INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY
ISSN: 0974 – 3154 Volume 6, Number 2 (2013), pp. 106 – 109

AURDINO AND DTMF CONTROLLED INDUSTRIAL TRANSPORTER

Haran.P.C[1], N.Gokulraj[2] , P.Arun[3] and Dr.M.Saravanan[4]

[1,2,3] Department of EEE, Thiagarajar College of Engineering, Madurai,Tamil Nadu, India
[4] Professor, ECE, Thiagarajar College of Engineering, Madurai,Tamil Nadu,India.

ABSTRACT

Industrial transportation faces various challenges. Transporting cargos from one section to another through monotonous paths are costly with human run vehicles. This commercially viable DTMF transporter controlled by aurdino controllers and run by stepper motor allows a customer to use a specially designed application (designed with processing) to specify the path it has to travel. This can then be controlled by the customer from a remote location using his mobile to drive it from one location to many other locations.

Keywords—aurdino, processing, dtmf, stepper motors

1. INTRODUCTION

Industrial automation has grown leaps and bounds but challenges always remain. Take instances of locations where human entry is harmful in case of labs or mines with harmful gases. Consider paths which are travelled often for transportation of materials from one section to the other which takes place more often. In these situations employing a vehicle with driver is costlier. Employing an automated machine allows languor free operation. This bot provides solution for these problems. This robot could be controlled from any remote location to make it travel a predetermined path to any location. The number of locations could be increased proportionally with the cost.

2. COMPLETE PACKAGE

The customer is provided with the robot with no programmed locations and paths set. The customer is specified the number of locations which could be set for a particular range of distances between the locations. For example the maximum locations are 5 if the average distance between locations is 200 meters but it is 2 for average distance of 500 meters. This example is not the accurate data but a random data to give an idea on the specifications. The user is then provided with a specially designed application wherein he gets to enter the path between locations by entering the distances. Now after programming the robot is placed initially in any particular location. A sim card inserted cell phone (in our prototype) or a gsm module is embedded with the bot. on calling that sim from any remote cell phone you are connected to the bot. on pressing any particular key in your mobile the bot moves to that location. Example if the bot is in position 1 and on pressing 2 on your mobile it moves to position 2.

Figure 1. *Industrial Transporter prototype*

3. SOFTWARE AND HARDWARE

3.1 IBV1

The application named ibv1 is created using open source software processing. A screen shot of it is shown below.

Once the robot is serially connected to your computer through one of the USB ports, the application can be used for programming. Clicking the START button initiates the path setting process. The start and stop locations are entered next. Example clicking 1 and 2 after start indicates that the consumer is going to enter the path from 1 to 2. After entering the start and stop locations POSITION button is pressed. RIGHT and LEFT button indicates a right and left turn respectively. For a straight line path the STRAIGHT button is pressed and the distance to move is entered using the numbers. After the distance to move in straight line is entered, END button is pressed. END is only to indicate the end of a straight path. After the customer finishes setting a path for one start stop location the STOP button is pressed. There is no need to set the reverse path, i.e. entering the location from 1 to 2 is sufficient. 2 to 1 path will automatically deduced by the bot.

The processing software identifies each mouse click and sends a unique byte representation for each button to the serial port connecting the bot to the computer. So the entire path setting control is done inside the customer's premises without the interference from the seller.

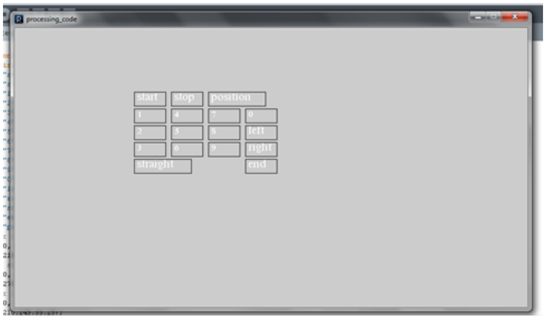

Figure 2. *Screen shot of IBV1*

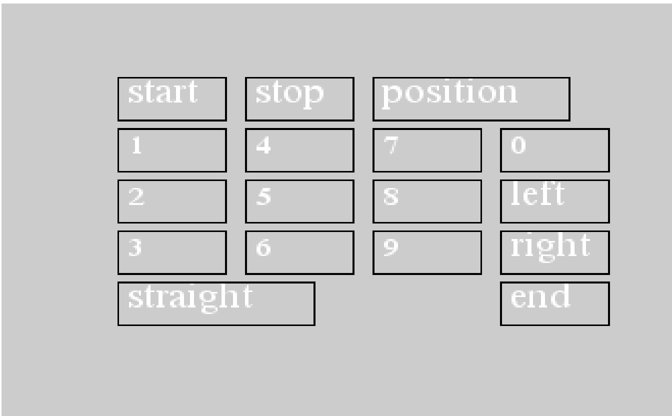

Figure 3. *Zoomed screen shot of IBV1*

3.2 Aurdino controller

The main control of the bot is by Aurdino Uno. This has an ATMEGA328 controller. The open source aurdino software makes the programming easier. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, 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. 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.

