

# METRO TRAIN PROTOTYPE USING AT mega 16 MICROCONTROLLER

**Mohammad Mohsin**

*M .tech ( Electronic Circuits And Systems )*

mohammadmohsin339@gmail.com

**Er.Tarana Afrin Chandel**

*Jr. Associate Professor (Integral University)*

taranaafrin82@gmail.com

**Abstract** - Metro trains are the most efficient means of transportation used in major cities so as to avoid extreme congestion on the road, ever slowing speeds, increasing accident rate, fuel wastage and environmental pollution. This paper presents a prototype of this train such that it can be developed into a driverless system and controlled by AT mega16 microcontroller which behaves as the CPU which helps us to operate metros 24\*7 days and reduce human errors and provide us sustainable, non polluting transportation option anytime and anywhere . These trains are equipped with the microcontroller AT mega 16 which controls the train. The train is programmed for the specific path. Every station on the path is defined; stoppage timing of the train and distance between the two stations is predefined, interrupts are used to handle emergencies . Simulation is done on a Development Board which is a hand-held, battery-powered microcontroller board based on the Atmel and a robotic model is generated by use of embedded system. The working of train is pre-defined to AT mega 16 by use of BASCOM software , program is converted to hex file by compiling the program and AVR DUDE software upload the hex file in AT mega 16 microcontroller .

**Keywords:** AT mega 16 microcontroller, Stepper Motor, LDR sensor, Switch, IC L293D.

## 1. INTRODUCTION

Transportation is the backbone of our country which depends almost entirely on buses, autos, taxis etc. for mode of mass transport. All these services are inadequate and heavily over-crowded. Pollution related health problems are reaching disconcerting levels. Immediate steps are, therefore, needed to improve both the quality and availability of mass transport service. This is possible only if a rail-based mass transit system "Ref. [1]", which is non-polluting,

is introduced in every city without further delay . Here comes the concept of metro train which is time saving

option for commuters, provides reliable and safe journey, reduces atmospheric pollution. Reduces road accident and fuel consumption is reduced. Vehicle operating costs decreases and increase in the average speed of road vehicles. It provides improvement in the quality of life and is more attractive for cities for encouraging economic investment and growth.

## 2. TECHNIQUE USED

An **embedded system** is a combination of hardware and software in such a way that it performs a dedicated task with the help of a microcontroller or microprocessor and has various daily life operations within a larger mechanical or electrical system, often with real time computing constraints. Due to the concept of embedded system it is possible to design a train controlled by a CPU and it leads to the rise of driverless system "Ref. [2]" which actually benefits us to reduce human errors and increase the working hours of metros as human intervention is very little required .

## 3. METRO TRAIN WORKING CONCEPT

METRO TRAIN is controlled by microcontroller AT mega 16 "Ref. [3]" which acts as CPU and is programmed by the user as per the need. In our prototype, we have made provisions for the train to stop at every station for 2 seconds which is made by white strip on black track, then door is opened in 2 microseconds; waits 1 sec for entry, then closes in 2 microsecond; moves for 2 sec to pass through the white strip of station and then again follows the track until the next station. The track is sensed by IR sensor which moves only on black track line laid and stops for the station which is indicated by white marked strip. Door is made by a stepper motor and the wheels of train are also driven by stepper motor. But stepper motor requires higher current value; hence IC L293D is interfaced with microcontroller which performs current amplification. 7 segment LCD screen is used so as to display the status of the train whether moving forward or is at station. LDR sensors are used to detect the track as well as the obstacles in front of train. A buzzer is

also connected with microcontroller which alarms when an obstacle comes in front of train. As soon as obstacle is present on the track, the buzzer starts alarming and also alarm provision is also provided for emergency situation and train stops. It contains a provision of emergency switch for alarming; may function as a chain. As soon as switch is pressed, buzzer starts ringing and train stops until the switch is in off state. The switch when pressed generates an interrupt signal which enables the buzzer pin to get on and microcontroller halts the journey simultaneously. It acts as a potential protection for threat against robbery and fire detecting alarm, so that emergency is treated and care is taken instantly.

