

APPLICATION OF DUAL TONE MULTI FREQUENCY TECHNOLOGY AND SENSING IN AUTONOMOUS ROBOTIC NAVIGATION

Vishveshwar Ramanathan Subramanian

Student, Grade 12, American Embassy School, New Delhi, India, ishwar.subra@gmail.com

Abstract

The goal of this project is to use DTMF decoders, sensors and micro-controllers to control autonomous navigation of a robot between a start and end point along a fixed path. This has very practical applications, eg. a mobile phone can be used to start the car and allow it to autonomously drive itself to a pick up point and vehicles in warehouses to carry goods from point A to point B, etc. A DTMF decoder is interfaced with the microcontroller, and a phone on auto answer mode is connected to the decoder. When the robot is called and a number on the user’s phone is pressed, the tone is sent to the decoder, which decodes and sends a signal to the microcontroller, and the robot is started. To avoid obstacles the robot has inbuilt Infrared sensors which detect obstacles and inform the microcontroller, which accordingly guides the robot. In this setup only three sensors were used given the limitation in terms of input ports. However, a greater number of sensors would allow for the robot to be aware of more specific situations. Therefore, it can be made more adaptable in varying situations. The test robot was programmed using Embedded C with AVR Studios. This approach uses existing technologies and is very inexpensive.

Index Terms: Drivers, Autonomous and Navigate.

-----***-----

1. INTRODUCTION

Dual Tone Multi Frequency is a technology that can be used to send information through phone lines. It is very common in Australia, where many companies use it as a way to decode and call typed numbers [5]. The idea of using DTMF in applications is not new, but the purpose of this paper is to address a new approach to this. Most mobiles have this technology and by using the right techniques, decoded DTMF signals can be used in robotic microcontrollers to make the robot perform certain actions.

One could use DTMF to create a robot that would navigate its way through a set “arena” or space to a specific location by press of a button on a cell phone. The investigation will be based on the situation where one or several trips have to be made between two points following a path and will try taking into account the changing nature of this environment. The algorithm will mainly focus on starting the robot using a mobile and controlling the robot to navigate past the obstacles in its path to a specific point.

2. WORKING OF DTMF

The DTMF system uses eight frequency signals transmitted in pairs to represent sixteen numbers, symbols and letters (Table1). Pressing a key will cause a high and low tone for each of the two frequencies. The HT9170B IC, using digital counting techniques, decodes these two tones to determine the key being pressed. The tone from the phone is filtered through the operational filter. A high voltage on the Output enable pin

(pin10) enables outputs D0, D1, D2 and D3, which are the decoded outputs of the IC. DV (pin 15) is an output pin that is set high by the IC just after the output pins D0-3 have been filled; this action shows the data is valid and usable. RT/GT and EST monitor the time taken for the authenticity check of the tone. If the tone is too long the steering control mechanism of these pins will automatic fill the latches of the D0, D1, D2 and D3 pins after which it will set the DV pin high to indicate the output is ready for use by the microcontroller. By this process the DTMF decoder can decode 16 different key tones in 4 bit binary code output.[1]

Table 1- DTMF Keypad frequencies [1]

	1209 Hz	1336Hz	1477Hz	1633Hz
697Hz	1	2	3	A
770Hz	4	5	6	B
852Hz	7	8	9	C
941Hz	*	9	#	D

Table 2- Four Bit binary code for each key tone for number [1]

Digit	OE	D0	D1	D2	D3
1	H	1	0	0	0
2	H	0	1	0	0
3	H	1	1	0	0
4	H	0	0	1	0
5	H	1	0	1	0
6	H	0	1	1	0
7	H	1	1	1	0
8	H	0	0	0	1
9	H	1	0	0	1
0	H	0	1	0	1

