the arduino is connected to the wiper of the speed control pot R4. Using the analog read function, the voltage at this pin is converted into a value between 0 and 1023. In order to fit the value between 0 to 255, this value is divided by 4. Using the analog Write function, the value is written to the PWM pin 9. Result is seen at pin 9. Its duty cycle is proportional to the value written using analog write function. For example, if the value is 127, the duty cycle will be 50% and if the value is 255, the duty cycle will be 100%. C1 is a noise by pass capacitor and D1 is a freewheeling diode. Using the standard liquid crystal library, the RPM and duty cycle are displayed.

Contact less tachometer is a device used to measure speed of a motor by counting the number of rotations per second of a rotating shaft using micro controller. As the name indicates, it can very accurately measure the rotational speed of a shaft without even touching it. This is interesting because making direct contact with the shaft is not an option, and will reduce the speed of the shaft giving faulty readings.

5. EXPERIMENTAL RESULT

This can also be employed to measure the speed of motors which are at unreachable places. This device is built on a microcontroller – Atmega328, speed is detected using the IR transmitter and receiver pair, readings are displayed using a LCD display and the speed is controlled by a pot. It works on the principle that the number of times the IR receiver-transmitter circuit is cut and re-established in a second gives the number of rotations per second. The value is displayed on the LCD display.

The image shows the tachometer circuit that has been used as our test model for tracking the RPM of a dc motor that has been powered independently. The Figure 4 shows the initialization of the circuit when there is no movement and LCD reads 0 RPM. The Figure 5 shows an RPM of 273 that the IR module tracks from the motor rotational circuit that is being tested by us. The LCD displaying final RPM output as shown in Figure 6.

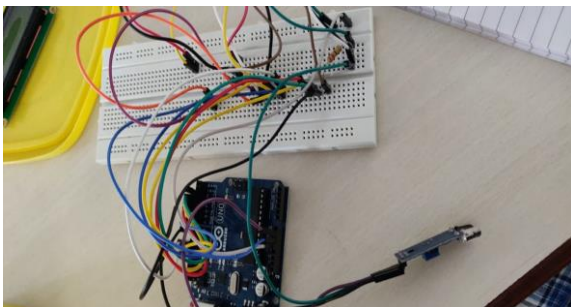


Figure 4. Experimental setup

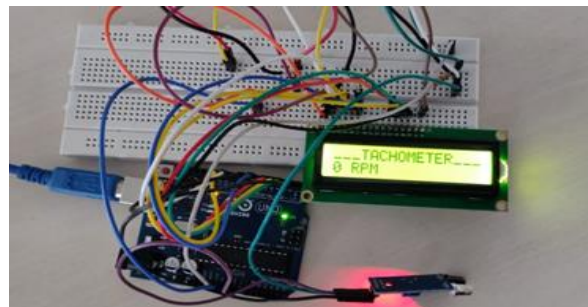


Figure 5. Initialized circuit for RPM measurement



Figure 6. LCD displaying final RPM output

6. CONCLUSION

In this fast-growing world of technologies in the industrial enterprise, motor and generator are one of the essential parts of the industry. For a particular application in industries, the speed is needed to be fixed and for this reason it is necessary to know the speed in revolution per minute (RPM). Tachometer is a device that is used to count the revolution per minute (RPM) of a rotating machine. In our project that is (contact less digital tachometer), we've accomplished satisfactory results as per our motive to create an experimental setup that can be used to calculate the RPM of the machine.