#### 4. PROPOSED CIRCUIT FOR PROTOTYPE

This circuit diagram is used for interfacing components with microcontroller AT mega 16 and is used for assembling the hardware of robot metro train which is shown in figure 1. "Ref. [4]."

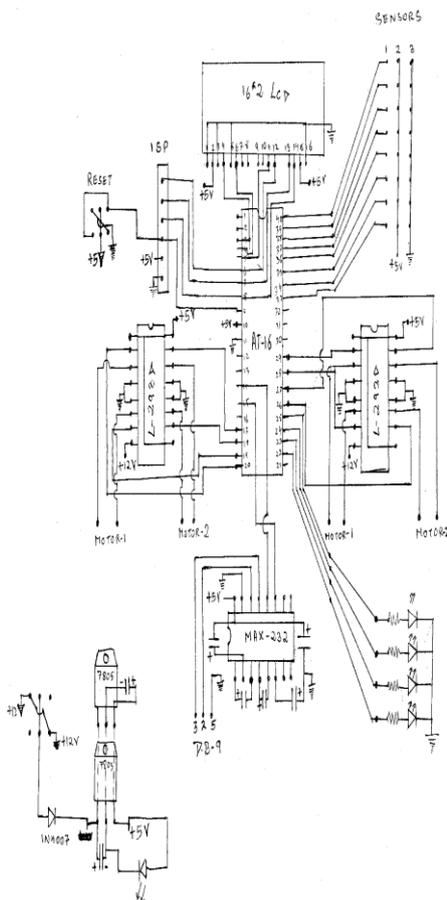


Figure 1: Hardware connections of Microcontroller AT mega16

Development Board is a hand-held, battery-powered microcontroller board which is based on the Atmel ATmega16/32 microcontroller.

It includes 16/32K of Flash RAM, outputs for four DC motors or two stepper motors, inputs for a variety of sensors, and a 16\*2 alphanumeric LCD screen. The Development Board runs BASCOM-AVR, a cross-platform, specialized AVR version of the BASIC programming language. "Ref. [5]".

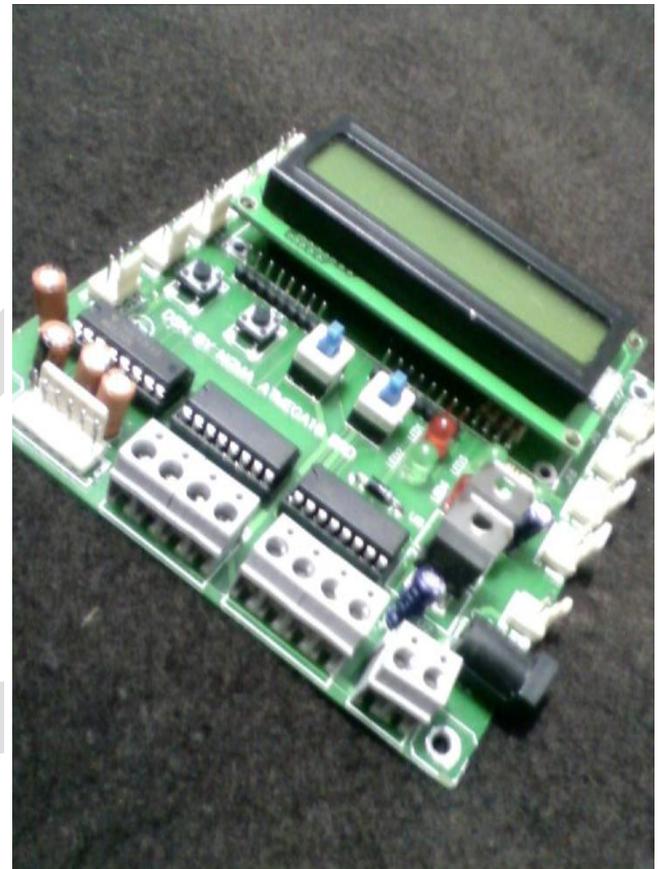


Figure 2: AT mega 16 microcontroller placed Development Board.