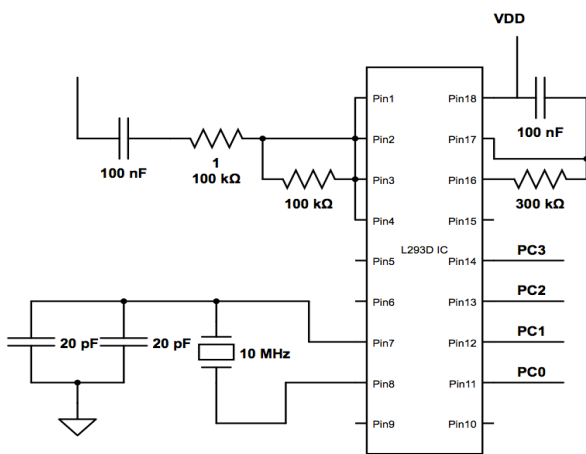


Figure 1 – HT9170B IC Decoder

3. THE MOTOR DRIVE AND MC

3.1 Pins and Basic Working of the Driver

The motor being used is a Dual H-Bridge Motor Driver that can provide bidirectional currents, the motors are controlled by a 5V “logical” supply voltage and run on a separate 12V supply. It consists of Input pins 1-4 (pins 2, 7, 10, 15) and output pins 1-4 (Pins 3, 6, 11, 14). Pin 1 (Enable 1) starts the left Bridge while Pin 9 (Enable 2) starts the right bridge. Inputs 1 and 2 correspond to one motor where if Input 1 has a high voltage (5V) and Input 2 has a low voltage (0V), the

motor moves forward, whereas if the opposite occurs the motor moves in reverse. The VSS pin provides the logical supply voltage, 5V, whereas the VS pin provides the voltage for the motors to run on: this is anywhere between 5V to 12V. OUPUT 1-4 is the corresponding output pin to INPUT1-4.

3.2 The Microcontroller Unit

The microcontroller used in this circuit is an AVR® Atmega 8 built with RISC architecture and 8Kb of self-Programmable flash memory. It can process 8 bits of data in one clock cycle. The RISC architecture allows the MCU to process an instruction per second. [1]

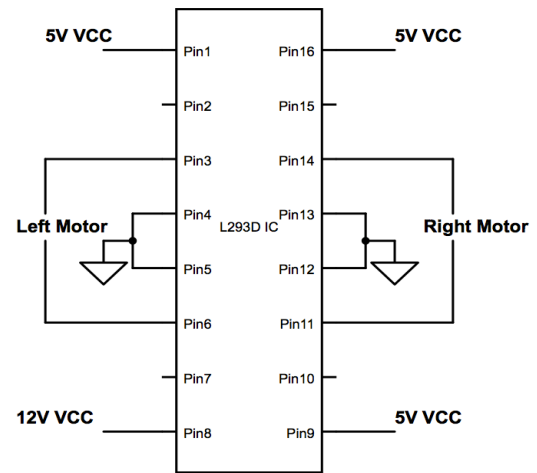


Figure 12 – Diagram of the Motor Driver

3.3 Interfacing to the MCU, HT9170B IC and the IB L293D

In this case, D0-3 of the DTMF decoder will connect to the Pins C0-3 (pins24-26) on the Microcontroller as the program sets PORTC as the input port of the Microcontroller. Port B is set to be the output port of the microcontroller. Pins B4-7 (pins 18,19,9,10) on the microcontroller are connected to Input pins 1-4 on the IBL293D, the enable pin on the motor driver is connected to the 5V logical voltage supply so it is enabled fully. [3]

When a call is made to the cell phone in the robot and that call is accepted by the cell-phone (by setting the phone to auto-accept) a connection between the two phones is established. When a tone is pressed on one phone (i.e. the user’s phone) the phone on the robot receives this tone and feeds it to the circuit through the headphone jack. The decoder then identifies the tone and feeds it to the Microcontroller as binary code, e.g. 8 is 1000. According to the program the input will be responded to with an output from the microcontroller, for example, key ‘8’ outputs 00000110, causing the robot to move in reverse. In this case pins PB5 and PB6 are high and

therefore drive the motors in reverse [1][2]. However when two is pressed, the microcontroller is told to read the inputs from the sensors in order to determine the direction of its motion.

4. The IR Sensor

4.1 The working of the IR Sensor

To detect obstacles, an IR sensor will be used in the robot. The IR sensor consists of a Photo-Diode and an LED. The LED emits Infrared radiation. If the surface is reflective it will reflect this light and the reflected light will be absorbed by the photodiode. The resistance of the Photo diode decreases as light falls on it and the potential difference across diode increases.[4] The Op-Amp in this circuit is set in comparative mode so once the increasing voltage in Pin 3 of the Op-Amp exceeds the voltage of Pin 2, the Op-Amp will produce a high output (5V). If the surface is not reflective and the Photo-Diode detects no reflected IR radiation, the resistance of the diode will remain infinite.

