Automatic engine locking system for drunken drivers

K. Murali Chandra Babu¹, K. Spandana², E. Pranitha³ ^{1,2,3} Vignan Institute of Technology and Science. Deshmukhi, Telangana (E-mail: muralichandra454@gmail.com)

Abstract— Drunken Driving is the major cause for the accidents. Drunken drivers are not in stable condition which leads to reckless driving and make inconvenience to others. To avoid these accidents a new design has developed when an alcohol is detected then automatically the vehicle engine turned off. The proposed design is implemented using AT89S52 microcontroller. The controller is programmed using PIC C compiler and alcohol is detected by using alcohol detector sensor. This design is used for safety of peoples life.

Keywords— AT89S52 controller, PIC c compiler, Alcohol detector sensor

I. INTRODUCTION

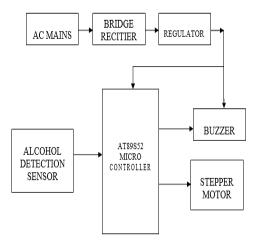
Now-a-days most of the accidents occurred due to alcohol consumption by the driver. The drivers who drank alcohol will be in unstable which cause rash driving which leads loss of valuable life of himself snd other people who moves on roads. Detection of alcohol can be done by stopping the vehicle and using a breath analyzer. The breath analyzer will detect the alcohol content from driver.

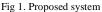
To avert these accidents new design is proposed which will turned off the engine when the alcohol is detected from driver. This auto engine off system will be safety for the drunken drivers and the people used the road. This system detects the alcohol in the vehicle then engine will be locked automatically which will reduce the road accidents due to drunken driving.

II. BLOCK DIAGRAM

The proposed system is illustrated in fig 1 which consists of At89S52 micro controller, Alcohol detection sensor, and buzzer, motor.

Initially all the devices are initiated, the alcohol detection sensor will measure the levels alcohol and sends the data to the micro controller. The micro controller will compare with the predefined limit if alcohol limit below the level then the vehicle will be run without any interruption else if alcohol limit consumed by driver exceeds the limit then the buzzer will be blown and LED will glow to the indication of other people that vehicle will stop.





As soon as the alcohol is detected, it is not favorable to stop the vehicle when it is in accelerated speed. So here we are using a stepper motor which decreases the speed over a time and finally the vehicle gets stopped. Fig 2 depicts the flow chart of the system

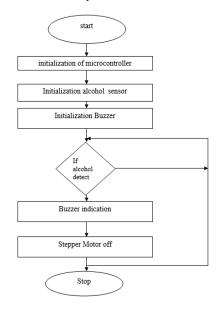


Fig 2 Flow chart of design

IJRECE VOL. 7 ISSUE 2 (APRIL- JUNE 2019)

III. IMPLEMENTATION

The design is implemented in PIC C compiler, Proteus 7 where the controller is programmed in C in PIC C compiler and dumped on to it by Proteus 7 tool.

In PIC C compiler the steps need to compile the C code is

Open a PIC C compiler and create a new source file which is sown in fig 3

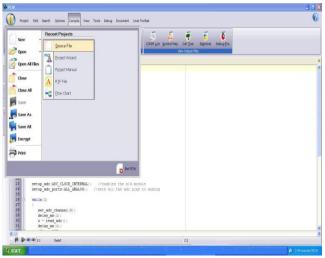


Fig 3: creating new file using PIC C compiler

After creating a new source file write the logic C code and go to Compile and check for the errors the compilation process shown in fig 4

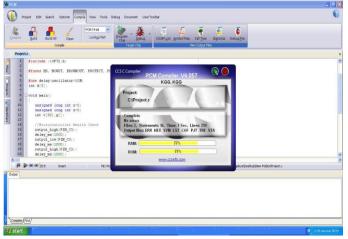


Fig 4: compiling for errors and warnings

Fig 5 shows after compiling the file .hex file is generated in the folder which is used to dump file in controller

ISSN: 2393-9028 (PRINT) | ISSN: 2348-2281 (ONLINE)

	H
110 DR.Fie	
7.KB ESVM File	
1 KB C Source File	
210 STARIe	
2 KB C Output File	
THE COUDULTIN	
	25.118 🔮 My Computer
	7.18 ESHH File 1.18 C Object His 5.18 C Output His 1.18 C Source His

Fig 5:.hex file in the folder

To dump the .hex file in controller now Open Proteus 7 and click on IS1S6. Then select AT89S52 microcontroller. Then click on tools and select the check communication to make the connection between the PC and controller the process is shown in fig 6.

