A Low Cost MSP430 Based Automated Manual Transmission System

Aronee Dasgupta[1], **Roopa Nagaraj**[2]

[1, 2]Department of Telecommunication, RV College of Engineering. Bangalore, India

Abstract: ***Automatic gearboxes change the gear ratios mechanically and relieve the motive force from doing it manually. However, their high cost and low fuel economies have prevented their widespread adoption in the developing world. Manual transmission however causes physical stress and has a steep learning making it a difficult choice for novice drivers. This paper describes a prototype of a Low Cost add-on that can be installed in existing cars and automates the process of changing the gear ratios. The device contributes to a high fuel economy and prevents unnecessary gear shifts.***

Keywords: MSP430, automated manual transmission;; low cost automatic car, automatic cars in asia

1. Introduction

Most of the lower end cars sold in India do not have the option of automatic transmission. Even if they do, their prices vary by two to three lakh INR (compared to manual transmission). The manual transmission has certain advantages that the automatic transmission models cannot provide. Manual transmissions are a lot cheaper than automatics. Choosing to buy a manual over an automatic saves you an average of 2 lakh INR, depending on the make and model of the car. Manual transmissions are lighter and easier to maintain. Also, a manual transmission doesn't have a hydraulic pump, which puts a big load on your engine. One of the major faults of automatic transmission systems is that they are vulnerable to leakage of engine fluids and this in turn is very destructive. Alongwith this repairs and maintainence is costiler for automatic transmissions**.** In 2010 25.03 million light vehicles and cars were equipped with automatic transmission systems. ***[1]*** Analysts forcast this number will rise to around 28.65 milllion in 2015, an increase of over 12% .The automobile industry in India has quickly caught on to the growth of the transport sector and produces nearly 4.6 million vehicles each year. On an average, there are about 1400 cars added to the roads of the capital city of New Delhi on a daily basis. ***[2]*** It's the second fastest growing automobile market in the world after China. This steep increase in car population on our streets leads to increased stop-go traffic. Due to a maturing markets India, Brazil and China there has been a renewed interest in automatic transmission cars among customers in crowded cities. The latest trend to grip the automobile manufacturers and the Indian audience alike is automated manual transmissio (AMT). However the cost difference between a purely manual transmisson and a AMT transmission is close to $800 in India. ***[3]*** However drivers in these countries suffer from stress due to adverse road condtions. Researchers have found a stong link between increasaing blood sugar levels, anxiety, high blood pressure and driving manual transmission cars in heavy traffic. Due to constant pressing of the clutching some motorists have reported lower back pain. Almost non-existent are diesel-powered automatics, In the hatch there really isn't a choice when it comes to diesel automatics, because diesel engines are more expensive (CRDI and turbocharging), adding an auto box simply throws the pricing equation into disarray.There is a huge market potential for a sub 5 lakh INR diesel automatic hatch. So any existing manual transmission car can be converted to an automatic. The low popularity of automatic cars in India is driven by cost.

A. Technical Background

The solution and prototype is unique and hasn't been developed before. We have used the standard datasheets and performance parameters for MSP430, Hall Effect Sensor DRV5013 and the servo motors.

B. Proposed Solution

Hence we have developed a prototype that automates the process of shifting gears in manual transmission. It is a low cost add-on which can be fitted by a certified technician on the powertrain of the car. The add-on is expected to cost less than $300 (taking into account, testing and NRE). This product provides automatic gear shifts based on the rpm reading from the tachometer. Enthusiasts can disengage the automatic system and take over the manual gear shift when they hit an uncongested road. On the click of a button provided to the driver the system takes over the gear stick and the clutch from the driver to take finer control over the gear shifts. By doing so, we aim to perfect this device so we can launch it in the market within the next decade. It will be a boon for thousands of motorists in developing countries. There should be marked decrease in health problems and anxiety arising from traffic related issues.