Figure 4. *Aurdino uno*

3.3 Interface with computer

After connecting the bot to a computer, the controller is programmed to accept the data from computer through serial interface. Whenever there is a data to be read from the serial port, it is read. The serial data as mentioned represents the path from one location to the other. Data values are usually stored in RAM which is volatile but the data from serial port is stored in EEPROM (1kb) so that the values gets stored permanently unless the bot is reprogrammed.

The controller keeps a 'first' and a 'last' pointer for each path, i.e. for 1-2 path a 'first' of 1 and 'last' of 32 indicates that the entire data needed to move from 1 to 2 is in EEPROM locations from 1 to 32.During this process of entering path a pointer keeps track of the next free EEPROM location, so that all of the 1kb (1024 bytes) can be used in storing data. Example if at the end of setting path for 1-2 positions 32 bytes of EEPROM are filled and if the customer starts entering path for 3-4 it will start from 33 bytes and 33 will be stored as the 3-4 'first' pointer. At the end of the path setting for 3-4, the EEPROM location will be stored as the 'last' pointer of 3-4 path.

3.4 Virtual link between caller and robot

After the path setting process the bot is ready. When a call is made to the sim card in the bot, you are connected to the bot. On pressing any number, using IC 8870 which is a DTMF decoder IC, the number pressed is converted into its binary output. This digital output is given to the controller. The controller detects what number is pressed based on the digital input. The controller always keeps note of the current location and on receiving the next location it jumps to the part of the EEPROM which holds the data for that path. The data are referenced one by one and necessary actions based on the code are taken.

3.5 Stepper motors

The motion of the bot is controlled through two stepper motors. Using the stepper library functions in aurdino software, these motors could be precisely controlled. A bi-directional stepper motor is used. These motors are driven by H-bridge IC SN754410NE which provides the necessary current to drive the motors.

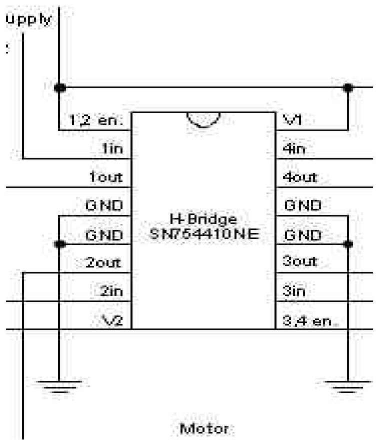

Figure 5. *H-Bridge connection*

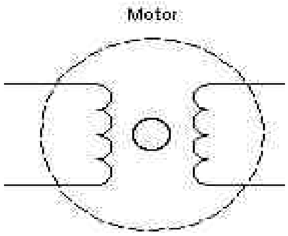

Figure 6. *Bipolar stepper motor*

4. CALIBRATION

The distance travelled in one revolution of the motor is the circumference of the wheel which the motor drives. In a 1.8 degree per step motor. 200 steps are needed to move a distance equal to the circumference of the wheel. Based on this relation the number of steps required to move a distance x is calculated.

When both the motors rotate in the opposite direction, the bot moves in circle of diameter of the width of the bot. So moving right or left involves rotating the motor in opposite direction till they complete one fourth of one complete revolution.

The prototype model could be further improved to design a fully commercial model. It can be widely used in reducing manual labour and in carrying heavier cargo from one place to the other.

5. CONCLUSION

The prototype model could be further improved to design a fully commercial model. It can be widely used in reducing manual labor and in carrying heavier cargo from one place to the other.

ACKNOWLEGEMENT

I would like to thank Dr.M.Saravanan, professor in electrical department, Thiagarajar College of engineering Madurai. for his valuable inputs.

REFERENCES

Massimo Banzi(2008) , Getting Started With Arduino, O’Reilly media, USA.
Casey Reas and Ben Fry(2010), Getting Started with Processing, O’Reilly media, USA.
Tom Igoe(2011) , Making Things Talk: Using Sensors, Networks, and Arduino to see, hear, and feel your world , O’Reilly media, USA.
http://arduino.cc/en/Main/ArduinoBoardUno?token=efREVgAWqTT5f8aSTI9CE
http://www.arduino.cc/en/Reference/EEPROM?token=9FuEEdQbQtaUPGhe5lifC.
http://arduino.cc/forum/index.php/topic,96280.0.html