		Programm	er Too	ls View	Help					
Midrange/St	andard Cor	figuration		Enable Cos	de Protect		Ctrl+P			
Device	Hermon	orted Part (0-4	Enable Dat						
				OSCCAL						
User ID s:	FF FF FF	FF						1		
Checksum	FC00			Target VDD Source +						
				Display Uni	implemente	d Config Bi	ts	•		
PICkit 2 no	t found.	Check I	USI	Calibrate V	DD & Set U	init ID		DCI		
use Tools	->Check	Commu	nice	Use VPP Fi	rst Program	Entry		PL		
Contraction of the		Network Concerns		Lice IVP Pr	ogram Entr					
				Fast Progr				2.5		
Read	Write	Vesty		-				-		
Program M	emory			LIART Tool						
Enabled	Hex On	y 😢	Se	Logic Tool.						
000	SFFF	SFFF		Check Com	munication			TFT	10	
008	SFFF	SFFF	3	Troublesho				FFF		
010	SFFF	SFFF	3	The second se				FFF		
018	SFFF	SFFF	1	THE GO AND	13-10-0 P-11-P			FFF		
020	SFFF	SFFF	SFFF		SFFF	SFFF	SFFF	SFFF		
028	SFFF	SFFF	SFFF	SFFF	SFFF	SFFF	SFFF	SFFF		
030	SFFF	SFFF	3FFF	3FFF	SFFF	3FFF	SFFF	SFFF		
	OFFF	SFFF	SFFF	3FFF	SFFF	SFFF	SFFF	SFFF		
038		SFFF	SFFF	SFFF	SFFF	SFFF	3FFF	SFFF		
040	SFFF				SFFF	SFFF	SFFF	3FFF		
0.40 0.48	SFFF	SFFF	SFFF	SFFF						
040			3FFF 3FFF 3FFF	SFFF SFFF	SFFF	SFFF	SFFF	SFFF		

Fig 6: checking communications

After finding the communication then it displays PIC kit 2 found and connected in the window shown in fig 7

							BandGap:		
PICkit 2 fou	ind and	I connect	ed.			5	Mic	ROCH	-118
						VD	D PICkit 2		
Read	Write	Verity	Eras	se Bi	ank Check	5 8	On /MCLR	2.5	0
Program Me	mory		-			-			
Enabled	Hex Or	16 ~	Source:	None (En	pty/Erased	0			
000	SFFF	DFFF	OFFF	OFFF	SFFT	SFFF	SFFF	SFFF	~
008	SFFF	SFFF	SFFF	3FFF	SFFF	SFFF	SFFF	SFFF	
	SFFF	SFFF	3FFF.	SFFF	3FFF	SFFF	3FFF	3FFF	
018	SFFF	SFFF	SFFF	SFFF	SFFF	3FFF	3FFF	3FFF	
	SFFF	3FFF	3FFF	SFFF	SFFF	3FFF	SFFF	3FFF	
028	SFFF	SFFF	SFFF	SFFF	SFFF	SFFF	SFFF	SFFF	
	SFFF	OFFF	3FFF	SFFF	SFFF	SFFF	SFFF	SFFF	
0.38	SFFF	SFFF	SFFF	SFFF	SFFF	SFFF	SFFF	SFFF	
0.40	SFFF	SFFF	SFFF	OFFF	SFFF	JFFF	SFFF	3FFF	
0.48	SFFF	3FFF	3FFF	SFFF	SFFF	SFFF	SFFF	SFFF	
	SFFF	SFFF	3FFF	SFFF	SFFF	SFFF	3FFF	SFFF	
0.58	3FFF	3FFF	SFFF	SFFF	3FFF	SFFF	SFFF	SFFF	
EEPROM D									

Fig 7: connecting the dumper to microcontroller

IJRECE VOL. 7 ISSUE 2 (APRIL- JUNE 2019)

After making the communication then Import the .hex file from saved location by selecting File option and clicking on 'Import Hex' as shown in below window.