2. Proposed Solution

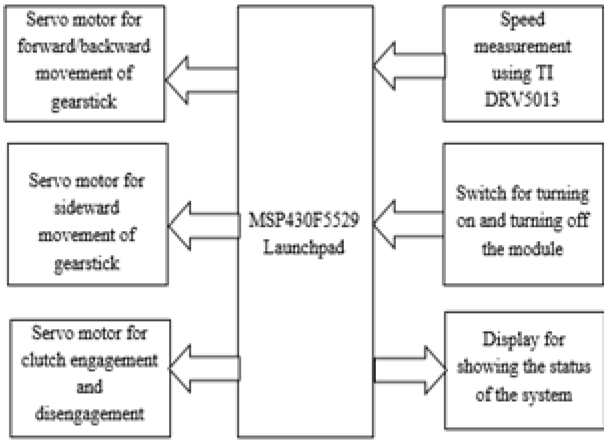
Servo motor for forward/backward movement of gearstick
Servo motor for sideward movement of gearstick
Servo motor for clutch engagement and disengagement
MSP430F5529 Launchpad
Speed measurement using TI DRV5013
Switch for turning on and turning off the module
Display for showing the status of the system

Figure 1: Block diagram of the Proposed Module

The module is controlled by a MSP430F5529 Launchpad. There are three servo motors. One motor controls the forward/backward motion of the gear stick, the second motor controls the sideward motion of the gear stick and the last motor is used to control the clutch pedal. There are two L angles placed close to the inner sides of the two front seats. Two rods are fixed between these L angles. These rods are used to support the movable motor housing. The motor housing has two servo motors. Just behind these rods there is a small rack and pinion setup, with the rack parallel to the rods. The pinion is controlled by one of the servo motors in the housing, which is used to give the sideward motion of the gear stick. The second motor has its shaft parallel to the rod and it is connected to a rod, which in turn is fixed to the gear stick. This motor can make the gear move in front/backward direction by making quarter of a rotation in a suitable direction (i.e. clockwise or anticlockwise). When the driver starts the car and turn on the module, the servo motor for controlling the clutch rotates in such a way that the clutch is fully engaged. The servo motor controlling the rack and pinion moves in such a way that the gear sticks moves to the left end of the neutral line. The servo motor controlling the rod connected to the moves a quarter of a rotation in clockwise direction so that the first gear is engaged. When the driver begins to accelerate the clutch is slowly disengaged, and the vehicle starts to move in the front. The speed of the car is measured by fixing the DRV5013 on to non-rotary brake drum, a strong permanent is fixed onto the wheel alloy such that its North Pole faces inwards, another strong permanent magnet is fixed on the alloy radially opposite to the first magnet, the second magnet is fixed in such a way that its south pole faces inwards. When the car begins to move the DRV5013 is alternatively exposed to the north and south poles of the magnets. Thus the DRV5013 produces a square wave in its output pin. The time period of this square wave tells us the time the wheel takes to complete one revolution. The time period is measured using the capture mode of the MSP439F5529 microcontroller. Using the time take for completion of one revolution, the speed of the vehicle can be calculated. When the speed of the car goes out of range for the particular gear that is engaged, the module applies the clutch and the appropriate gear is applied.

Constraints:

1) The size of the components should not exceed the space between the two front seats
2) The components should not adversely affect the aesthetics inside the car. In order to tackle this issue, to plan to keep the module beneath the gear stick in the final product.
3) The moving parts should not make any noise. The noise caused by moving parts can cause an unpleasant environment inside the car.

3. Implementation

A. Hardware Implementation

The demonstration of the actual product in a moving car is quite difficult. Hence for the demonstration and the production of the video, we have created a setup that simulates the pedals and the gear stick of a real car. The simulator consists of a box with the gear stick in it. This gear stick can move in all directions the gear stick in a real car can. We use a sewing machine motor connected to a small wheel, and the sewing machine pedal to simulate the actual wheel and accelerator pedal respectively. Initially we had decided to go ahead with stepper motors to control the gear stick and clutch. After some research we discovered that the stepper motor is not suitable for high torque and high RPM. This would lead to a non-real time effect in the working of the product, i.e. the gear and clutch will be engaged slowly. Hence we choose to use the servo motors which can drive at higher torque and higher RPM than stepper motors [1].