One can change the resistance of the variable resistor, by setting the resistance high, the voltage at Pin 2, the reference voltage, is much smaller making the sensor more sensitive and by setting the resistance low the reference voltage increases decreasing the IR sensors sensitivity. [1] [4]

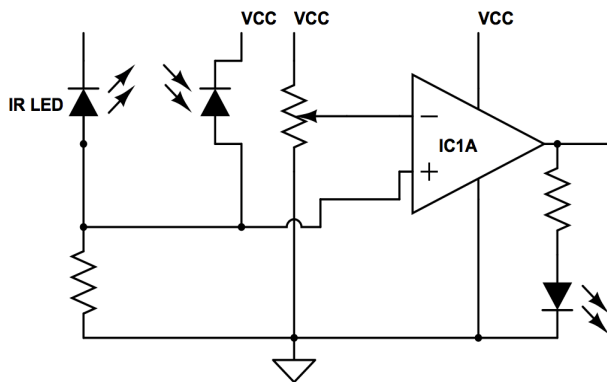


Figure 4- IR Sensor Diagram

4.2 Interfacing the IR Sensor with the motor driver

PC4-6 are interfaced with IR sensor. When the key '2' is pressed, the autonomous program is activated. The reading from the sensor is fed through the ports PC4-6 to the microcontroller, which creates the programmed output through the pins PB4-7, which are connected to the input pins 1-4 on the motor driver. For example, if the right and left sensors detect a wall or any other obstacle, the robot will move forward, 00001001.[1]

5. Setup, Control and Environment

5.1 Setup and Control

To setup this robot, cut out the headphone jack from old headphones and connect it to the circuit. After which the headphone jack is inserted into the port of the phone. An external phone must then call the phone connected to the circuit. Put the phone connected to the circuit on auto answer. Then one can press the buttons on the external phone and the DTMF tones are conveyed to the phone connected to the circuit. The robot acts accordingly.

In this case key '2' initiates the sensors and the robot reads the input from the sensor to determine the direction in which it moves. However manual control is still a possibility given that 1, 4, 6, 8 move the robot forward, left, right, and backward respectively.

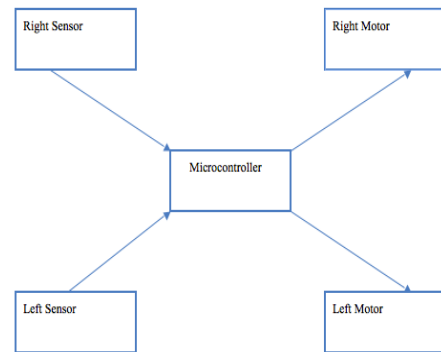


Figure 5- Sensor Circuit

5.2 Environments/Scenarios

The design is flexible to many uses, it can be used in cars when they come from their parking space to their owner and it can be applied to automated trolleys in factories that transport parts around the premises. The sensors are programmed to detect obstacles and so will navigate themselves through pathways by using these obstacles as guides. The following program was done with the trolley application in mind.

6. ALGORITHMS AND PROGRAMMING

6.1 Algorithm

```

Call phone in robot
Initiation of DTMF decoder
Set sensor L, sensor R, sensor F as 0
Sensor L is = SL and Sensor R = SR and sensor F = SF
While(1)if DTMF=2
Initiate logic
End
    
```

Algorithm Logic

<i>Situation</i>	<i>SF</i>	<i>SR</i>	<i>SL</i>	<i>Action</i>
1	0	0	0	Forward
2	1	0	0	Stop
3	1	0	1	Right 90
4	1	1	1	Stop
5	0	1	0	Forward
6	0	0	1	Forward
7	0	1	1	Forward
8	1	1	0	Left 90