#### 5. SOFTWARE CODING TRANSFER

Now the microcontroller AT mega 16 is programmed using BASCOM software. The instructions which are needed to drive and control the metro train is given using programming language "Ref. [6]".

Once the programming is done, press F7 to compile. Plug in the USB programming cable in your computer. Now install the driver of the USB programmer, whose folder is defined by the name WINDRIVER. Location is traced in the folder when the computer asks for the driver. Now open avrdude-gui which will help you to transfer the hex code of your programme to the microcontroller. "Ref. [7]" Programming in BASIC language using BASCOM-AVR is shown in fig 3.

AVR Dude- GUI looks like which is shown in figure 4 and in this way after compilation we put the commands necessary to control the metro train within the microcontroller memory .

### 6. STIMULATION RESULTS AND TEST RUN

Power supply is switch ON , Reset and Clear switch both are ON. LCD displays message “SENSE OBSTACLE”. LDR SENSOR is calibrated and obstacle is sensed. LCD displays message “SENSE NO OBSTACLE”. LDR SENSOR is again calibrated and no obstacle is sensed. When LCD displays message “SENSE COLOUR”. LDR SENSOR is calibrated and color of track is sensed, whether it is black or white If the color is black, black is sensed, if the color is white it will sense and produce a message that the track is white the train to stop at every station for 2 seconds which is made by white strip on black track , then door is opened in 2 microseconds; waits 1 sec for entry, then closes in 2 microsecond ; moves for 2 sec to pass through the white strip of station and then again follows the track until the next station.. As soon as obstacle is present on the track, the buzzer starts ringing and alarming for emergency situation and train stops. When obstacle is removed, train recovers its journey. It contains a provision of emergency switch for alarming As soon as switch is pressed, buzzer starts alarming and train stops until the switch is in off state. It can be used in case of health related emergencies to make system alert and could act as a potential protection for threat against robbery and fire detecting alarm “Ref [8]”.

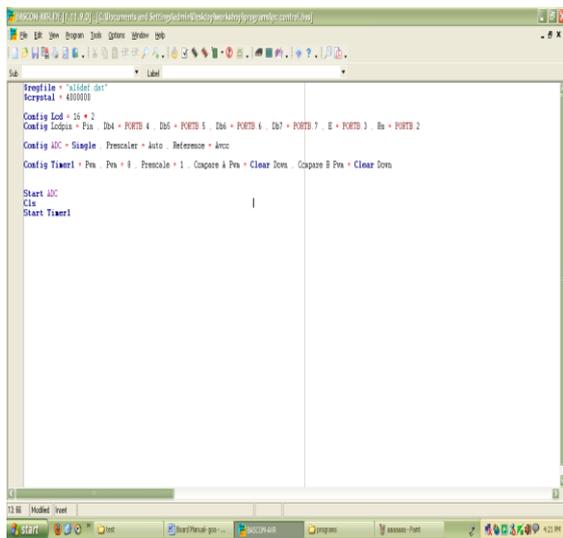


Figure3: bascom avr programming done using BASIC language .