B. Software Implementation

TI Code Composer Studio 6.0 was used to develop the code for the module.

```
initiliase the baud as 9600;
attach servo motor for front-back movement to pin 10;
attatch servo motor for clutch movement to pin 11;
fix rpm reading from pin 7;
initiliase the current gear to 0;
loop{
 if (rpm falls within a range and correct gear is
 not engaged){
 engage the clutch;
 change the gear using front_back and
 and side servo;
 print the gear change to LCD;
 disengage clutch;
 }
 else
 continue reading frequency from 7;
 and loop till a interrupt occurs;
}
```

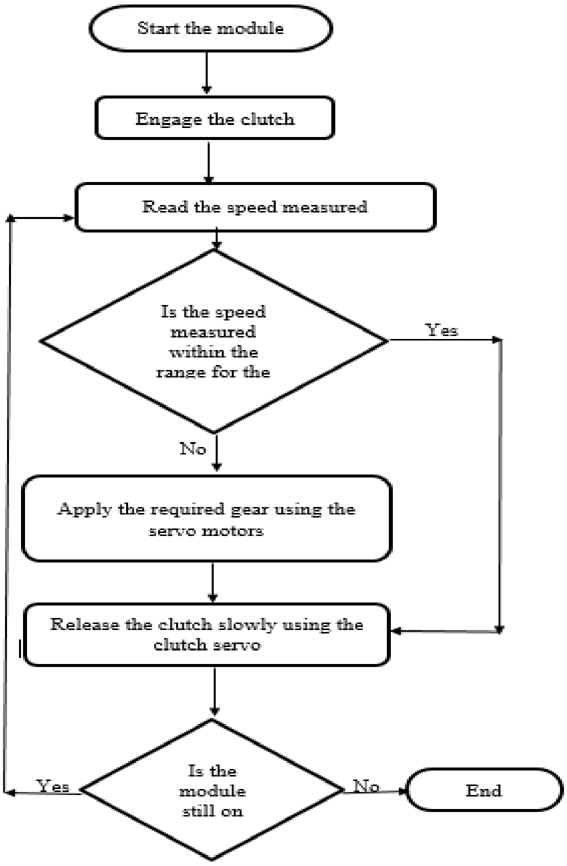
Start the module
Engage the clutch
Read the speed measured
Is the speed measured within the range for the
Yes
No
Apply the required gear using the servo motors
Release the clutch slowly using the clutch servo
Is the module still on
Yes
No
End

Figure 2: Flow chart of the automated transmission process

4. Results

The prototype has been tested using a standard AC motor to simulate the motor of a car. Real time simulation will be difficult and impractical due to safety considerations. The gears were correctly engaged at the correct RPM and (velocity readings). The module has been designed keeping in mind its use in a diesel car which has high low end torque.

5. Conclusion

The goal of the project is to design, test and validate the add-on for a hatchback in the market. The first phase of the phase aims to looks to develop the add-on for a mass market hatchback car (Maruti Alto 800 or Hyundai Santro) in the country. We aim to test the add-on on a real car by the end of the current year (2015) and possible hit the market by 2017. By then worsening traffic conditions would make such a device necessity in India. The device must pass ISO 8001 norms and be complaint with local emission and safety standards. Also, car manufacturers can license the product from us as an after sales accessory. We aim to provide the device free of cost to senior citizens and pregnant women. Currently, a working prototype has been developed in house. However, it has yet to undergo field trials. The code needs to be optimized for practical conditions. The torque of the servo motor and the gear switcher has to be optimized to prevent the device from damaging the gear shift or the clutch. Also a control switch has to be configured. The inputs from the ADC need to be optimized to reflect real world conditions. Even though prototype is robust it has to be strong enough to withstand shocks and have a low failure rate. The device has been designed in such a way such that it can be easily maintained and repaired. The major strength of the project and such a device is that it will be the only one to hit the market. Currently, nothing like this exists in the consumer automobile accessory market. Coupled with its low cost and simplicity and ease of maintenance it promises to be take the automobile industry in the developing countries by storm. Buyers are looking for a cheap diesel automatic hatch in the country, but with the cheapest one starting at 6 lakhs plus they are relegating to a manual hatch. However as stated, current traffic conditions and market analysis show that there is a large market for such a device. It will enable even the most utilitarian of cars to achieve automatic transmission option. In short there is a large untapped market for a Low cost add-on to provide automatic transmission.

The project has its share of limitations. Firstly, the device needs to miniaturize to be installed near the gear box of the car. The size of the components should not exceed the space between the two front seats. Secondly, the device will be added externally. It needs to be aesthetically pleasing enough such that it doesn’t obstruct with cabin dynamics. In order to tackle this issue, we are planning to keep the module beneath the gear stick in the final product. Thirdly, the drivetrain and steering column should not be damaged during the aftermarket installation of the add-on. Secondly, components need to be installed such that they can handle practical driving conditions and be resistant to wear and tear. Lastly, the moving parts should not make any noise. The noise caused by moving parts can cause an unpleasant environment inside the car. The module at this staged has been designed for diesel cars. Their high torque at low RPM’s make them ideal for such automation.

