VEHICLE OVERHEATED PREVENTION SYSTEM: COOLING FAN FAILURE ALERT SYSTEM

CHAU SHEN LIANG

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering (Hons.) Electrical and Electronic

Lee Kong Chai Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

AUGUST 2016

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : ______________________

Name : Chau Shen Liang

ID No. : 12UEB00843

Date : ______________________

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"VEHICLE OVERHEATED PREVENTION SYSTEM: COOLING FAN FAILURE ALERT SYSTEM"** was prepared by **CHAU SHEN LIANG** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Engineering (Hons.) Electrical and Electronic at Universiti Tunku Abdul Rahman.

Approved by,

Signature : ______________________

Supervisor : Dr. Chew Kuew Wai

Date : ______________________

The copyright of this report belongs to the author under the terms of the copyright Act 1987 as qualified by Intellectual Property Policy of Universiti Tunku Abdul Rahman. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

© 2016,Chau Shen Liang. All right reserved.

VEHICLE OVERHEATED PREVENTION SYSTEM: COOLING FAN FAILURE ALERT SYSTEM

ABSTRACT

The main focus in the project is to study the causes of cooling fan failure in vehicle cooling system and design and built a vehicle's cooling fan failure alert system to prevent vehicle from overheated. The reason why an alert system is needed because most of the vehicle overheated is due to cooling fan failure, a system can be develop to monitor the cooling fan system, and if any parts of the cooling fan fail to operate such as relay, the alert system will alert the driver to replace the malfunction relay. With the invention of such alert system, drivers are able to detect the problems in early that might cause overheat before vehicle overheated. This system monitor the health of the vehicle fan cooling system and tell the drivers if any part of the fan cooling system has failed. The possible causes of cooling fan failure and how heat is generated from engine are also studied in the chapter 2. The method to build the cooling fan failure alert system is explained in chapter 3. A prototype is built and to test the functionality of the system. Chapter 4 shows the result of functionality testing obtained from the current sensor, voltage sensor, coolant temperature sensor (CTS) and Hall Effect tachometer.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

V_{in}	input voltage, volt
V_{out}	output voltage, volt
R	resistor, ohm
I	current, ampere

SI	spark ignition
CI	compression ignition
PIC	peripheral interface controller
RPM	revolutions per minute
CTS	coolant temperature sensor
ADC	analogue digital converter
LED	light-emitting diode
PCM	power train control module
PWM	pulse width modulation
IC	integrated circuit

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

The cooling or radiator fan motor is important in vehicle's cooling system. It takes heat away from engine and dissipates it to the air outside. The cooling fan avoid the engine from overheat so that engine stay in good working condition. Vehicle overheat is a common problem happen for everyone, most of the vehicles that overheat are due to cooling fan failure. Before this happen, the vehicle does not give any signal to the driver that if any part of the sensors or relay fail might lead to overheat. The driver only know the car is overheated when the dashboard show an overheated warming lamp. When a vehicle is overheating, some of the driver might force to stop at the roadside and wait for the vehicle to cool down by filling cool water to the radiator water tank.

The consequence of vehicle overheat is serious, which may cost a great amount of money to repair the engine. The engine may be blow or serious damage when vehicle is overheating. In order to avoid this serious problem, an overheat prevention system is developed to alert driver if any of the parts in the cooling system fail, so that the driver can fix it before it lead to an serious consequences. This alert system will help to monitors the fan cooling system, such as fan voltage, current, speed and also other parts like relay and temperature sensor. This system will alert the driver to change if any parts of the cooling system fail, so that vehicle overheated can be avoided.

1.2 Aims and Objectives

The aim of the research is to study the causes of cooling fan failure in vehicle cooling system and then design and built a vehicle's cooling fan failure alert system to prevent vehicle being overheated.

The purpose of the research is to:

- Investigate the possible failure of the cooling fan system
- Determine the type of cooling fan used in vehicle
- Determine the characteristic of the electric fan
- Design and built a cooling fan failure alert system

CHAPTER 2

LITERATURE REVIEW

2.1 General View of Engine System:

The reciprocating internal combustion engine is the most common engine. Two types of internal combustion engine are familiar, the four stroke engines and the two stroke engines. The cycle consists of four stroke engines as shown in Figure 2.1. The two stoke engines is distinct from the four stroke engines in absence of separate stroke in intake and exhaust. The construction of two stroke engine is simple but due to its poor reducing harmful exhaust emission, eventually become less common. The Otto cycle is refering to 4 stroke cycle and used as a basis comparison for spark ignition (SI) and compression ignition (CI) engines. The ideal Otto cycle is shown in Figure 2.2. The SI engine is sometime refer to engines that used petrol, gasoline or gas. The CI engine is refer to engines that used diesel or oil. For an SI engine, fuel is ignited by spark where a CI engine, fuel is ignited during compression by the rise in temperature and pressure during compression (Stone, 1992).

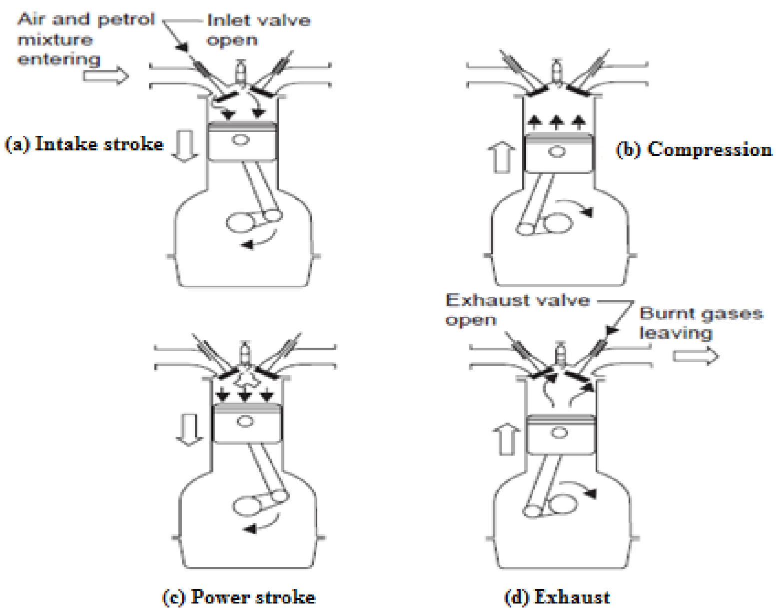


Figure 2.1: The Four Stroke Petrol Engine Cycle (Nuney, 2006) namely (a) Intake stroke, (b) Compression, (c) Power stroke and (d) Exhaust stroke

Intake stroke: The inlet valve will open and the piston descends, the volume of the chamber increases to let air enter in the CI engine. For an air fuel mixture in the SI engines, the volume of the combustion chamber increases to let air enter as fuel is injected.

Compression: All valves are closed in this stroke and the piston comes back up compressing the air or air fuel mixture. The electrical contact is remained opened during this stroke. The pressure is the highest when lesser volume of the combustion chamber, the contact is closed, current will then flows through the plug and ignition begins. Consequently, the pressure, temperature and density are raised as described by the ideal gas law. For a SI engine, the spark plug receives a high voltage pulse and generates spark later which ignites the charge. For a CI engine, the fuel is injected into the combustion chamber, the fuel ignites due to the high temperature. During the compression, no heat is transferred to the air or air fuel mixture.

Power stroke: Initially at this stroke cycle, the electrical contact is opened. The sudden open of the contact creates a spark to ignite the air fuel mixture. The resultant pressure of the combustion gases forces the piston retreating, perform useful work in compressing the air or air fuel mixture. As volume increased, the density, pressure and temperature decreases. When the piston is at its largest volume, the exhaust valve opens. Rapid combustion of the fuel releases heat, and the exhaust gases are produced in the combustion chamber.

Exhaust: The piston comes back up forcing the combustion gases out to the open exhaust valves. When this stroke cycle end, the exhaust valve will close and the intake valve will open, and it start from the intake stoke.

The continuing of burning fuel inside of the engine produced heat in the exhaust system when the combustion gases go out to the open exhaust valves (Karim Nice, 2016).

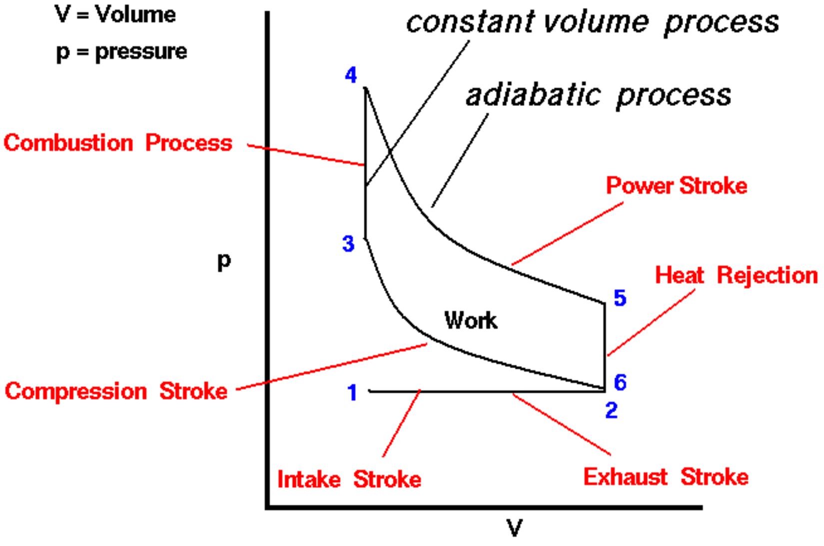


Figure 2.2: Ideal Otto Cycle (NASA, 2015)

During the compression and expansion, it is assumed adiabatic (no heat is transferred to the air or air fuel mixture) (Stone, 1992).

2.2 Overview of Vehicle Cooling System

The engine water cooling system, also known as thermosyphonic circulation was used in many early motor vehicles until the late of 1930s. The water cooling system consists of water jack, radiator, fan, rubber hoses and other fittings. In most early thermosyphonic cooling systems, the fan was driven by belt and pulley. The water circulation rate in the simple thermosyphonic cooling system is proportion to the heat output or the engine load and not the engine speed. As the load increased, the circulation rate also increased due to the formation of small steam bubbles which reducing the density of the coolant being heated (Nunney, 2006).

There are two types of cooling systems, namely liquid cooling system and air cooling system. Some of the old cars, such as original Volkswagen Beetle and Chevrolet Corvair use engines that come with air cooling system. Many modern motorcycles still using air cooling system, but liquid cooling systems are commonly used by modern vehicles and trucks (Charles Ofria, 2016). The typical vehicle liquid cooling system is shown in Figure 2.3.

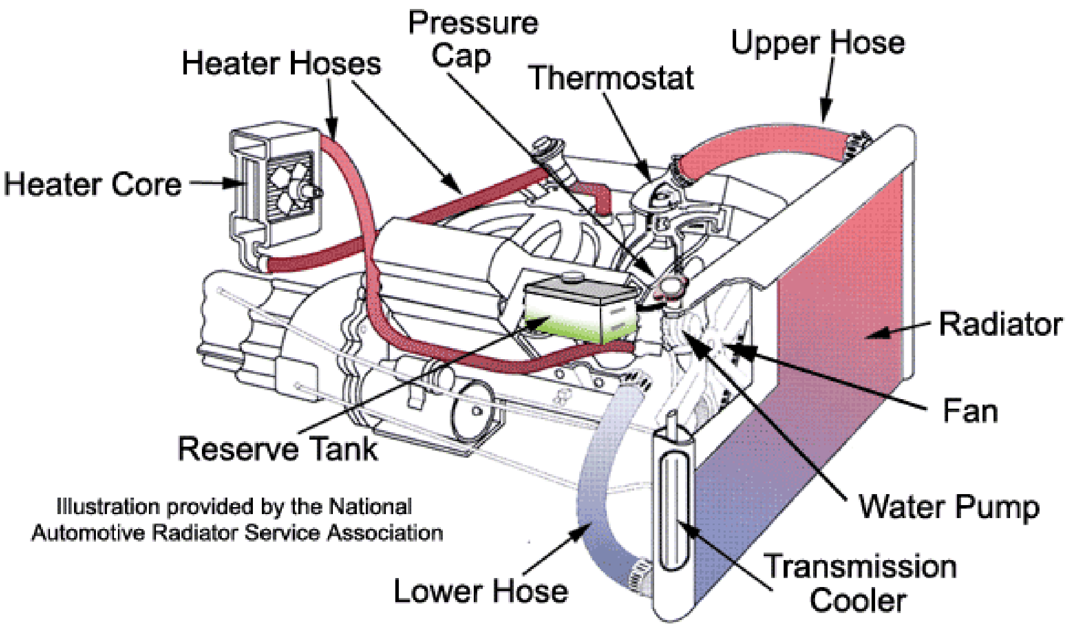


Figure 2.3: Typical Vehicle Cooling System (Charles Ofria, 2016)

The water pump is used to circulate the coolant, while the temperature of the coolant is controlled by a thermostat and a radiator to cool the coolant. In case if the coolant is below a particular temperature value, the thermostat stop the coolant from entering the radiator and bypass back to the engine until the coolant reached preset value, the thermostat will open a valve and allow the coolant enter the radiator. A radiator cap is used to control the pressure in the system, preventing the coolant from boiling and burst the houses if the pressure is too high by increasing the boiling point of the coolant (Karim Nice, 2016). As the coolant flows through the hoses to the engine, it absorbed heat from the engine. The interconnecting hoses are used to direct the coolant from the engine to radiator. The heating system is consider as a secondary cooling system that mirrors the primary cooling system. The coolant will flow to the car's heating system on a cold day to heat up the coolant, hot coolant is used to warm up the vehicle's engine (Charles Ofria, 2016).