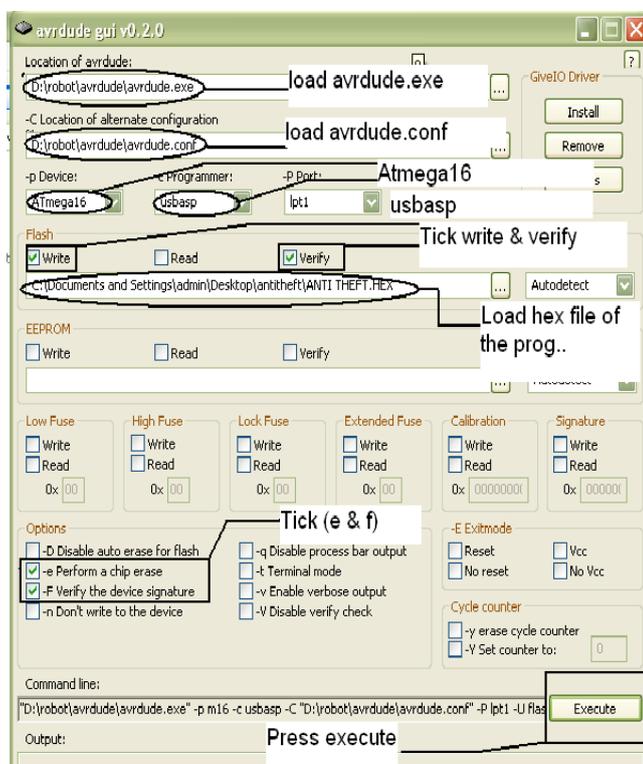


Figure 4: transferring of hex file to micro controller using avr dude gui

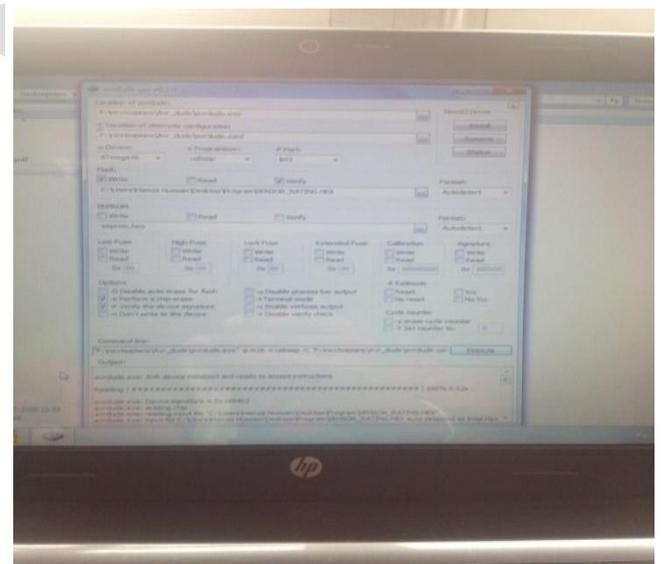


Figure 5: real-time avrdude gui window shown on computer screen



Figure 6 : proposed metro train model showing on LCD status of train .

Fig.6 shows the real time picture of the proposed model physically. Test run is done on white sheet of paper which has black colored line metro track and white strips as stations.

## 7. CONCLUSION

Test run is completed and hence we produce a prototype of a metro train which could run without need of driver and we demonstrate the concept of microcontroller based simulation for incident management. Architecture for the purpose was developed.

## 8. REFERENCES

- [1] V.Sridhar , “Automated system design for metro train” , International Journal of Computer Science Engineering (IJCSE) .
- [2] Thabit Sultan Mohammed , Wisam Fahmi Al-Azzo , Mohammed Ahmed Akaak , Mohammed Laheeb Suroor , “ Full Automation in Driverless Trains: A Microcontroller-Based Prototype .” International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (An ISO 3297: 2007 Certified Organization) Vol. 3, Issue 7, July 2014.

[3] Nikhil , et al , “A MINOR PROJECT REPORT ON METRO TRAIN PROTOTYPE “, KURUKSHETRA UNIVERSITY, KURUKSHETRA .

[4] Udit Bhatnagar, Vibhuji Tiwari, Veer Vikram Singh, Preeti Yadav B.tech (EC) “ Pseudo drive metro train ”, International Journal of Advanced Technology & Engineering Research (IJATER) [www.ijater.com](http://www.ijater.com) ISSN No: 2250-3536 Volume 3, Issue 3, May 2013 27.

[5] Astrom P (1992), “Control electronics on rail vehicles”. In: Proceedings of the ASME/IEEE Spring Joint Railroad Conference, pp 107–116

[6] Raymond AJ (2000) “Formal software techniques in Railway Systems”. In: Proc. 9th IFAC Symposium on Control in Transportation Systems, Braunschweig, Germany, pp 1–12.

[7] M. Verle . “PIC Microcontrollers - Programming in C”. mikro-Elektronika; 1st edition .2009.

[8] Hachiga A (1993) “ The concepts and technologies of dependable and real time computer systems for Shinkansen train control.” and “Dependable computing and fault tolerant systems.” Springer Verlag, Berlin, pp 225–252.