6. Acknowledgments

We would like to thank our Professor and Guide Dr. P.G. Shah for his valuable inputs and his guidance during the course of this project. We are also indebted to Dr. HV Kumarswamy for his moral and technical support and for sanctioning the project and helping us in the necessary logistics.

References

[1] "statista.com," January 2011. [Online]. Available: http://www.statista.com/statistics/204092/automatic-transmission-forecast-for-car-production-worldwide/.

[2] K. Angre, "ndtv.com," NDTV, 13 February 2014. [Online]. Available: http://www.ndtv.com/delhi-news/delhi-1400-cars-added-to-citys-roads-every-day-550647.

[3] Ravi, "http://indiatransportportal.com/," 2014. [Online]. Available: http://indiatransportportal.com/amt-vehicles-india-29211.

APPENDIX A

<u>Code for the Module:</u>

```
#include <Servo.h>
#include<energia.h>
#include <LiquidCrystal.h>
Servo
myservo_side,myservo_front_back,myservo_clut
ch;
LiquidCrystal  lcd(P2_2,  P2_3,  P2_4,  P3_0,
P3_1, P3_2, P3_3, P3_4, P3_5, P3_6, P3_7
int pos = 0;
unsigned long frequency;
float duration;
int current_gear,counter,clutch;
byte up_arrow[8] = { // used for up arrow
display
B00100,
B01110,
B10101,
B00100,
B00100,
B00100,
B00100,
B00100
};
byte  down_arrow[8]  =  {  //  used  for  down
arrow display
B00100,
B00100,
B00100,
B00100,
B00100,
B10101,
B01110,
B00100,
};
```

```
void setup()
{
myservo_side.attach(9);
myservo_front_back.attach(10);
myservo_clutch.attach(11);
pos=0;
myservo_side.write(pos);
delay(2000);
pos=95;
myservo_front_back.write(pos);
delay(2000);
pos=0;
myservo_clutch.write(pos);
delay(2000);
lcd.begin(16, 4);
current_gear=0;
}
void    print_car_details(int    speed_car,int
gear)
{
if(gear==1)
{
lcd.setCursor(0,0);
lcd.print("SHIFT");
lcd.setCursor(0,1);
lcd.print("Mode:Auto 1 3 5");
lcd.setCursor(0,2);
lcd.print("Speed:0-15Kmph ++++");
lcd.setCursor(15,2);
lcd.write(byte(0));
String speed_string=String(speed_car);
lcd.createChar(0, up_arrow);
lcd.setCursor(6,2);
lcd.print(speed_string);
}
if(gear==2)
{
lcd.setCursor(0,0);
lcd.print("SHIFT");
lcd.setCursor(0,1);
lcd.print("Mode:Auto 1 3 5");
lcd.setCursor(0,2);
lcd.print("Speed:15-25Kmph ++++");
String speed_string=String(speed_car);
lcd.setCursor(6,2);
lcd.print(speed_string);
lcd.createChar(0, down_arrow);
lcd.setCursor(15,2);
lcd.write(byte(0));
}
if(gear==3)
{
lcd.setCursor(0,0);
lcd.print("SHIFT");
lcd.setCursor(0,1);
lcd.print("Mode:Auto 1 3 5");
lcd.setCursor(0,2);
lcd.print("Speed:25-35Kmph++ ++");
String speed_string=String(speed_car);
lcd.setCursor(6,2);
lcd.print(speed_string);
lcd.createChar(0, up_arrow);
lcd.setCursor(17,2);
lcd.write(byte(0));
}
if(gear==4)
{
lcd.setCursor(0,0);
lcd.print("SHIFT");
lcd.createChar(0, down_arrow);
lcd.setCursor(0,1);
lcd.print("Mode:Auto 1 3 5");
lcd.setCursor(0,2);
lcd.print("Speed:35-50Kmph++ ++");
String speed_string=String(speed_car);
lcd.setCursor(6,2);
lcd.print(speed_string);
lcd.setCursor(17,2);
lcd.write(byte(0));
}
if(gear==5)
{
lcd.setCursor(0,0);
lcd.print("SHIFT");
lcd.createChar(0, up_arrow);
lcd.setCursor(0,1);