The radiator needs a constant flow of air through its core to cool it when a car is not accelerating or moving, a fan is used to help the airflow. While many modern vehicle used an electric fan instead of belt driven fan, the fan is controlled by a temperature sensor or thermostatic switch (How a Car Works, 2016).

2.3 Vehicle Cooling Fan

A cooling fan is usually mounted behind the radiator, the fan direct air through the radiator when a vehicle is stopped or not travelling fast enough to air flow flows through the radiator (Wright, 2016). Without air flow flowing through the radiator (when a vehicle is stopped with the engine is running), the engine temperature will increase and thus overheat take place (Charles Ofria, 2016).

In older application system, a belt driven fan is used for cooling system and connected to the front of water pump, the fan turned on whenever the engine is running. Vehicle overheat is due to lack of air flow flowing through the radiator especially on a hot day or when the vehicle is stopped with the engine running (Charles Ofria, 2016). There is also a fan is controlled by a temperature switch located in radiator or engine in an early system. When the temperature of coolant above the switch's rating, the switch will close and the relay will energize to supply voltage to the fan. The fans stop when coolant temperature decreased to normal operating value (AA1Car, 2016).

Virtually, the modern vehicle cooling system widely used an electric motorized fan. The electric fan does not rely on the engine speed, the fan can turn on for certain time after the engine begin to cool down and turn off when the engine is running at desired temperature. The fan will cycle on and off to maintain the engine temperature at it normal operating value (Stern, 1994).

There are few types of fans, such as fixed drive fan, variable drive fan, off on fan and electrically fan (Nunney, 2006). Many of the modern cars cooling fan used an electric fan for primary or extra cooling. The fan direct air through the radiator when car are not moving to cool down the coolant. There is often also an electric fan on the air conditioning system (Wright, 2016).

Two cooling fans are used in some cars with large radiator, or a separate cooling fan for the air conditioning system. Many vehicles coolant run at a temperature around 200 ^{o}F (Roling Hills, 2016). The engine coolant sensor is used to sense the engine temperature, when the engine temperature is above the normal operating temperature (195 ^{o}F to 215 ^{o}F), the fan will start operate (AA1Car, 2016).

2.3.1 Fixed-drive Fan

In early system, a fan driven by a belt is used for cooling system with the coolant pump and shared the same drive spindle rotated by a V-belt and pulley system connected the crankshaft of the engine, the fan turned on whenever the engine is running (Charles Ofria, 2016). The speed of fan changes with the speed of engine, depend on the speed ratio chosen of the pulley system (Nunney, 2006).

2.3.2 Variable-drive Fan

The variable drive fan can classified into two types, either torque limiting or temperature sensitive type. Some late model cars, the cooling fan can changes its speed as needed. Some fans even have different speed range like high or low speed (AA1Car, 2016). In order to avoid a fan is not driven too much and overcool occurs, a car embody a viscous coupling which allow the fan uncouples if coolant temperature is at normal operating point (How a Car Works, 2016).

The drawback of torque limit fan is it can only run within a fixed speed range, regardless of the needs of cooling. Hence, the cooling may not be enough when car speed is low and engine speed is high. Furthermore, low engine speed with fan is running, the engine need more time to warm up. A temperature sensitive type can solve this problem (Newbold and Bonnick, 2013).

2.3.3 Off-on Drive Fan

The on off drive fan is sometimes used together with the diesel engine of heavy vehicles to fast warm up the engine in order to maintain the operating temperature. Initially the coolant is stay under the preset operating temperature, a thermostat is used to direct air flow from the vehicle's compressed air supply to fan hub, which solved the spring load on the clutch and disconnect the fan from pulley system. The fan is turned off until the temperature of coolant increase more than preset operating value (Nunney, 2006).

2.3.4 Electrically Driven Fan

Many vehicles nowadays used an electric fan for primary or extra cooling. It is common that an electric fan is used in the air conditioning system (Wright, 2016). Electric cooling fans are used instead of a fan driven by engine to save fuel and reduce fan noise, especially when vehicle is running at high speed. A fan driven by engine can consume up to 12 or more horsepower depending on engine speed and cooling load (AA1Car, 2016).

The fan's temperature sensing power circuit controlled the fan to switch on when extra cooling is needed. A separate circuit turns on the fan when the air condition compressor clutch is engaged. In newer vehicles, fan are controlled by vehicle’s computer, fan operation is often regulated by the Power train Control Module (PCM) or a fan control module. A coolant sensor monitors engine temperature, ambient air temperature and send information to the computer to decide if the fan should turn on and energized the fan relay. Vehicles with variable fan speeds, the PCM generates an on off duty signal for the fan motor that control the fan speed (AA1Car, 2016).

2.4 Common Causes of Electric Fan Fail

A electric fan fail may due to over current (electrical overload), the fan motor sometimes tends draw more current when it start to run, this happen very suddenly and will have great impact to the motor. Some devices may use to solve this problem, these devices are usually wired in the circuit and turn off the fan motor when experiencing excessive current (Inc, 2010). In our system design, when current is excessive, there will be an alert system to alert to driver, so if over current occurs because of the fan is stall or stuck, the driver can manually remove the obstacle.

The other possibilities may due to failure of fan relay, bad control circuit, defective temperature sensor and others sensor as well, engine thermostat is stuck open, a defective fan relay, a fan motor going bad, fan wiring connection problem such as blown fuse or short circuit or even a bad fan control module (AA1Car, 2016).

2.5 Problem Statement

The cooling fan motor works in a high temperature environment, it has potential to fail over time. The failing part of cooling fan will then lead to engine start overheating. If the engine is overheating continue without solving it, the engine may blow and eventually spoiled, the vehicle can become inoperable. Repairing the vehicle's engine is much more expensive than replacing a fail cooling fan motor (Mellema, 2016). Not only that, imagine if a vehicle is overheated in a traffic congestion, the vehicle's engine will eventually become inoperable, the vehicle might force to stop at roadside for the engine to cool down, sometimes it is possible worsen the traffic congestion as well.

With the design of cooling fan failure alert system, the alert system will alert the driver if any parts of the cooling system fail, so that the driver know what will happen and giving them to have enough time to look for solution and fix the problem in before it damage the engine. The alert system work in a way that monitors the temperature of coolant, speed (RPM) of the fan, voltage and current of the fan using a PIC microcontroller. The microcontroller will send a signal to the relay to turn of the fan if the temperature of coolant is above the normal operating temperature. The microcontroller also tells the parts such as relay, sensors and fan if they fail to operate.

By using one PIC microcontroller and few sensors, without adding in any extra expensive spare part of the car, a cooling fan failure alert system can be develop. A cheap alert system can avoid the damage of engine which caused by failure of cooling fan, it is worthy to build the alert system. If early precaution is taken, it can prevent the vehicle from overheated, hence, saving a lot of time of travelling as the vehicle does not need to stop at roadside to wait for engine cool down in case the engine is overheated.

CHAPTER 3

METHODOLOGY

3.1 Overview

The basic idea of the project is to design a cooling fan failure alert system to prevent vehicle from overheat. Two units 4 digit 7 segments displays are used to display the RPM of fan, temperature of the coolant. There will also have indicator such as which parts of the hardware (such as relay, fan or coolant temperature sensor) malfunction (if any). Hence, this will alert the driver to look for solution to solve the overheat problem. In this chapter, the method and implementation of hardware will be discussed.

3.2 Design of the Cooling Fan Alert System

In designing the cooling fan alert system, an electric cooling fan from typical sedan vehicle is used in the project. Data will be collected from coolant temperature sensor (CTS) output, cooling fan speed sensor, cooling fan voltage and current. These data will send it to the PIC microcontroller, the PIC microcontroller will process this data through the analogue to digital (ADC) channel and gives appropriate output to the digital port and send signal to relay and 7 segments displays. The 7 segments displays will then display the RPM and temperature of the coolant.

Based on the data collected from coolant temperature sensor (CTS), the PIC microcontroller will process the data send signal to relay and turn on the fan if coolant temperature is exceed the preset value. A high or low signal will send to the relay to trigger the fan to operate. The relay of cooling fan will trigger the cooling fan to run when the temperature exceed 100 ˚C and stop when temperature below 90 ˚C. The RPM, voltage and current of the cooling fan are monitored once the fan is triggered to run.

The alert system will show type of fault if any abnormal value of the RPM, current and voltage of the cooling fan or failure of relay. For instance, if there is no voltage or current flow through the cooling fan after the relay is triggered, we can conclude that is relay failure. If the current slowing through the cooling fan is higher than typical operating current, we can conclude that the cooling fan is either stall or stuck or the motor of the cooling fan is damaged. Another possibility is when the RPM of cooling fan is low, there is possible the cooling fan's motor is going bad and is time to replace a new fan motor. If the engine of the vehicle is turned on after sometimes, the CTS show the temperature value below the operating temperature, this show that the CTS is failed to operate.

3.3 Hardware

3.3.1 Microcontroller PIC16F877A

The PIC microcontroller is a 8 bits based microcontroller which has 35 single word instructions. It consists of 40 pins as shown in Figure 3.1, feature 2 Comparators, 8 channels of 10 bits Analog-to-Digital (ADC) converter. Since we have only 4 sensors, the 8 channels of ADC provide enough channel and yet not all the sensor needs ADC. The ADC is needed for the PIC to process data send from the sensor since PIC is only work with digital input. The Figure 3.1 shows the pin diagram of the PIC16F877A. The UIC00B show in Figure 3.2 is used to load the program into PIC16F877A.

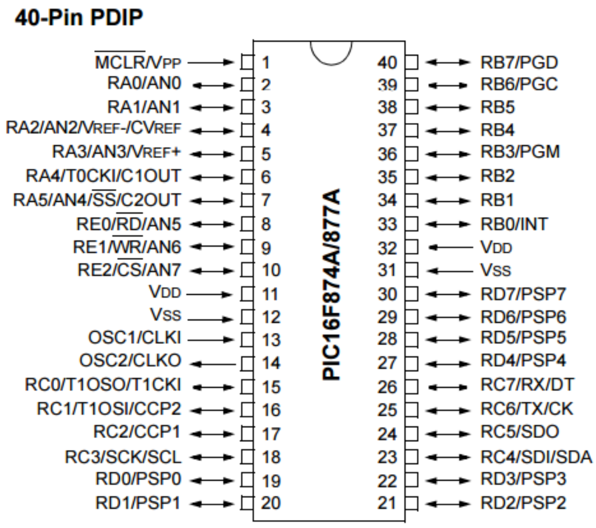


Figure 3.1: Pin Diagram of PIC16F877A

Figure 3.2: UIC00B of USB ICSP PIC Programmer

3.3.2 Voltage detector Circuit

A voltage divider circuit consists of two resistors in series which are 1 MΩ and 250 KΩ as shown in Figure 3.3. The circuit act as a voltage sensor to measure the voltage of the cooling fan. The circuit is expected to measure voltage up to 24 V. The output of the voltage divider is designed within the PIC microcontroller operating voltage range which is expected to be within 5 V for the output. If input voltage, V_{in} is increased by one volt, the output voltage, V_{out}, will increase by 0.2 V, this can be calculated from the formula.

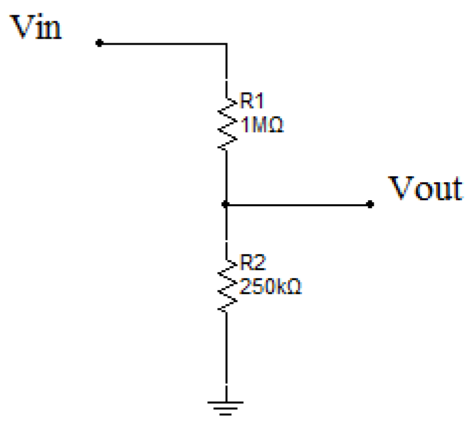


Figure 3.3: Voltage Divider Circuit

$$V_{out} = V_{in}\left(\frac{R_2}{R_1+R_2}\right) \tag{3.1}$$

where

V_{out} = Output Voltage, V

V_{in} = Input Voltage, V

$R_{1/2}$ = Resistor, Ω

$$I = \frac{V_{in}}{(R_1+R_2)} \tag{3.2}$$

where

V_{in} = Input Voltage, V

$R_{1/2}$ = Resistor, Ω

I = Current, A

The equation of 3.1 and 3.2 shows the formula to calculate the output voltage and total current flowing throughout the circuit respectively. The total resistance is 1.25 MΩ. The current flowing to the microcontroller must be set less than 25 mA otherwise it will damage the microcontroller.

3.3.3 Hall Effect Current Sensor

Allegro ACS712ELCTR-30A-T fully integrated, hall-effect-based linear current sensor IC is used. The sensor is able to sense high current flowing through the conductor of cooling fan. Furthermore, the current sensor provides precise and cheap solution for AC and DC current sensing (MicroSystems, 2016). The IC used copper as conduction path which has internal resistance of 1.2 mΩ which then providing low power loss and a precise, low-offset, linear Hall Sensor circuit which has nearly zero magnetic hysteresis (MicroSystems, 2016). The high temperature operating range from -40 ˚C to 150 ˚C which made it suitable for high temperature working condition since this sensor will be placed at the engine compartment which is always performs work at high temperature environment. Figure 3.4, 3.5 and 3.6 show the pin-out diagram, ACS712ELCTR-30A-T IC and typical application circuit respectively.