REFERENCES

- [1] V. Ramya, and B. Palaniappan, "Embedded system Hazardous Gas detection and Alerting," *International Journal of Distributed and Parallel Systems*, vol. 3, no. 3, pp. 287-300, 2012.
- [2] Selvapriya, Sathya Prabha, Abdulrahim, Aarthi K., "LPG Leakage Multilevel Alerting System," *SathyaPrabha S2, Abdulrahim M 3*, 2013.
- [3] Sagar Shinde, S. B. Patil, A. J. Patil "Development of Movable Tanker Leakage Detection Using Wireless Sensor Network Based on Embedded System," *International Journal of Engineering Research and Application (IJTERA)*, vol. 2, no. 6, pp. 1180-1183, 2012.
- [4] Ashish Shrivastava, et al., "GSM Based Gas Leakage Detection System," *International Journal of Emerging Trends in Electrical and Electronics*, vol. 3, no. 2, pp. 42-45, 2013.
- [5] T. Soundarya, and J. V. Anchitalagammai, "Control and Monitoring System for Liquefied Petroleum Gas (LPG) Detection," *Int. Conference on Innovations in Engineering and Technology*, vol. 3, no. 3, pp. 696-700, 2014.
- [6] Zhao Yang, et al., "Research on Leakage for Detection Analysis of Leakage Point in the Gas Pipeline System," *Open Journal of Safety Science and Technology*, vol. 1, no. 3, pp. 94-100, 2011.
- [7] Selvakumar K, Palanisamy R, and Vijayakumar K, "Hysteresis control 3-Level SI-NPC inverter with Wind energy system," *International Journal of Power Electronics and Drive Systems*, vol. 8, no. 4, pp. 1764-1770, 2017.
- [8] Bhagya Lakshmi., et al., "Heart Beat a Detector using Infrared Pulse Sensor," *IJSRD - International Journal Scientific Research & Development*, vol. 3, no. 09, 2015.
- [9] Nazmus Saquib, et al., "Measurement of Heart Rate Using Photoplethysmography" *Embedded Lab. PC-based rate monitor using Arduino and Easy Pulse sensor*, 2015.
- [10] R. Raj and S. J. Jothi, "Estimation of Heart Rate from Photoplethysmographic Signal Using SVR Method," *The International Journal of Science & Technoledge*, vol. 2, no. 2, 2014.
- [11] Palanisamy R, and Vijayakumar K, "Paper SVPWM for 3-phase 3-level neutral point clamped inverter fed induction motor control," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 9, no. 3, pp. 703-710, 2018.
- [12] Sagar C. Chhatrala, Mitul R. Khandhedhia., "Ubiquitous Physiological Monitoring of SPO2 & Heart Rate," *International Journal for Research in Technological Studies*, vol. 1, no. 2, pp. 4-7, Jan. 2014.
- [13] Ufoaroh S. U, Oranugo C. O, Uchechukwu M. E., "Heartbeat monitoring system using GSM technology," *International Journal of Engineering Research and General Science*, vol. 3, no. 4, pp. 26-34, 2015.
- [14] B. S. Kim and S. K. Yoo, "Motion artifact reduction in photoplethysmography using independent component analysis," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 3, pp. 566-568, 2006.
- [15] Z. Zhang, Z. Pi, and B. Liu, "Troika: A general framework for heart rate monitoring using wrist-type photoplethysmographic signals during intensive physical exercise," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 2, pp. 522-531, 2015.
- [16] F. Wadehn, D. Carnal, and H.-A. Loeliger, "Estimation of heart rate and heart rate variability from pulse oximeter recordings using localized model fitting," *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 3815-3818, 2015.

- [17] R. Graaff, A. Dassel, W. Zijlstra, F. De Mul, and J. Aarnoudse, "How tissue optics influences reflectance pulse oximetry," *Oxygen transport to Tissue XVII*. Springer, Boston, MA, pp. 117-132, 1996.
- [18] S. M. L'opez, et al., "Heuristic algorithm for photoplethysmographic heart rate tracking during maximal exercise test," *Journal of Medical and Biological Engineering*, vol. 32, no. 3, pp. 181-188, 2012.
- [19] Z. Zhang, "Heart rate monitoring from wrist-type photoplethysmographic (PPG) signals during intensive physical exercise," *The 2nd IEEE Global Conference on Signal and Information Processing*, pp. 698-702, 2014.
- [20] P. Stoica and R. L. Moses, "Spectral analysis of signals," *Pearson/Prentice Hall Upper Saddle River, NJ*, 2005.
- [21] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm," *IEEE Trans. on Signal Processing*, vol. 45, no. 3, pp. 600-616, 1997.
- [22] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289-1306, 2006.
- [23] M. Elad, "Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing," *Springer Science & Business Media*, 2010.
- [24] M. F. Duarte and R. G. Baraniuk, "Spectral compressive sensing," *Applied and Computational Harmonic Analysis*, vol. 35, no. 1, pp. 111-129, 2013.
- [25] R. Krishnan, B. Natarajan, and S. Warren, "Two-stage approach for detection and reduction of motion artifacts in photoplethysmographic data," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 8, pp. 1867-1876, 2010.
- [26] Fernandez S. G., et al., "Unmanned and autonomous ground vehicle," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 5, pp. 4466-4472, 2019.
- [27] M. Ram, et al., "A novel approach for motion artifact reduction in PPG signals based on AS-LMS adaptive filter," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 5, pp. 1445-1457, 2012.
- [28] R. Yousefi, et al., "A motion-tolerant adaptive algorithm for wearable photoplethysmographic biosensors," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 2, pp. 670-681, 2014.