e Di	evice Family	Programme	er Tools	View	Help				
Imp	ort Hex				Ctrl+I				
Exp	ort Hex				Ctrl+E				
						E	BandGap:		
						5	Mic	ROCH	-111
Exit					Ctrl+Q	VDI) PICkit 2		
							On	5.0	-
Read	Write	Verify	Eras	e Bl	ank Check		/MCLR		
rogra	m Memory								
Enat	Hex Or	nly 🔽	Source:	None (En	npty/Erased))			
000	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	1
008	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	-
010	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	
018	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	
020	3FFF	SFFF	3FFF	3FFF	SFFF	3FFF	3FFF	SFFF	
028	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	
030	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	
038	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	
040	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	
048	3FFF	SFFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	
050	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	
058	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	Y
EPRC	M Data								
Enat	oled Hex Or	nly 💌					Lª	uto Import I Write Dev	Hex ice
								lead Devic xport Hex f	
							P	ICkit"	2

Fig 8: importing .hex file

After 'Import Hex' option browse the location of program and click the 'prog.hex' and click on 'open' for dumping the program into the microcontroller then it displays the programmed and successful in the window

le Device	Family	Programmer	Tools	View	Help				
fidrange/St	andard Co	nfiguration							
Device:	PIC16F	72		Config	uration: 00	42			
Iser IDs:	FF FF F			Code F	Protect				
Checksum	0041			OSCO			B and G ap:		
rogramm	ing Suc	cessful.				5	Mic	ROCH	11P
						D VD	D PICkit 2		
				-			On	5.0	-
Read	Write	Verify	Erase	BI	ank Check		/MCLR	0.0	¥
Program M	emoru								
Enabled	Hex Or	alu 🗸	Source:						
	-						11		-
000	3000	A800	2817	0000	3027	0084	0800	1903	-
008	2816 307B	3006	00A1 0BA0	0140	OBAO	2800	OBA1	280B	
010	307B 301F	00A0 0583	1025	2812	1683	2809	3400	1407	
020	1683	141F	1025 149F	151F	30FF	1283	1283 00A5	11407	
020	1683	1683	1491	1283	1587	3064	DOAS	2004	
028	1145	0825	1683	0087	1283	3064	3004	2004	
030	30FA	0025 00A7	2004		2838	2827	0063	SFFF	
038	SUFA	3FFF	2004 [0: 3FFF	3A DAG	2030 3FFF	3FFF	3FFF	SFFF	
040	SFFF	SFFF	SFFF	SFFF	SFFF	SFFF	SFFF	SFFF	
050	SFFF	SFFF	SFFF	SFFF	SFFF	SFFF	SFFF	SFFF	
058	SFFF	SFFF	SFFF	SFFF	SFFF	SFFF	SFFF	SFFF	v
									-
EPROM I	Data Hex Di	ly 💌						to Import H Write Devi	
								ead Device port Hex F	
							PI	Ckit [~]	2

Fig 9: successfully dumped into microcontroller

IV. RESULTS

The fig 9 illustrates the prototyped design of the auto engine off when alcohol is detected where the Alcohol sensor, LCD display buzzer and Motor is connected to microcontroller.

ISSN: 2393-9028 (PRINT) | ISSN: 2348-2281 (ONLINE)



Fig 9. Connections of proposed design

When the vehicle is started. The sensor will check for the alcohol if there is no alcohol is detected then the motor runs and it displays a Vehicle started and alcohol is clear in LCD display which is illustrated in fig 10. The lights in driver circuit indicate the running motor



Fig 10 when alcohol is clear

When the alcohol is detected above the limit then buzzer will blow and LED is glow to indicate that vehicle is going to stop and Displays Alcohol is detected in LCD display



Fig 11: when alcohol is detected

IJRECE VOL. 7 ISSUE 2 (APRIL- JUNE 2019)

References

- [1] "The 8051 Microcontroller Architecture, Programming & Applications" By Kenneth JAyala.
- [2] "The 8051 Microcontroller & Embedded Systems" by Mohammed Ali Mazidi and Janice GillispieMazidi
- [3] "Linear Integrated Circuits" by D Roy Choudary & ShailJain
- [4] "Electrical Machines" by S KBhattacharya
- [5] WWW.MITEL.DATABOOK.COM
- [6] WWW.ATMEL.DATABOOK.COM
- [7] <u>WWW.KEIL.COM</u>