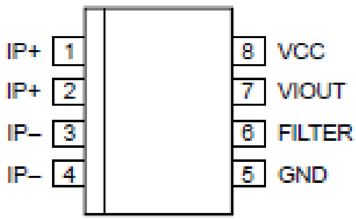


Figure 3.4: Pin-out Diagram of ACS712ELCTR-30A-T

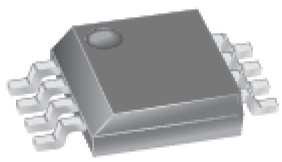


Figure 3.5: ACS714ELCTR-30A-T IC

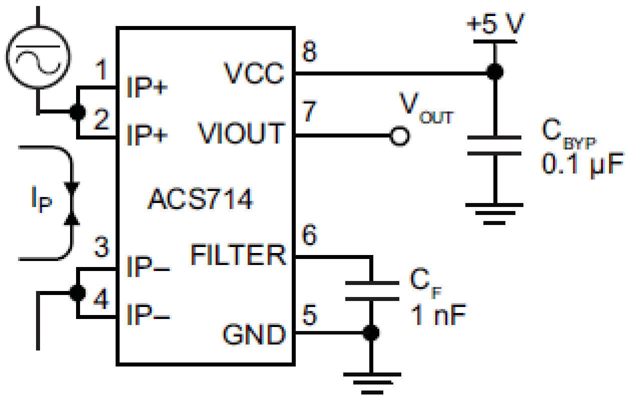


Figure 3.6: Typical application circuit of ACS712ELCTR-30A-T

3.3.4 Hall Effect Tachometer

Allegro A1220LUA-T Chopper Stabilized Precision Hall Effect Latches sensor is a magnetic digital position sensor IC used to measure the RPM of the cooling fan. It has wide voltage operating range from 3.0 V to 24 V, high temperature performance which rating from -40 °C to 150 °C and a package type UA is a three pins ultra mini size for through hole mounting which makes it easily fit into pack place (MicroSystems, 2016).

The sensor is activated by the magnetic field. When a south pole of the permanent magnet approaching the active area of the sensor, it turns the output on and a north pole of the permanent magnet turns the output off. The output voltage of sensor is equal to the power supply when the switch is off (when approaching north pole of permanent magnet). As the south pole of the permanent magnet turned on the sensor, the output voltage will then approaching 0 V. Once the output is turned on, it send a low signal to the microcontroller due to low voltage, and it send a high signal to the microcontroller once it turned off. The generated pulse can be like a PWM signal which then sent to PIC microcontroller to calculate the speed or RPM of the cooling fan. Figure 3.7 and 3.8 show the operating of the A1220LUA-T sensor and typical three-wire circuit respectively (MicroSystems, 2016).

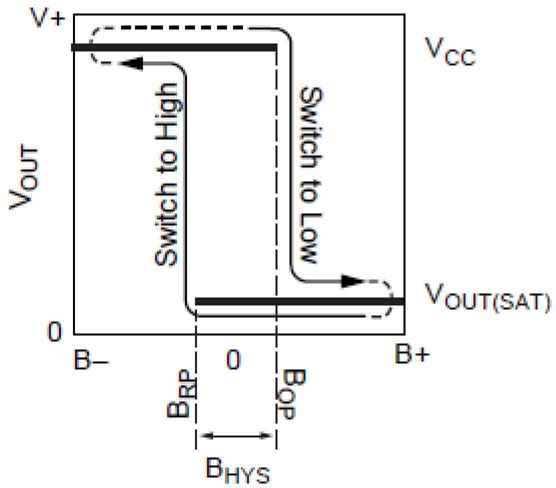


Figure 3.7: Operating of A1220LUA-T (B+ direction indicates increasing south polarity magnetic field strength and B- direction decreasing south polarity field strength with increasing north polarity)

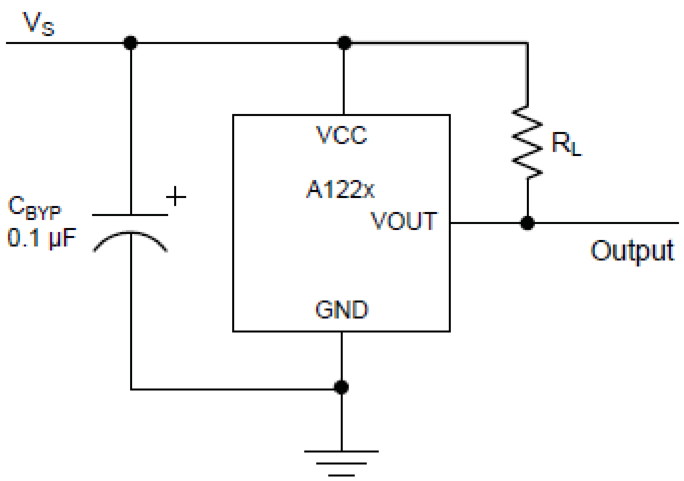


Figure 3.8: Typical three-wire application circuit of A1220LUA-T

3.3.5 Coolant Temperature Sensor (CTS)

Coolant temperature sensor (CTS) is used to measure the temperature of the coolant. A Negative Temperature Coefficient type of CTS is considered in this application, this mean that resistance will decrease as the temperature increased. The resistance of the CTS varies with temperature. The output voltage of the CTS is used to tell the temperature of coolant to the microcontroller. Figure 3.9 and 3.10 show the CTS circuit and CTS respectively.

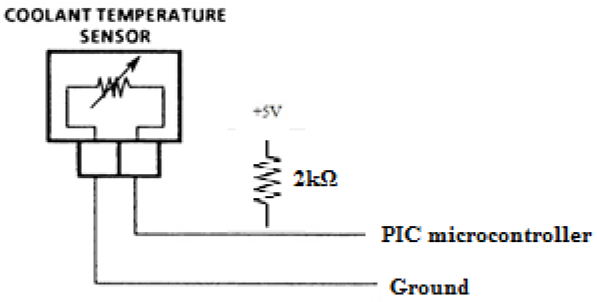


Figure 3.9: CTS circuit.

Figure 3.10: CTS (Autozone, 2015)

3.3.6 Cooling Fan Relay Trigger Circuit

A simple circuit is designed to trigger the relay of the cooling fan. When PIC microcontroller sends a high signal to the relay, the relay will be triggered and the cooling fan is turned on. Figure 3.11 show the relay trigger circuit.

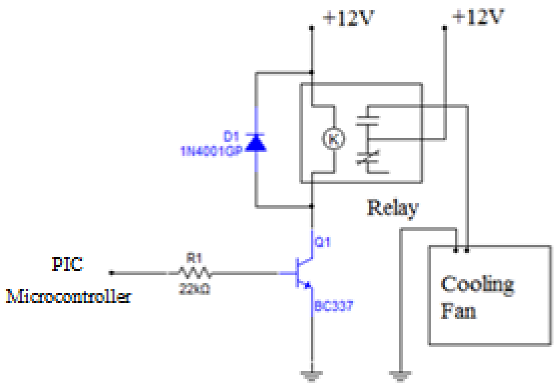


Figure 3.11: Relay trigger circuit

3.3.7 4-Digit 7-Segment Display

Two 7FR5641AS of 4-Digit 7-Segment is used, one is used for display RPM, and the another one is used to display temperature. This is a basic, 4-digit 7-segment display. It has a common cathode. The display features one decimal point per digit, and individually controllable apostrophe and colon points.

The LEDs have a forward voltage of maximum 2.5 V DC and a maximum forward current of 150 mA. Figure 4.12 show the 4-Digit 7-Segment Display of 7FR5641AS and Figure 4.13 show circuit diagram of 7FR5641AS.

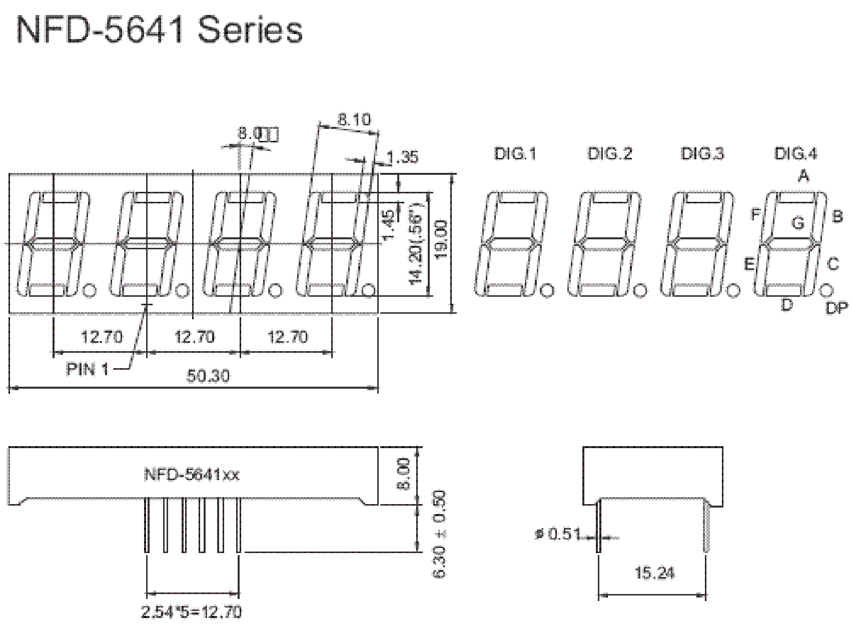


Figure 3.12: 4-Digit 7-Segment Display of 7FR5641AS

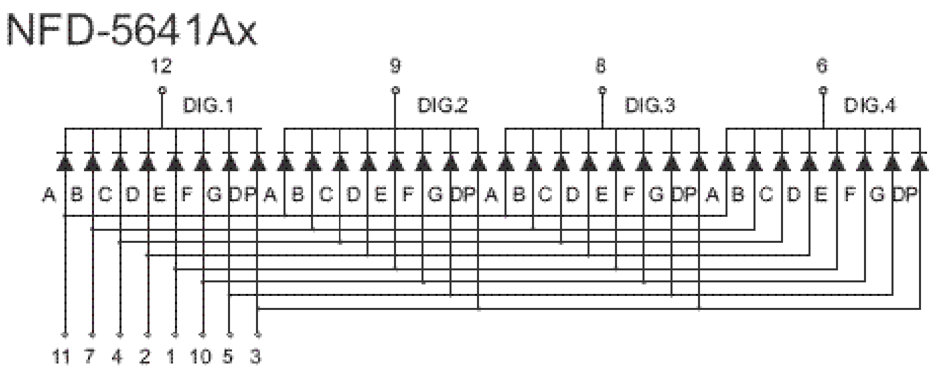


Figure 3.13: Circuit Diagram of 7FR5641AS

3.3.8 Design of Display

The displays setting is expected as shown in the Figure 3.14, the temperature of coolant and RPM of the cooling fan are displayed when the engine is turned on. In case there is a failure, warning icon will light on. Depending on the type of failure, three icons indicating the functionality of the fan, CTS and relay. The word “FAIL” icon represent the failure icon and use LED for indication. For normal operation, when the fan is triggered to turn on, the fan logo will turn on and display in blue colour and turned off when the cooling fan is turned off. Once the fan is turned on, the 4 digits 7 segments LED display will show the RPM of cooling fan and the 7 segments LED display will turn off when fan is turned off. The temperature of coolant is show once the engine is started and turned off when engine is turned off. The CTS logo beside the 4 digits 7 segments LED display for temperature will change its colour based on the temperature, it is expected to be three set of different colours, which are blue, yellow and red. Blue colour represent cold, yellow colour represent warm (normal operation) and red colour indicate hot. When there is a failure such as fan fail, CTS fail or relay fail, the word “FAIL” will be lighted up at the bottom together with the particular icon has fail to operate.

The circuit of the display box contain MAX7221 LED display driver, LM 7805 voltage regulator, two 4 digits 7 segments LEDs, buzzer, capacitors, resistors and LEDs. Figure 3.15 show the schematic diagram of the display box.