lcd.print("Mode:Auto 1 3 5");
lcd.setCursor(0,2);
lcd.print("Speed:50-80Kmph ++++");
String speed_string=String(speed_car);
lcd.setCursor(6,2);
lcd.print(speed_string);
lcd.setCursor(19,2);
lcd.write(byte(0));
}
lcd.setCursor(0,3);
lcd.print("Gear: 2 4 R");
String gear_string=String(gear);
lcd.setCursor(5,3);
lcd.print(gear_string);
}
void loop()
{
int speed_car=0;
delay(25);
{
current_gear=1;
print_car_details(speed_car,current_gear);
delay(10);
for(pos  =  0;  pos  <  90;  pos  +=  1)  //apply
clutch
{
delay(20);
myservo_clutch.write(pos);
delay(15);
}
pos=0;
myservo_side.write(pos);
delay(15);
for(pos  =  90;  pos  <=  115;  pos+=1)  //push
forward
{
myservo_front_back.write(pos);
delay(15);
```

```
}
for(pos = 90; pos > 0; pos -= 1) //release
clutch
{
myservo_clutch.write(pos);
delay(20);
}
}
delay(40);
{
current_gear=2;
print_car_details(speed_car,current_gear);
for(pos = 0; pos < 90; pos += 1) // apply
clutch
{
myservo_clutch.write(pos);
delay(15);
}
for(pos = 115; pos >=85; pos -= 1) //bring
stick back
{
myservo_front_back.write(pos);
delay(15);
}
for(pos = 90; pos > 0; pos -= 1) //release
clutch
{
myservo_clutch.write(pos);
delay(15);
}
}
//gear 3
delay(40);
{
current_gear=3;
print_car_details(speed_car,current_gear);
delay(20);
for(pos = 0; pos < 90; pos += 1) // apply
clutch
{
myservo_clutch.write(pos);
delay(15);
}
for(pos = 85; pos<= 95; pos += 1)
{
myservo_front_back.write(pos);
delay(15);
}
for(pos = 0; pos <=90; pos+=1)
{
myservo_side.write(pos);
delay(15);
}
for(pos = 95; pos <= 115; pos += 1)
{
myservo_front_back.write(pos);
delay(15);
}
for(pos = 90; pos > 0; pos -= 1) //release
clutch
{
clutch=0;
myservo_clutch.write(pos);
delay(15);
}
}
//gear4
delay(40);
{
current_gear=4;
print_car_details(speed_car,current_gear);
delay(20);
for(pos = 0; pos < 90; pos += 1) // apply
clutch
{
clutch=1;
myservo_clutch.write(pos);
delay(15);
}
for(pos = 115; pos >= 85; pos -= 1)
{
myservo_front_back.write(pos);
delay(15);
}
for(pos = 90; pos > 0; pos -= 1) //release
clutch
{
clutch=0;
myservo_clutch.write(pos);
delay(15);
}
}
//gear 5
delay(40);
{
current_gear=5;
print_car_details(speed_car,current_gear);
delay(20);
for(pos = 0; pos < 90; pos += 1) // apply
clutch
{
clutch=1;
myservo_clutch.write(pos);
delay(15);
}
for(pos = 85; pos <= 95; pos += 1)
{
myservo_front_back.write(pos);
delay(15);
}
for(pos = 90; pos <= 180; pos+=1)
{
myservo_side.write(pos);
delay(15);
}
for(pos = 95; pos <= 115; pos += 1)
{
myservo_front_back.write(pos);
delay(15);
}
for(pos = 90; pos > 0; pos -= 1) //release
clutch
{
clutch=0;
```

```
myservo_clutch.write(pos);
delay(15);
}
}
//gear 5
delay(40);
{
print_car_details(speed_car,current_gear);
delay(20);
for(pos = 0; pos < 90; pos += 1) // apply
clutch
{
clutch=1;
myservo_clutch.write(pos);
delay(15);
}
for(pos = 95; pos >= 90; pos -= 1)
{
myservo_front_back.write(pos);
delay(15);
}
for(pos = 180; pos >=0; pos-=1)
{
myservo_side.write(pos);
delay(15);
}
for(pos = 90; pos > 0; pos -= 1) //release
clutch
{
clutch=0;
myservo_clutch.write(pos);
delay(15);
}
}
}
```