6.2 Program

The program was done using AVR® studios using the C language. It detects lines other obstacles in the car park.

```
#define F_CPU 1000000UL
#include <avr/io.h>
#include<util/delay.h>
int move_back=0b00000110;
int move_right=0b00001010;
int move_forward=0b00001001;
int move_left=0b00000101;
int stop=0b00000000;
int left_sensor_on=0b0010000;
int right_sensor_on=0b0100010;
int forward_sensor_on=0b0001000;
int left_sensor_off=0b0000000;
int right_sensor_off=0b0000000;
int forward_sensor_off=0b0000000;
int main(void)
{
  DDRB=0b11111111;
  DDRC=0b00000000;
  int ls=0;int rs=0; int fs=0;
  int DTMF=0;
  while(1)
  {
    ls = (PINC&0b0010000);
    rs = (PINC&0b0100000);
    fs = (PINC&0b0001000);
    DTMF=PINC&0b0001111;
    if(DTMF==2)
    {
      PORTB=move_forward;
      _delay_ms(4000);
      PORTB=move_right;
      if ((ls==left_sensor_on) & (rs==right_sensor_on) &
      (fs==forward_sensor_on))
      {
        PORTB=stop;
      }
    }
  }
}
```

```
if((ls==left_sensor_on) & (rs==right_sensor_off) &
(fs==forward_sensor_on))
{
  PORTB=move_back;
  _delay_ms(3000);
  PORTB=move_right;
  _delay_ms(3000);
}
if((ls==left_sensor_off) & (rs==right_sensor_on) &
(fs==forward_sensor_on))
{
  PORTB=move_back;
  _delay_ms(3000);
  PORTB=move_left;
  _delay_ms(3000);
}
if((ls==left_sensor_off) & (rs==right_sensor_off) &
(fs==forward_sensor_on))
{
  PORTB=stop;
}
if((ls==left_sensor_on) & (rs==right_sensor_on) &
(fs==forward_sensor_off))
{
  PORTB=move_forward;
}
if((ls==left_sensor_on) & (rs==right_sensor_off) &
(fs==forward_sensor_off))
{
  PORTB=move_forward;
}
if((ls==left_sensor_off) & (rs==right_sensor_on) &
(fs==forward_sensor_off))
{
  PORTB=move_forward;
}
if((ls==left_sensor_off) & (rs==right_sensor_off) &
(fs==forward_sensor_off))
{
  PORTB=move_forward;
  if((ls==left_sensor_off) & (rs==right_sensor_off) &
(fs==forward_sensor_on))
  {
    PORTB = move_right;
  }
}
}
if(DTMF==4)
{
  PORTB=0b00001000;
}
if(DTMF==6)
{
  PORTB=0b00000001;
}
if(DTMF==8) // if Key 8 of cell phone pressed
```

```
{  
PORTB=0b00000110; // move back  
}  
}
```

Vishveshwar became exposed to various technologies like DTMF and UART. Ever since he has been investigating the application of these technologies to the real world

CONCLUSIONS

This robot is only a rudimentary version of what could be implemented in the real world. Given the limitation in terms of input ports, only three sensors were used in this robot. Therefore only a certain set of conditions could be taken into account. What is represented here is a reasonably accurate solution to the navigation of vehicles in a specified area.

The program allows for both manual and autonomous features so the user can choose between the two. A camera could also be added as an additional feature to allow the user to track the progress of the vehicle as it performs its' task. By using Ultrasound sensors instead of infrared, one could extend the application of this technology to scenarios in which infrared would be ineffective. The investigation represents a step into autonomous vehicles that could well be implemented in the very near future. While Autonomous cars are still in their infancy, the simple use of IR technology and Dual Tone Multi Frequency could lead the way to more advances in autonomous vehicles. It is also fairly affordable given that IR and DTMF are common technologies.

REFERENCES

- [1] Sharma, Toshendra K. Robotics with AVR. 2nd ed. Noida: Robosapiens India Publications, 2011. Print.
- [2] Gupta, Sabuj D., Arman R. Ochi, Mohammad S. Hossain, and Nahid A. Siddique. "Designing & Implementation of Mobile Operated Toy Car by DTMF" International Journal of Scientific and Research 3.1 (2013): n. pag. Web. 12 July 2013. <<http://www.ijsrp.org/>>.
- [3] Singh, Surinder, Mr. "DTMF-Tester." Electronics for You 2003: n. pag. Web.
- [4] "Infrared Sensor Designing- Basic." Robosense. N.p., n.d. Web. 14 July 2013.
- [5] Hector of SCP. "Dual Tone Multiple Frequency A Guide to Understanding and Exploiting Australia's Most Common Telecommunications Signaling Method." N.p., n.d. Web. 12 July 2013. <<http://www.pogden.co.uk/>>.

BIOGRAPHIES



Vishveshwar R Subramanian is currently a 12th grade student at the American Embassy School of New Delhi. After taking part in his schools FTC and IRO teams, he became interested in autonomous systems, especially in field robotics. While at a Summer Internship at Robosapiens India,