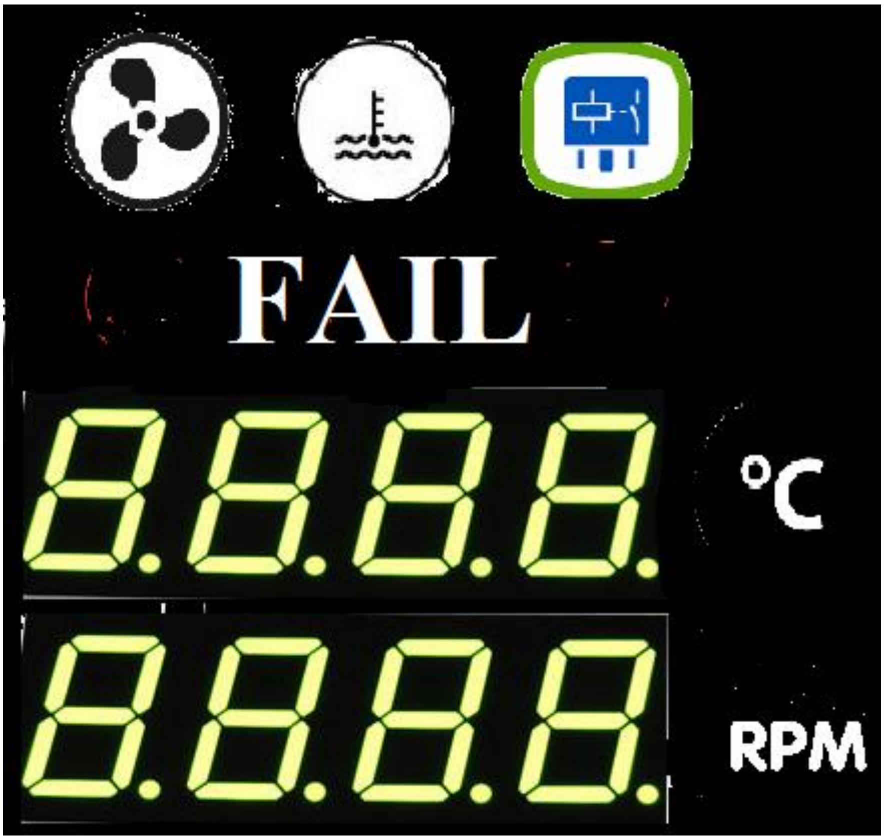


Figure 3.14: Display Design

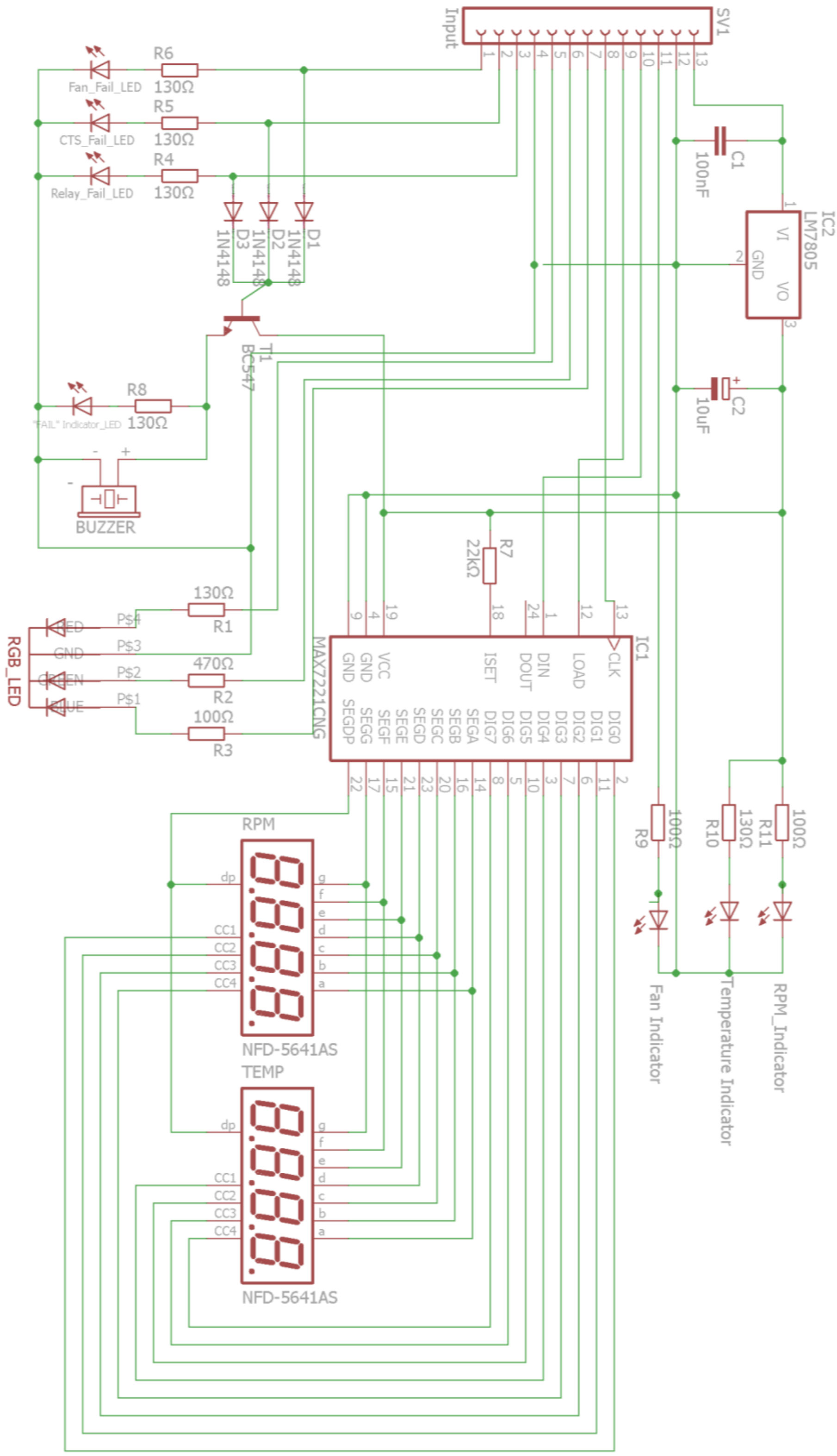


Figure 3.15: Display Circuit Diagram

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Prototype

A prototype of the cooling fan failure alert system is built as shown in Figure 4.1 and the functionality of the system is tested. The prototype consists of a fan, relay, PIC microcontroller, display box and a current sensor, voltage sensor, tachometer, coolant temperature sensor (CTS) and is powered up by a battery. A 3 layer plywood is used as base, 3 wood stick joined together to form a rectangular frame for the fan and is built on the plywood base.

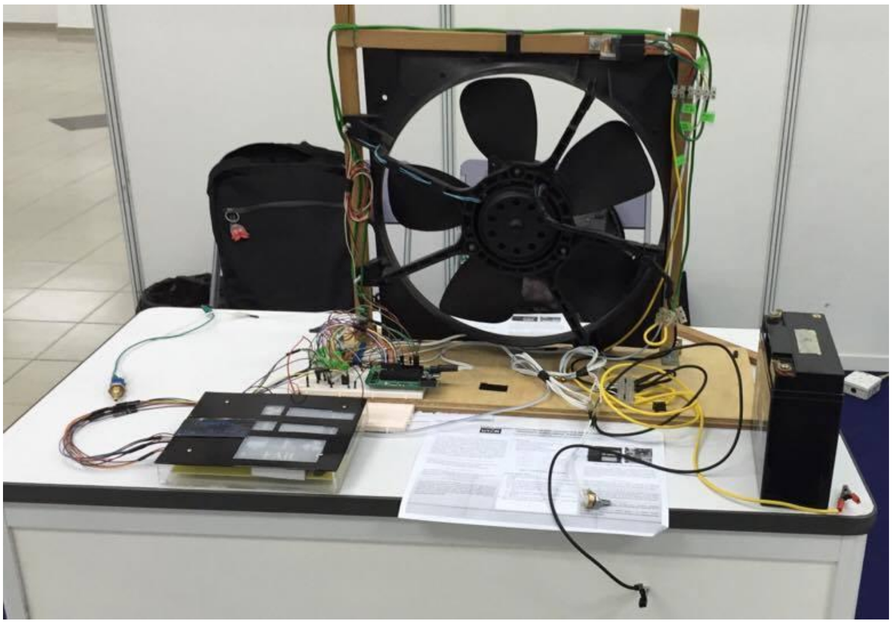


Figure 4.1: Cooling Fan Failure Alert System Prototype

The voltage sensor (voltage divider circuit), relay triggered circuit, Hall Effect current sensor and CTS circuit are combined and soldered in a solderable copper board as show in Figure 4.2. The PIC16F877A is used with SK40C (a 40pin PIC microcontroller starter kit designed). This board comes with basic element for user to begin project development as show in Figure 4.3. It offers plug and use features.

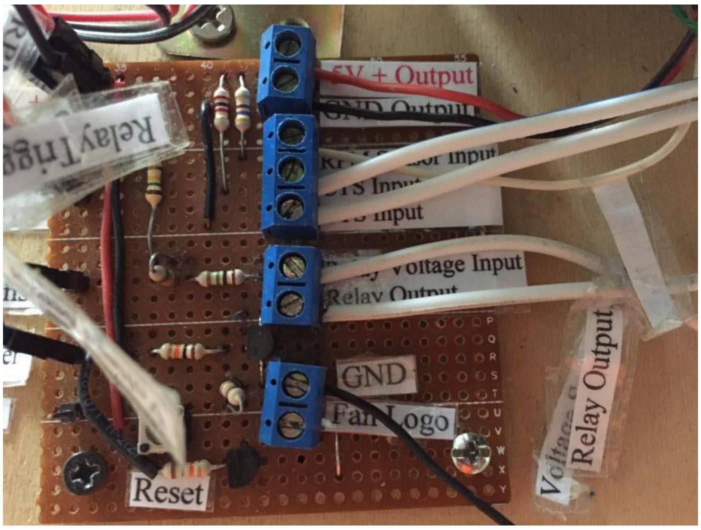


Figure 4.2: Circuit Board

Figure 4.3: PIC16F877A Circuit Board

4.2 Input Data from Sensor

The PIC microcontroller will extract data from four sensors that are the voltage sensor, current sensor, Hall Effect tachometer and CTS (coolant temperature sensor), and the PIC microcontroller will interpret the data obtained from the sensors and output to the display box and relay. The RPM of fan and temperature of engine will show in the display box.

4.2.1 Voltage Sensor

The voltage divider circuit is shown in Figure 3.3 that is used for voltage sensing. The circuit able to measure voltage ranging from 0 V to maximum of 5 V feeding to the PIC microcontroller. The accuracy of voltage sensor is based on the voltage supply to the Vcc pin of PIC microcontroller, it is better to supply the pin with exactly 5 V, but sadly the SK40C circuit board only manage to supply 4.71 V, so it does affect the accuracy for the reading, also there will be some voltage drop if the supply voltage to Vcc is used to supply voltage to the sensors. When calculating the voltage result, we should use 4.71 V as the voltage reference since the supply voltage to the Vcc pin is 4.71 V, so that the result obtained are closer to the actual result. Table 4.1 show the voltage obtained from the multimeter and PIC microcontroller respectively, we can see from the table, by comparing the average values of both measured and calculated, there are ±5 % in difference in voltage. Table 4.1 and Table 4.2 show the reading for measured and calculated current for battery at 11 V and 13 V respectively. Figure 4.4 show the measured voltage using multimeter.

Table 4.1: Voltage Reading for measured and calculated for battery at 11 V

Reading	Voltage Reading(V)	
	Calculated	**Measured**
1	11.49	10.74
2	11.49	10.69
3	10.94	10.73
4	10.83	10.75
5	11.16	10.73
Average	11.18	10.73

Table 4.2: Voltage Reading for measured and calculated for battery at 13 V

Reading	Voltage Reading(V)	
	Calculated	Measured
1	12.08	11.27
2	11.78	11.28
3	11.78	11.28
4	11.78	11.28
5	11.78	11.28
Average	11.84	11.28

Figure 4.4: Voltage Measured by Multimeter

4.2.2 Current Sensor

The starting current when the battery is at 11 V is ranging from 12 A to 13 A. When the battery is at fully charged until 13 V, the starting current draw by the fan motor is ranging from 9.54 A to 13.63 A. The motor drawing high current when starting only last for 1 second and later gradually drop to nominal operating current. This is due to the electric motor need high current to overcome torque. When the fan is stuck, the current draw by the fan motor is around 22.12 A to 22.72 A. Based on the result, we can see that large current is draw when electric fan motor is stuck to rotate, current is increasing as the torque required to turn the electric fan is increasing. The result obtained is intended to find out the nominal current for this electric fan operating current as well as the starting current. The calculated current and measured current there are less than ±2 % in difference. Table 4.3 and Table 4.4 show the measured and calculated current for battery at 11 V and 13 V respectively. Figure 4.5 show the measured current using multimeter.

Table 4.3: Current Reading for measured and calculated for battery at 11 V

Reading	Current Reading(A)	
	Calculated	**Measured**
1	5.75	5.6
2	5.61	5.6
3	5.61	5.6
4	5.61	5.6
5	5.61	5.6
Average	5.64	5.6

Table 4.4: Current Reading for measured and calculated for battery at 13 V

Reading	Current Reading(A)	
	Calculated	**Measured**
1	6.50	6.4
2	6.36	6.4
3	6.36	6.4
4	6.36	6.4
5	6.36	6.4
Average	6.39	6.4

Figure 4.5: Measured Current by Multimeter

4.2.3 Coolant Temperature Sensor

The coolant temperature sensor is the most accurate sensor among all sensors. The results measured by thermometer are very close to the calculated results by the PIC microcontroller. There is only less than ±1 % in difference for calculated and measured values. Figure 4.6 shows the value of temperature measured by thermometer. Table 4.5 shows the temperature reading by measured and calculated.

For every 10 °C, the CTS will have a different characteristic equation, hence, in order to obtain accurate result of CTS, for every 10 °C step will have a different formula to calculate the temperature. The resistance obtained from the CTS will be compared with each range of resistance in each 10 °C. The temperature will be calculated using the formula based on the range of resistance for each 10 °C it entered.

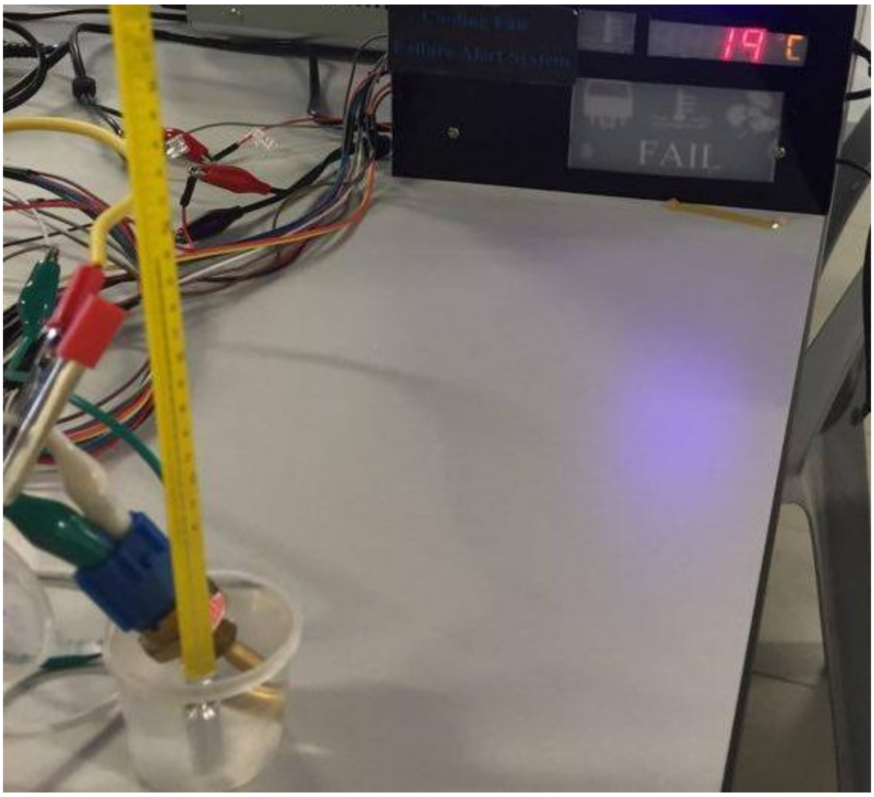


Figure 4.6: Measured Temperature and Display Temperature

Table 4.5: Temperature Reading for measured and calculated

Reading	Temperature Reading (°C)	
	Measured	Calculated
1	21.0	21
2	22.4	23
3	26.8	27
4	30.2	30
5	34.8	35
6	39.8	41
7	66.8	68
8	69.0	68

4.2.4 Tachometer

The implementation of the Hall Effect tachometer is simple, the Hall Effect sensor is attached on the motor and two magnets with different polarity are mounted on the fan blade, which consists of a south pole and a north pole as show in Figure 4.7.

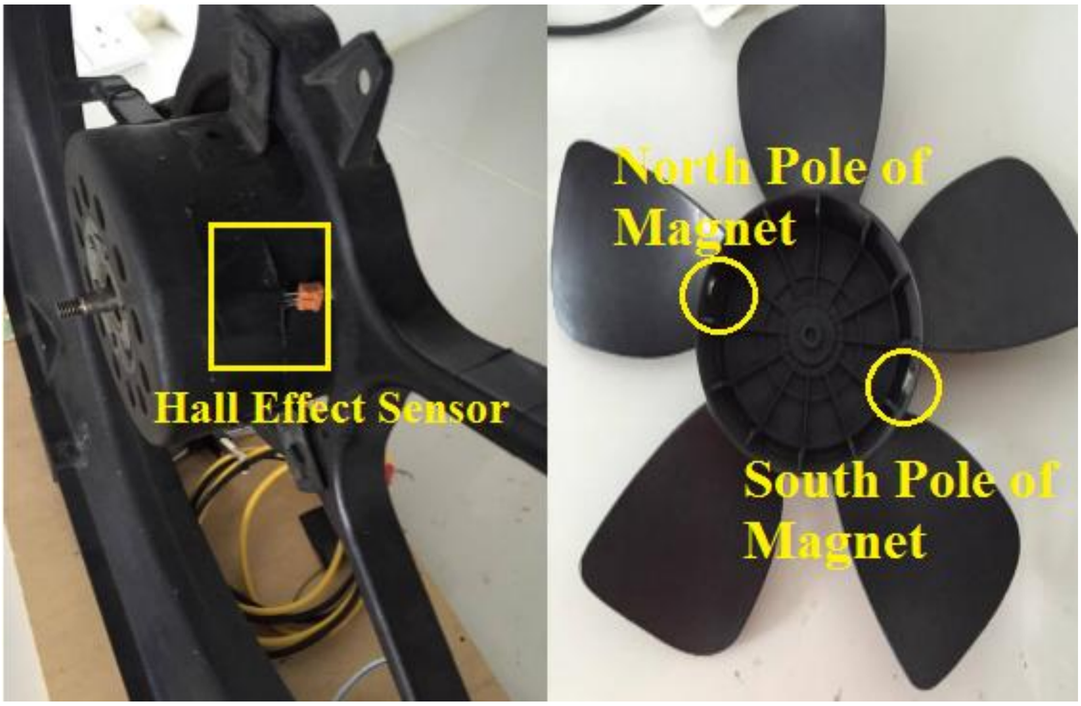


Figure 4.7: Implementation of Hall Effect Sensor

Figure 4.8 shows the measured RPM. The Hall Effect tachometer will generate a signals that serve like PWM signal to the microcontroller and then it is calculated as RPM. Hence, the input signals in a second determine the Hertz which then multiplied by 60 to get the RPM value. However, there will be ±5 % in difference for measured and calculated value, but this range of difference is acceptable. Table 4.6 shows the result for calculated and measured RPM reading.

Figure 4.8: Measured of RPM by Tachometer and Hall Effect Sensor

Table 4.6: Voltage Reading for measured and calculated for battery at 13 V

Reading	**Tachometer Reading(RPM)**	
	Measured	**Calculated**
1	1617	1680
2	1614	1680
3	1625	1680
4	1631	1680
5	1630	1680
6	1612	1680
7	1659	1680
8	1612	1680
9	1619	1680
10	1629	1680
Average	1624.8	1680

4.3 Output Data from PIC Microcontroller

The output data from PIC microcontroller will send signal to relay and the LED display driver in display box, the display box will then display the temperature in °C and also the RPM of the fan if the relay is triggered to turn on. The LEDs of the unit "RPM" and "°C" are always on when the display box is powered up (supply 12 V to the display box). The three indication icon relay, CTS and fan will lighted up together with the word "FAIL" when there is a failure.

4.3.1 Cooling Fan Relay Circuit

The Figure 4.10 show the cooling fan relay circuit, the fan is connected to normally open switch of the relay, when relay is triggered by the microcontroller, the fan will turn on. Figure 4.9 show the schematic connection of relay, pin 86 of the relay is supplied with 12 V. Pin 30 is the common supply of 12 V and will connected to cooling fan by pin 87 when the coil of relay is energized. Pin 85 is connected to the collector of the transistor, once the PIC microcontroller give signal to the base of the transistor, the transistor collector will connect to ground and trigger the relay to turn on the fan.

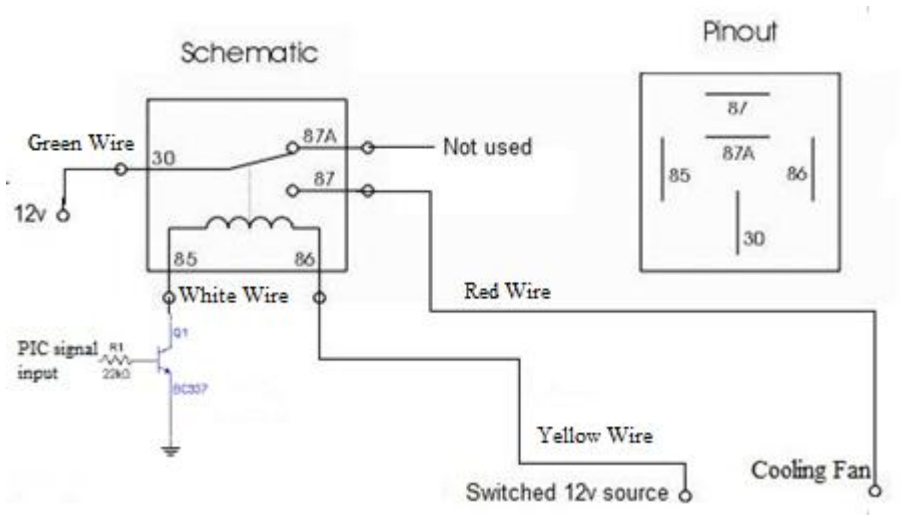


Figure 4.9: Automotive Relay Connection

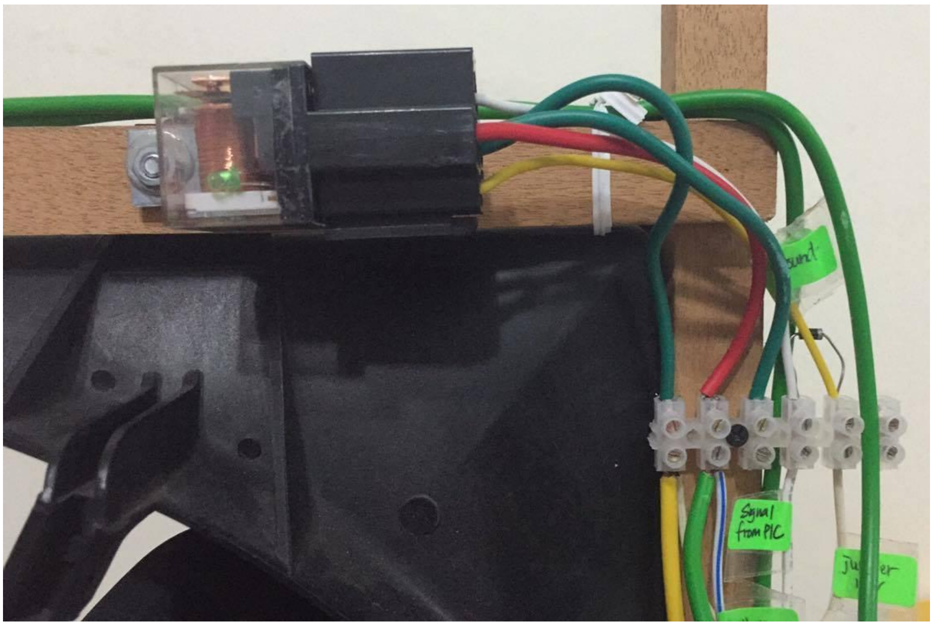


Figure 4.10: Cooling Fan Relay Circuit

4.3.2 Display

The temperature is set to three different colours indication for different range of temperature. For the temperature range of 91 °C to 120 °C, the CTS icon will show in red colour represent hot state, for temperature range of 31 °C to 90 °C, the CTS icon will show in yellow colour represent warm state, for temperature below 30 °C, the CTS icon will show in blue colour represent cold state. Figure 4.11 shows the three different stages in red, yellow and blue colour.

Figure 4.11: Red, Yellow and Blue Colour Indication for Different Temperature

Figure 4.12 shows the RPM of the fan when the fan is triggered to turn on and when fan is turned off, the fan logo also turned off. The fan logo is lighted up when the fan is turned on.

Figure 4.12: Cooling Fan On (right) and Cooling Fan Off (left)

In case of failure, the indicator icon such as fan, relay and CTS will light up together with the word “FAIL”. For instance, when the relay is unattached as show in

Figure 4.17 (assuming we are having a bad fan relay), the relay icon will light up together with the word “FAIL” show in Figure 4.13. Other than that, when the temperature is below 50 °C after two minutes the engine start up, the CTS icon will light up together with the word “FAIL” to indicate that the CTS is broken down show in Figure 4.14. Lastly, when the fan is stuck, the fan icon will also light up together with the word “FAIL” show in Figure 4.15. The system has been tested with all three situation stated above and the system is working well.

Figure 4.13: Relay Failure

Figure 4.14: CTS Failure

Figure 4.15: Fan Failure

4.4 Overall of System Working

4.4.1 PIC16F877A Microcontroller

The allocation of pin for PIC16F877A microcontroller is show in Table 4.7. Only 15 pins of the PIC16F877A are used, the rest of the pins are not used. Duo to the SK40C board cannot output exactly 5 V to supply power to power up the Hall Effect tachometer, current sensor and also CTS voltage divider circuit, also there is voltage drop if using supply from the board, so a voltage regulator LM7805 is used to step down the voltage from battery and supply 5 V to the sensors and CTS voltage divider circuit. All ground are connected to common ground for all the sensors, ground from supply battery and microcontroller.

The reason why the SK40C board is used because the crystal oscillator is very sensitive, if the PIC microcontroller is used in the breadboard, the crystal oscillator sometime is not working and the circuit is not stable, hence it affect the signal and clock pulse send to the MAX7221 IC LED display driver. The crystal oscillator is stable when using the SK40C board.

Table 4.7: Pin Allocation for PIC16F877A Microcontroller

Input/Output	Pin	Port	Description
Input	2	AN0	Voltage sensor
Input	3	AN1	Current sensor
Input	4	AN2	Temperature sensor
Input	6	T0CKI	Hall Effect tachometer
Output	10	RE2	LOAD(/CS)-pin for MAX7221
Output	18	SCK/SCL	Clock input for MAX7221
Output	22	RD3	Fan Logo
Output	23	RC4	Coolant Logo LED – Blue
Output	24	SDO	DATA IN-pin for MAX7221
Output	25	RC6	Coolant Logo LED – Green
Output	26	RC7	Coolant Logo LED – Red
Output	27	RD4	CTS failure icon
Output	28	RD5	Fan failure icon
Output	29	RD6	Relay failure icon
Output	30	RD7	Relay trigger circuit

4.4.2 System Flow

Figure 4.16 shows the program flow chart. The program start with initialize all parameters for all input and output ports and registers that stored values for ADC and counter for PIC microcontroller. And then the program will run through a loop in the main function with an interrupt function which interrupt every one second.

When interrupt happens, it will start to calculate the RPM and then display the calculated value of RPM in display box. If the RPM value is 0 (when the fan is off state), the display will display nothing on the 7 segments LED for RPM. After that, the program will resume back to where the program has stopped.

In the main function, the program start to read CTS voltage and based on the voltage to calculate the temperature. The temperature will be monitored, if engine temperature is still below 50°C after two minutes when engine start up, then the CTS icon will light up together will the word “FAIL” indicate CTS is broken down. In case the CTS is broken down, the temperature will still display in the 7 segments LED display. When the temperature has hit up to 100 °C or CTS is brown down, the fan will triggered to turn on. If the CTS has failed, the fan is triggered to turn on until a new CTS is replaced and the program is reset. There will be a counter to check if the fan is turned on more than 2 seconds, if less than 2 seconds, the program will go back to the beginning. If more than 2 second, the voltage and current are monitored. In case if voltage is less than 10 V after fan is triggered to turn on, the relay has failed and relay icon LED will turn on together with the word ”FAIL”. The program will go back to the beginning, current and RPM will not be checked due to fan is not running. If voltage is more than 10 V, it will proceed to check current and RPM. When current more than 21 A or RPM less than 1000, then fan failure icon will turn on, the program will go back to the beginning after that.

The fan will turn off when temperature dropped to 90 °C and turn on again when temperature reached 100 °C.

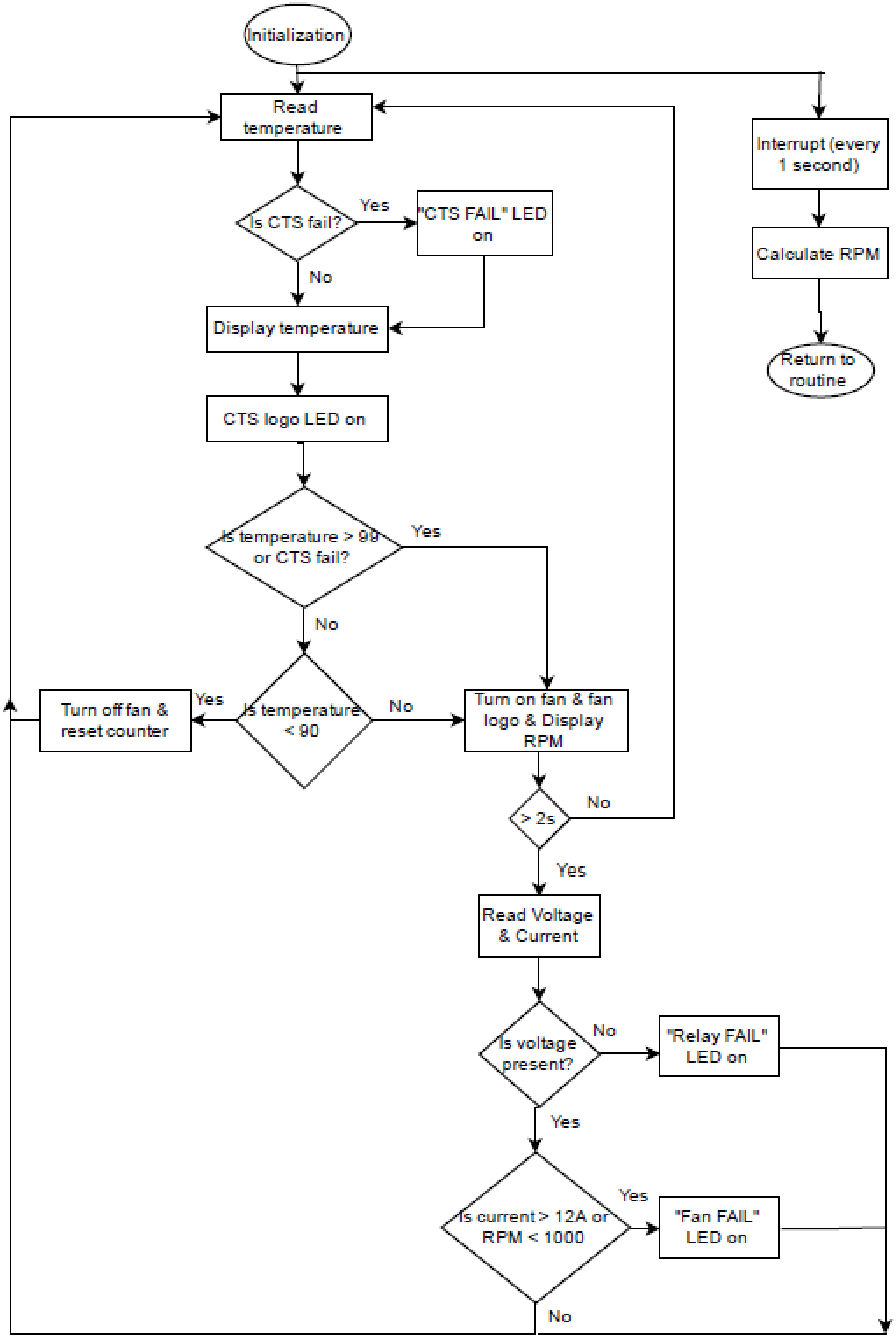


Figure 4.16: Flow Chart of the Program

4.4.3 Function Testing

The CTS is replaced with a potentiometer to control the resistance of the CTS, so that we can control the temperature to test the function of the program. The CTS is tested separately and the result is satisfying. The potentiometer is adjusted to three different range of temperature to test if the CTS logo will change its colour based on setting. The potentiometer is then adjusted to 100 °C to turn on the fan and then observe the RPM value, it is adjusted to 90 °C later to turn off the fan.

After that, the relay is detached (it act like having a bad relay) and temperature is adjusted to 100 °C to test if the relay fail icon will turn on. The detached relay is show in Figure 4.17.

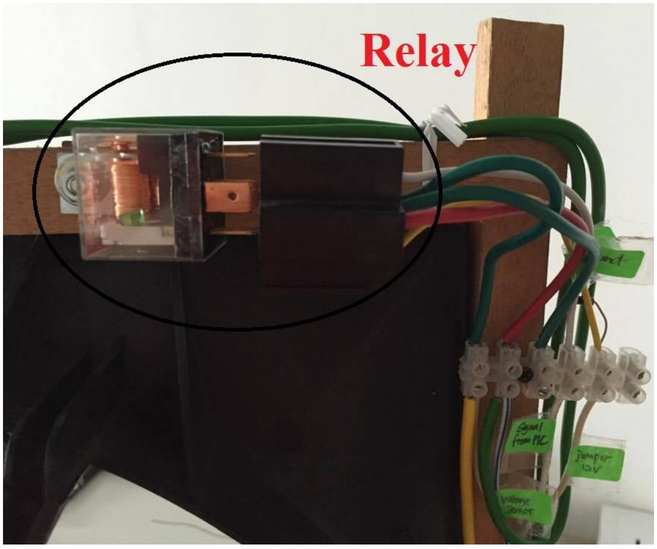


Figure 4.17: Detached Relay

Lastly, the relay is then attached and the temperature is adjusted to 100 °C to turn on the fan and the fan is stuck with a wood stick to test the fan fail as show in Figure 4.18, this will increase the current drawing by the fan motor at the same time. When the fan is stuck, the RPM is also lower than 1000 RPM, so the fan fail logo will turn on also.

Figure 4.18: Fan is stuck with wood stick

CHAPTER 5

ACHIEVEMENT

5.1 Competition Participation

The author took part in Novel Research and Innovation Competition (NRIC) 2016 held at Universiti Sains Malaysia (USM), Penang, Malaysia on 16^{th} August 2016 to 18^{th} August 2016. The objective of the competition is to give recognition to the innovation research projects done by the undergraduates, postgraduates and also the diploma, by the university. This project caught the public's attention and is strongly supported by Ministry of Higher Education Malaysia due to its huge impact on the country's development and affects all sectors positively in terms of research and innovation, regardless of public or private as well. There are about 97 groups of participants participated in the competition, some of the participants come from different country. Figure 5.1 show that the author has won a bronze medal from the competition.

Figure 5.1: Bronze Medal Certification

The competition comprises of six different categories, namely Life Sciences, Fundamental Sciences, Information Technology & Communication, Health & Medical Sciences, Engineering & Technology, and Social Transformation & Creative Arts. The author has participated in the Engineering & Technology category.

Besides that, the author also took part in FYP poster competition held at UTAR 5th floor KB Block Sg. Long on 25th August 2016. The objective of this competition is to provide an opportunity for FYP II students to demonstrate their independence, originality and ethics. The chosen track for the competition was track 4, research, design, development in science & engineering. There were 4 different tracks in the competition.

Lastly, the author also took part in IEM Intervarsity Electrical Engineering Final Year Project Poster Competition (EEFYP 2016) which held on 19th August 2016 in Electrical Engineering Technical Division, The Instituition of Engineers

Malaysia, Bangunan Ingeniur, Petaling Jaya. The author has won a consolation in the competition.

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

In conclusion, the vehicle overheat prevention system: cooling fan failure alert system is built to prevent vehicle from overheated in early stage before the cooling fan system damage the engine when the engine is overheated. Implementation of the system does not cost a lot of money and complicated system, hence the system has huge commercialized potential. Based on the result obtained in chapter 4, the system is expected working well, the components are carefully chosen to make sure it can operate normally in the hot environment in vehicle cooling system.

The objective of the title of the project is achieved and accomplished. The system is built and able to detect the possible failure involved in vehicle cooling fan system and hence alert the driver. The characteristic of electric fan is also determined from the result in chapter 4, a prototype is built and the alert system is implemented and tested, the system is working well.

6.2 Improvement and Recommendation

Several improvement are suggested in the project in order to enhance the entire system to more advance stage.

6.2.1 Built a 40 pins PIC Microcontroller starter kit

The SK40C starter kit board as mention in chapter 4 earlier, the voltage able supply to Vcc pin is only 4.71V. It will affect the accuracy of the ADC of microcontroller, so it is recommended to build a board of 40 pins PIC microcontroller starter kit that similar with SK40C board that comes with basic element and also offers plug and use features. Use a voltage regulator in the board to step down voltage from battery to exactly 5V and supply it to the Vcc pin of PIC microcontroller in order to get an accurate result.

6.2.2 Use LCD Screen to replace the Display Box

The display box for alert system is bulky, so it can reduced the size of it into a LCD screen as the SK40C board offer existing pad for 16 x 2 characters LCD display. When a LCD screen is used, it will also reduce the cost of building this project as the LED display driver MAX7221 IC is expensive. It also reduced the cost of buying components involved in the project building this display box.

6.2.3 Enhancing more Detection of Failure in Cooling System

In order to make the vehicle cooling failure alert system more reliable and complete, instead of focus on fan cooling system, it is suggested the cooling system should also focus on failure like thermostat, water pump, pressure cap and water level of cooling system.

REFERENCES

Stone, R. (1992) *Introduction to internal combustion engines*. 2nd edn. Basingstoke, England: Palgrave Macmillan. In-line Citation: (Stone, 1992) ISBN 0-333-55083-8

Nunney, M.J. (2006) *Light and heavy vehicle technology. 4th edn*. United Kingdom: Butterworth-Heinemann Ltd, Oxford. In-line Citation: (Nunney, 2006) ISBN 978-0-7506-8037-0

Ofria, C. (2016) *A Short Course on Cooling Systems*. Available at: http://www.carparts.com/classroom/coolingsystem.htm (Accessed: 30 March 2016).

How a Car Works (2016) *How an engine cooling system works*. Available at: http://www.howacarworks.com/basics/how-an-engine-cooling-system-works (Accessed: 30 March 2016).

AA1Car (2016) *Troubleshoot electric cooling fan*. Available at: http://www.aa1car.com/library/electric_cooling_fan.htm (Accessed: 31 March 2016).

Wright, M. (2016) *How does my cooling system work?* Available at: http://autorepair.about.com/od/glossary/ss/def_coolsystem.htm (Accessed: 31 March 2016). In-line Citation: (Wright, 2016).

Stern, D. (1994) *Radiator fans: Mechanical and electric cooling for cars*. Available at: http://www.allpar.com/fix/engines/fans.html (Accessed: 2 April 2016).

Newbold, D. and Bonnick, A. (2013) *Practical motor vehicle engineering*. United Kingdom: Taylor & Francis. In-line Citation: (Newbold and Bonnick, 2013).

Publishing, R.H. (2016) *Cooling system*. Available at: http://www.autoupkeep.com/LCChapter10.htm (Accessed: 5 April 2016).

Mellema, V. (2016) *How long does a cooling/radiator fan motor last?* Available at: https://www.yourmechanic.com/article/how-long-does-a-cooling-radiator-fan-motor-last (Accessed: 5 April 2016).

Inc, B.H. (2010) *Common causes for electric motor failure - motor failure analysis.* Available at: http://www.brighthubengineering.com/commercial-electrical-applications/78579-determining-causes-for-electric-motor-failure/ (Accessed: 5 April 2016).

MicroSystems, A. (2016) *Allegro MicroSystems - ACS712: Fully integrated, hall-effect-based linear current sensor IC with 2.1 kVRMS voltage isolation and a low-resistance current conductor*. Available at: http://www.allegromicro.com/en/Products/Current-Sensor-ICs/Zero-To-Fifty-Amp-Integrated-Conductor-Sensor-ICs/ACS712.aspx (Accessed: 16 April 2016).

MicroSystems, A. (2016) *Allegro MicroSystems - A1220, A1221, A1222, and 1223: Chopper-stabilized precision hall-effect latches*. Available at: http://www.allegromicro.com/en/Products/Magnetic-Digital-Position-Sensor-ICs/Hall-Effect-Latches-Bipolar-Switches/A1220-1-2-3.aspx (Accessed: 16 April 2016).

NASA (2015) *Ideal Otto Cycle*. Available at: https://www.grc.nasa.gov/www/k-12/airplane/otto.html (Accessed: 16 April 2016).

Enginebasics.com (2016) Understanding automotive Relays. Available at: http://www.enginebasics.com/Engine%20Basics%20Root%20Folder/Automotive%20Relays.html (Accessed: 27 August 2016).

APPENDICES

APPENDIX A: Full PIC16F877A source code

```
/*
 * File:   ADC.c
 * Author: Liang
 *
 * Created on May 19, 2016, 10:55 AM
 */

#include <stdio.h>
#include <stdlib.h>
#include<pic16f877a.h>
#include <xc.h>
#define _XTAL_FREQ 20000000


// Temperature
short CTSResistance;
int TEMPResult=0;
int t1,t10,t100,t1000;
int temp_one;
int temp_ten;
int temp_hundred;
int temp_thousand;

//AN0=voltage,AN1=Current,AN2=CTS
int value1,AN0,AN1,AN2;

//volatge
int voltage_one=0,voltage_ten=0,voltage_hundred=0,voltage_thousand=0,v1=0;

// RPM
int rpm_one,freq=0,rpm,r1;
int rpm_ten;
int rpm_hundred;
int rpm_thousand;

float ADCResult;
int fanOn=0;
int count = 0,count1 = 0, count2 = 0, count3 = 0;
```

```
// CONFIG
#pragma config FOSC = HS        // Oscillator Selection bits (HS oscillator)
#pragma config WDTE = OFF       // Watchdog Timer Enable bit (WDT disabled)
#pragma config PWRTE = OFF      // Power-up Timer Enable bit (PWRT disabled)
#pragma config BOREN = OFF      // Brown-out Reset Enable bit (BOR disabled)

// Low-Voltage (Single-Supply) In-Circuit Serial Programming Enable bit
//(RB3 is digital I/O, HV on MCLR must be used for programming)
#pragma config LVP = OFF

// Data EEPROM Memory Code Protection bit (Data EEPROM code protection off)
#pragma config CPD = OFF

// Flash Program Memory Write Enable bits (Write protection off;
//all program memory may be written to by EECON control)
#pragma config WRT = OFF

// Flash Program Memory Code Protection bit (Code protection off)
#pragma config CP = OFF
// Flash Program Memory Code Protection bit (Code protection off)
#pragma config CP = OFF

#define cs PORTEbits.RE2
#define Fan PORTDbits.RD7
#define RelayF PORTDbits.RD6
#define FanF PORTDbits.RD5
#define CTSF PORTDbits.RD4
#define FanLogo PORTDbits.RD3
#define Red PORTCbits.RC7
#define Green PORTCbits.RC6
#define Blue PORTCbits.RC4

void RPM_display(int RPMvalue);
void temperature(float resultADC);
void ValueConv(int conv);
void display(int addr,int value);
int ReadADC(unsigned char channel);
void FailureCheck ();

// interrupt service routine
void interrupt ISR() {
  // has the interrupt been fired by timer 1?
  if(PIR1bits.TMR1IF == 1) {
    count++;        // increase the number of times interrupt has been fired

    // count till 19 interrupts have been fired (about 1 second has elapsed)
    if(count == 19 ) {
        freq = TMR0;
        rpm = freq*60*4;        //convert frequency to rpm
        TMR0=0;                 //reset register for counter
         count = 0;             //reset counter
          }
    // reset the overflow interrupt flag
    PIR1bits.TMR1IF = 0;

    }
}
```

```
void main() {

    TRISD = 0b00000000;     //Setting PORT D as output port
    PORTD = 0b00000000;     //Initializing PORTD
    TRISB = 0b00000000;    //Setting PORT B as output port,pin 34
    PORTB = 0b11000000;     //Initializing PORT B
    TRISC = 0b00000000;    //Setting PORT C as output port
    PORTCbits.RC4 = 0;     //Initializing PORT C ,Blue
    PORTCbits.RC6 = 0;     //Initializing PORT C ,Green
    PORTCbits.RC7 = 0;     //Initializing PORT C ,Red
    TRISAbits.TRISA0 = 1;   //MAKE PORT A AS INTPUT, FOR AN0 ONLY
    PORTAbits.RA0 = 0;     //Initializing PORT A
    TRISAbits.TRISA1 = 1;   //MAKE PORT A AS INTPUT, FOR AN1 ONLY
    PORTAbits.RA1 = 0;     //Initializing PORT A
    TRISAbits.TRISA2 = 1;   //MAKE PORT A AS INTPUT, FOR AN2 ONLY
    PORTAbits.RA2 = 0;     //Initializing PORT A
    TRISAbits.TRISA4 = 1;   //Setting PORT A as input, pin 6, RPM
    PORTAbits.RA4 = 0;     //Initializing PORT A


    // Configuration for SPI
    TRISEbits.TRISE2 = 0;          //used for Chip Select function;
    TRISCbits.TRISC3 = 0;          //used for SCK; configured as output,pin18
    TRISCbits.TRISC5 = 0;          //used for transferring serial data;
    SSPSTAT = 0xC0;                //SSPSTAT=11000000, configure SPI settings

    //SSPCON1=00100000 Enables serial port and configures
    //SCK, SDO and SDI as serial port pins & clock=FOSC/4
    SSPCON = 0b00100000;
    display(0x0B,0X07);            //scanlimit(digits 0-7)
    display(0x09,0X00);            //decode mode off
    display(0x0A,0X0F);            //intensity highest
    display(0x0C,0X01);            //shutdown mode off
    display(0x0F,0X00);            //test mode off

    //Configure for ADC
    ADCON0bits.ADCS0 = 0;   //Selecting the clk division factor = FOSC/64
    ADCON0bits.ADCS1 = 1;   //Selecting the clk division factor = FOSC/64
    ADCON1bits.ADCS2 = 1;   //Selecting the clk division factor = FOSC/64
    ADCON1bits.ADFM = 1;    //Result right justified
    ADCON1bits.PCFG = 0;    //Setting all the analog ports as Analog inputs
    ADCON0bits.ADON = 1;    //Turns on ADC module

    //Configuration for Timer0--RPM(counter-TMR0 & timer-TMR1)
    CCP1CON = 0x00;             // Disable Comparators
    CCP2CON = 0x00;             // Disable Comparators
    OPTION_REG = 0b00101000;    // Prescaler (1:1), TOCS =1,COUNTER,PIN 6
    TMR0=0;                     // Initialize timer 0

    //Configuration for Timer 1
    TMR1H = TMR1L = 0;
    T1CON = 0x00100001;         //PRESCALAR 1:4,INTERNAL CLOCK
    PIE1bits.TMR1IE = 1;        // enable interrupt flag for timer 1
    INTCONbits.PEIE = 1;        // enable peripheral interrupts
    INTCONbits.GIE = 1;         // enable global interrupts
```

```
while(1){

AN2 = ReadADC('2');              //read temperature from CTS
temperature(AN2);
if (count1<960){                  //Waits for 2 min for engine warm up
    count1++;
}
else if(TEMPResult<50&&count1>958){
    CTSF=1;     //CTS fail logo turn on
    Red=0;  Green=0;    Blue=0;     //red colour
    Fan=1;                          // turn on fan
        }

//if input voltage is 5V or CTS resistance more than 45000, CTS has failed
if (AN2==1023 || CTSResistance>=45000){
    CTSF=1;
    Fan =1;
    Red=0;
    Green=0;
    Blue=0;     //red colour
}
// if temperature = 100 C, trigger relay turn on fan
else if(TEMPResult>99 && TEMPResult<=120){
    //output signal to Relay turn on fan
    Fan =1;
    fanOn=1;
    Red=1;
    Green=0;
    Blue=0;     //red colour
}
//turn of fan when temperature less than (90 C)
else if (TEMPResult<90 && TEMPResult>50){
    //stop signal to relay to turn off fan
    Fan =0;
    fanOn=0;
    Red=1;
    Green=1;
    Blue=0;     //yellow colour
}
else if (TEMPResult<=30){
    Red=0;
    Green=0;
    Blue=1;     //blue colour

}
// if fan turn on
if(fanOn==1){
    count2++;
    if (count2<3){
        __delay_ms(1000);          //delay 1s
    }

     if (RelayF==0){
        FanLogo=1;
        RPM_display(rpm);
     }
     else if (RelayF==1 || FanF==1){
```

```
                FanLogo=0;
                display(1,'-');
                display(2,'-');
                display(3,'-');
                display(4,'-');
             }
            FailureCheck();
        }

        else if (fanOn==0){
            FanLogo=0;
            display(1,'-');
            display(2,'-');
            display(3,'-');
            display(4,'-');
        }

        }
}
void RPM display(int RPMvalue){
    r1=RPMvalue;
    rpm_one=r1%10;
    r1=r1/10;
    rpm_ten=r1%10;
    r1=r1/10;
    rpm_hundred=r1%10;
    r1=r1/10;
    rpm_thousand=r1;
    ValueConv(rpm_thousand);
    display(1,value1);
    ValueConv(rpm_hundred);
    display(2,value1);
    ValueConv(rpm_ten);
    display(3,value1);
    ValueConv(rpm_one);
    display(4,value1);
}
// checking for voltage and current

void FailureCheck ()
{

    AN0=ReadADC('0');                         //get voltage from sensor
    //voltage = (AN0*(25/1023))/0.2, calculate voltage value

    AN1=ReadADC('1');                          //get current from sensor
    //current=((AN1*(5/1024)-2.5)/0.066),
    //convert voltage to current,offset is 2.5V during 0A

    // allow voltage and current stabilized
     if (count3<150){
        count3++;
        }

    //voltage<2.2v in PIC,less than 10V, ratio is 1:0.2, relay fail on
    if (AN0<82){
        RelayF = 1;
    }
    else if (AN0>=82)
    {
        RelayF=0;
    }
```

```
    // output of current sensor more than 3.9V when stalk,
    //AN0=798, then display FanF=1, if current is high or fan rpm low, fan fail
    //if >21A OR RPM Below 1000 RPM with the condition voltage is present
    else if (((AN1>=798||rpm<=1000)&&count3>145)&&AN0>81){
        FanF=1;                               //fan icon on
    }
    else if (AN1<720&&AN0>=82){              // less than 15A, turn off fan icon
        FanF=0;
    }
}
int ReadADC(unsigned char channel)
 {
        // Select desired channel for ADC operation
         switch (channel)
            {
            case '0':                // AN0
                ADCON0 = 0b10000001;  //Selects channel 0 ( AN0 )
                break;
            case '1':                // AN1
                ADCON0 = 0b10001001;  //Selects channel 0 ( AN1 )
                break;
            case '2':                //AN2
                ADCON0 = 0b10010001;  //Selects channel 0 ( AN2 )
                break;
            default:                   // Any other value will default to AN0
//1000 0001-AN1 set up ADC ADCON0 Register it sets the frequency Fosc/64
                ADCON0 = 0b10010001;
            }    // switch (channel)
    // Channel selected proceed with ADC convertion
    ADCON0bits.GO = 1;        //Starts ADC conversion
    while (ADCON0bits.nDONE == 1);          //wait till ADC conversion is over
    ADCResult = ((ADRESH<<8) + ADRESL) ;      //Merging the MSB and LSB
    return ADCResult;
 }

//Calculate temperature in degree celcius
void temperature(float resultADC){

    resultADC = (resultADC*5)/1023;
    CTSResistance = (2000*resultADC)/(5-resultADC);

    if (CTSResistance>=26114)                // temp is  -40 to -30
    {  TEMPResult=((CTSResistance-31483)/-1920);}
    else if (CTSResistance>=15462)              // temp is  -30 to -20
    {  TEMPResult=((CTSResistance-5842)/-1065);}
    else if (CTSResistance>=9397)              // temp is  -20 to -10
    {  TEMPResult=((CTSResistance-3332)/-606.5);}
    else if (CTSResistance>=5896)              // temp is  -10 to 0
    {  TEMPResult=((CTSResistance-5896)/-305.1);}
    else if (CTSResistance>=3792)              // temp is  0 to 10
    {  TEMPResult=((CTSResistance-5896)/-210.4);}
    else if (CTSResistance>=2550)              // temp is  10 to 20
    {  TEMPResult= ((CTSResistance-5084)/-129.2);}
    else if (CTSResistance>=1707)              // temp is  20 to 30
    {  TEMPResult=((CTSResistance-4086)/-79.3);}
     else if (CTSResistance>=1175)              // temp is  30 to 40
    {  TEMPResult=((CTSResistance-3303)/-53.2);}
    else if (CTSResistance>=834)              // temp is  40 to 50
    {  TEMPResult=((CTSResistance-2539)/-34.1);}
    else if (CTSResistance>=596)              // temp is  50 to 60
    {  TEMPResult=((CTSResistance-2024)/-23.8);}
```

```
    {  TEMPResult=((CTSResistance-2539)/-34.1);}
    else if (CTSResistance>=596)                // temp is  50 to 60
    {  TEMPResult=((CTSResistance-2024)/-23.8);}
    else if (CTSResistance>=436)                // temp is  60 to 70
    {  TEMPResult=((CTSResistance-1556)/-16);}
    else if (CTSResistance>=323)                // temp is  70 to 80
    {  TEMPResult=((CTSResistance-1227)/(-11.3));}
    else if (CTSResistance>=243)                // temp is  80 to 90
    {  TEMPResult=((CTSResistance-963)/-8);}
    else if (CTSResistance>=187)                // temp is  90 to 100
    {  TEMPResult=((CTSResistance-747)/-5.6);}
    else if (CTSResistance>=144)                // temp is  100 to 110
    {  TEMPResult=((CTSResistance-617)/-4.3);}
    else if (CTSResistance>=113)                // temp is  110 to 120
    {  TEMPResult=((CTSResistance-485)/-3.1);}
    else if (CTSResistance>=113)                // temp more than 120
    {  TEMPResult=((CTSResistance-401)/-2.4);}

    t1= TEMPResult;
    temp_one=t1%10;
    t1=t1/10;
    temp_ten=t1%10;
    t1=t1/10;
    temp_hundred=t1%10;
    t1=t1/10;
    temp_thousand=t1;

    // turn off the display of 1st digit for temp
    display(5,'-');
    // turn on the display of 2nd digit of temp if temp >= 100
    if (temp_hundred==1)
    {ValueConv(temp_hundred);
        display(6,value1);
    }
    else
    {display(6,'-');}
    ValueConv(temp_ten);
    display(7,value1);
    ValueConv(temp_one);
    display(8,value1);

}
void display(int addr,int value)
{
  cs = 0;                           //enable MAX7221 to receive new value
  SSPBUF = addr;                    //sending the address of digit serially
  while(!SSPSTATbits.BF);           //wait until the address is sent
  switch (value)
  {  case '0':
    value = 126;
        break;
case '1':
     value = 48;
        break;
case '2':
    value = 109;
        break;
case '3':
    value= 121;
        break;
case '4':
    value = 51;
        break;
```

```
case '5':
    value = 91;
        break;
case '6':
    value = 95;
        break;
case '7':
    value = 112;
        break;
case '8':
    value = 127;
        break;
case '9':
    value = 115;
        break;
case '-':
    value = 0;break;}

  SSPBUF = value;               //sending the value which is to be displayed
  while(!SSPSTATbits.BF);       //wait until the value is sent
  cs = 1;                       //disable MAX7221 to receive any new value
}
void ValueConv(int conv)
{
    if (conv == 0)
         {value1 = '0';}
    else if (conv == 1)
         {value1 = '1';}
    else if (conv == 2)
         {value1 = '2';}
    else if (conv == 3)
         {value1 = '3';}
    else if (conv == 4)
         {value1 = '4';}
    else if (conv == 5)
         {value1 = '5';}
    else if (conv == 6)
         {value1 = '6';}
    else if (conv == 7)
         {value1 = '7';}
    else if (conv == 8)
         {value1 = '8';}
    else if (conv == 9)
         {value1 = '9';}
 }
```

APPENDIX B: MAX7221 IC Data Sheet

maxim integrated™

MAX7219/MAX7221

Serially Interfaced, 8-Digit LED Display Drivers

General Description

The MAX7219/MAX7221 are compact, serial input/output common-cathode display drivers that interface microprocessors (µPs) to 7-segment numeric LED displays of up to 8 digits, bar-graph displays, or 64 individual LEDs. Included on-chip are a BCD code-B decoder, multiplex scan circuitry, segment and digit drivers, and an 8x8 static RAM that stores each digit. Only one external resistor is required to set the segment current for all LEDs. The MAX7221 is compatible with SPI™, QSPI™, and MICROWIRE™, and has slew-rate-limited segment drivers to reduce EMI.

A convenient 4-wire serial interface connects to all common µPs. Individual digits may be addressed and updated without rewriting the entire display. The MAX7219/MAX7221 also allow the user to select code-B decoding or no-decode for each digit.

The devices include a 150µA low-power shutdown mode, analog and digital brightness control, a scan-limit register that allows the user to display from 1 to 8 digits, and a test mode that forces all LEDs on.

For applications requiring 3V operation or segment blinking, refer to the MAX6951 data sheet.

Applications

Bar-Graph Displays | Panel Meters
Industrial Controllers | LED Matrix Displays

Features

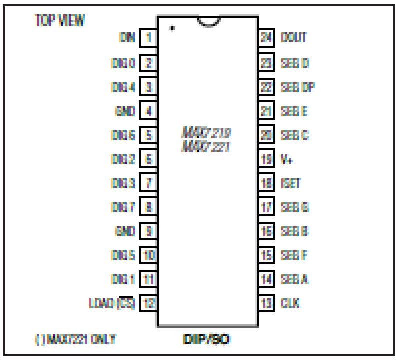
- 10MHz Serial Interface
- Individual LED Segment Control
- Decode/No-Decode Digit Selection
- 150µA Low-Power Shutdown (Data Retained)
- Digital and Analog Brightness Control
- Display Blanked on Power-Up
- Drive Common-Cathode LED Display
- Slew-Rate Limited Segment Drivers for Lower EMI (MAX7221)
- SPI, QSPI, MICROWIRE Serial Interface (MAX7221)
- 24-Pin DIP and SO Packages

Ordering Information

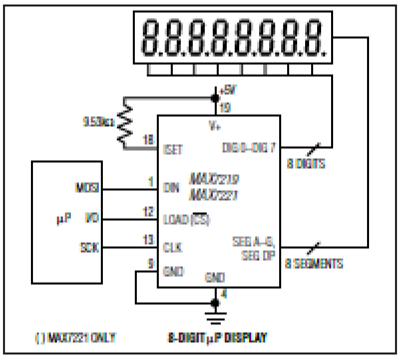
PART	TEMP RANGE	PIN-PACKAGE
MAX7219CNG	0°C to +70°C	24 Narrow Plastic DIP
MAX7219CWG	0°C to +70°C	24 Wide SO
MAX7219C/D	0°C to +70°C	Dice*
MAX7219ENG	-40°C to +85°C	24 Narrow Plastic DIP
MAX7219EWG	-40°C to +85°C	24 Wide SO
MAX7219ERG	-40°C to +85°C	24 Narrow CERDIP

Ordering Information continued at end of data sheet.
**Dice are specified at TA = +25°C.*

Pin Configuration



Typical Application Circuit



SPI and QSPI are trademarks of Motorola Inc. MICROWIRE is a trademark of National Semiconductor Corp.

For pricing, delivery, and ordering information, please contact Maxim Direct at 1-888-629-4642, or visit Maxim's website at www.maximintegrated.com.

19-4452; Rev 4; 7/03

APPENDIX C: SK40C Data Sheet

WHAT EXACTLY INSIDE SK40C

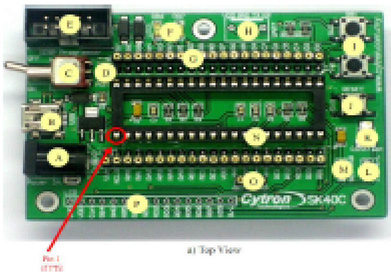
BOARD OVERVIEW

SK40C, another enhanced version after SK40B which can support for 40 pins PIC such as PIC16F and PIC18F family. This board comes with basic element that mostly use by hobbyist and experts. This kit is design to offer:

- Industrial grade PCB
- Every board is being fully tested before shipped to customer
- Compact, powerful, flexible and robust start-up platform
- Suitable for hobbyists and experts
- Save development and soldering time
- No extra components required for the PIC to function
- All 33 I/O pins are nicely labeled to avoid miss-connection by users
- Connector for UIC00A (low cost USB ICSP PIC Programmer) - simple and fast method to load program
- Fully compatible with SK40B
- No more frustrated work plugging PIC out and back for re-programming
- Perfectly fit for 40 pins 16F and PIC18F PIC
- With UIC00A, program can be loaded in less than 5 seconds
- More convenient to use and it is smaller than SK40B.
- Maximum current is 0.5A.
- Dimension: 85mm x 55mm

SK40C come with additional features:

- 2 x Programmable switch
- 2 x LED indicator
- Turn pin for crystal. User may use others crystal provided.
- LCD display (optional)
- UART communication
- USB on board.
- And all the necessities to eliminate users difficulty in using PIC.



a) Top View

LABEL	FUNCTION	LABEL	FUNCTION
A	DC Power Adapter Socket	I	Programmable Push Button
B	USB Connector	J	Reset Button
C	Toggle Switch for Power Supply	K	LCD Contrast
D	Power Indicator LED	L	JP8 for LCD Backlight
E	Connector for UIC00B Programmer	M	JP9 for USB
F	LED Indicator	N	40 pin IC Socket for PIC MCU
G	Header Pin and Turn Pin	O	Turn Pin for Crystal
H	UART Connector	P	LCD Display

A – DC power adapter socket for user to plug in DC adapter. The input voltage should be range 7 – 15V.

B – USB connector for communication between devices and a host controller. This function only valid for certain model of PIC microcontroller. Please refer to SK40C User Manual. The power LED will light ON when USB cable is connected.

C – Toggle switch to ON/OFF the power supply from DC adapter.

D – Power indicator LED for on board. It will light ON

APPENDIX D: ACS712ELCTR-30A-T datasheet

Allegro MicroSystems, LLC

ACS712

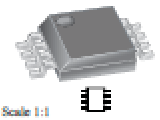
Fully Integrated, Hall Effect-Based Linear Current Sensor IC with 2.1 kVRMS Isolation and a Low-Resistance Current Conductor

Features and Benefits

- Low-noise analog signal path
- Device bandwidth is set via the new FILTER pin
- 5 μs output rise time in response to step input current
- 80 kHz bandwidth
- Total output error 1.5% at $T_A = 25°C$
- Small footprint, low-profile SOIC8 package
- 1.2 mΩ internal conductor resistance
- 2.1 kVRMS minimum isolation voltage from pins 1-4 to pins 5-8
- 5.0 V, single supply operation
- 66 to 185 mV/A output sensitivity
- Output voltage proportional to AC or DC currents
- Factory-trimmed for accuracy
- Extremely stable output offset voltage
- Nearly zero magnetic hysteresis
- Ratiometric output from supply voltage

TÜV America
Certificate Number:
U8V 06 05 54214 010

Package: 8 Lead SOIC (suffix LC)



Approximate Scale 1:1

Description

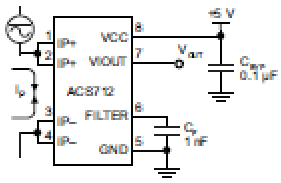
The Allegro™ ACS712 provides economical and precise solutions for AC or DC current sensing in industrial, commercial, and communications systems. The device package allows for easy implementation by the customer. Typical applications include motor control, load detection and management, switch-mode power supplies, and overcurrent fault protection. The device is not intended for automotive applications.

The device consists of a precise, low-offset, linear Hall circuit with a copper conduction path located near the surface of the die. Applied current flowing through this copper conduction path generates a magnetic field which the Hall IC converts into a proportional voltage. Device accuracy is optimized through the close proximity of the magnetic signal to the Hall transducer. A precise, proportional voltage is provided by the low-offset, chopper-stabilized BiCMOS Hall IC, which is programmed for accuracy after packaging.

The output of the device has a positive slope ($>V_{IOUT(Q)}$) when an increasing current flows through the primary copper conduction path (from pins 1 and 2, to pins 3 and 4), which is the path used for current sampling. The internal resistance of this conductive path is 1.2 mΩ typical, providing low power loss. The thickness of the copper conductor allows survival of

Continued on the next page…

Typical Application



Application 1. The ACS712 outputs an analog signal, V_{OUT}, that varies linearly with the uni- or bi-directional AC or DC primary sampled current, I_P, within the range specified. C_F is recommended for noise management, with values that depend on the application.

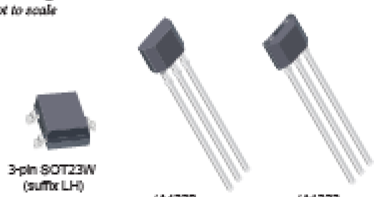
APPENDIX E: A1220 datasheet

A1220, A1221, A1222, and A1223

Chopper-Stabilized Precision Hall-Effect Latches

Features and Benefits
- AEC-Q100 automotive qualified
- Symmetrical latch switchpoints
- Resistant to physical stress
- Superior temperature stability
- Output short-circuit protection
- Operation from unregulated supply down to 3 V
- Reverse-battery protection
- Solid-state reliability
- Small package sizes

Packages:
Not to scale



3-pin SOT23W (suffix LH)

(A1220 and 1221)

(A1222 and A1223)

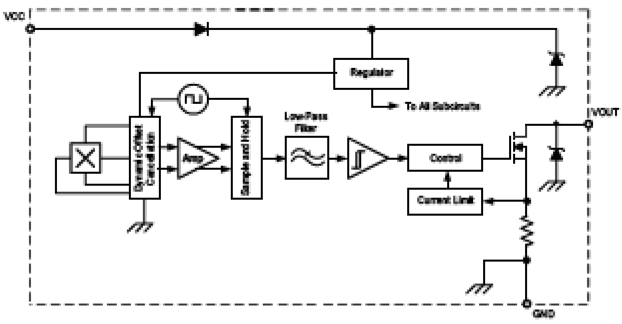
3-pin SIP (suffix UA)

Description
The A1220, A1221, A1222, and A1223 Hall-effect sensor ICs are extremely temperature-stable and stress-resistant devices especially suited for operation over extended temperature ranges to 150°C. Superior high-temperature performance is made possible through dynamic offset cancellation, which reduces the residual offset voltage normally caused by device overmolding, temperature dependencies, and thermal stress. Each device includes on a single silicon chip a voltage regulator, Hall-voltage generator, small-signal amplifier, chopper stabilization, Schmitt trigger, and a short-circuit protected open-drain output to sink up to 25 mA. A south pole of sufficient strength turns the output on. A north pole of sufficient strength is necessary to turn the output off.

An onboard regulator permits operation with supply voltages of 3 to 24 V. The advantage of operating down to 3 V is that the device can be used in 3 V applications or with additional external resistance in series with the supply pin for greater protection against high voltage transient events.

Two package styles provide magnetically optimized packages for most applications. Package type LH is a modified 3-pin SOT23W surface-mount package, while UA is a three-pin ultra-mini SIP for through-hole mounting. Both packages are lead (Pb) free, with 100% matte-tin-plated leadframes.

Functional Block Diagram



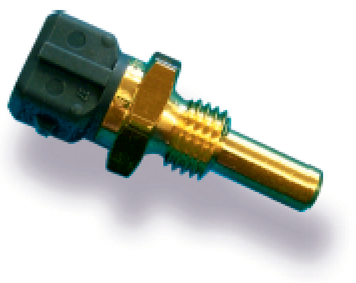
APPENDIX F: CTS Data Sheet

Bosch Motorsport | Temperature Sensor NTC M12

Temperature Sensor NTC M12

www.bosch-motorsport.com

BOSCH
Invented for life



- Measurement range: -40 to 130°C
- Robust design

This sensor is designed to measure fluid temperatures e.g. oil, water or fuel. This signal may be used as a control value for engine control units or as a measurement value which is logged in a data acquisition system.
The NTC sensing element has a negative temperature coefficient. This means, that with increasing temperature the conductivity rises. The sensing element of the temperature sensor is made of semiconducting heavy metal oxide and oxidized mixed crystals, which are equipped with a protective housing.
The main benefit of the sensor is the combination of a high quality production part and a robust compact design.

Application

Application	-40 to 130°C
Storage temp. range	0 to 100°C
Bio fuel compatibility	E85/M22
Max. vibration	600 m/s²

Technical Specifications

Mechanical Data

Male thread	M12x1.5
Wrench size	19 mm
Installation torque	25 Nm
Weight w/o wire	29 g
Sealing	Not included

Electrical Data

Characteristic	NTC
Nominal resistance at 20°C	2.5 kΩ ± 5 %

Characteristic

Accuracy at 25°C	± 1.4°C
Accuracy at 100°C	± 3.4°C
Response time tau 63 in still water	< 15 s