Nataraj Sunkara

Student

Vikas College of Engineering and Technology,

Nunna, Andhra Pradesh

B.V. Reddy

Associate Professor

Vikas College of Engineering and Technology,

Nunna, Andhra Pradesh
AUTOMATIC FIRE CONTROL USING ZIGBEE PROTOCOL

ABSTRACT

The fire fighting is a risky work where the fire fighters need to face the fire hazards to save life and properties. Different equipments such as fire extinguisher, axes and cutting equipment are used to fight against fire. Fire fighting vehicle is being used to transport firefighters to the scene as well as transport the fire fighting equipments. The fire fighter is still facing the risk of fire hazard because they still need to go near to fire scene. To reduce the fire fighter risk during fire incident, a fire fighting mobile robot is designed. Fire fighting mobile robot is a remote controlled machine that replace fire fighter to perform fire fighting task. The application of fire fighting mobile robot is able to reduce the direct contact of fire fighter to the fire hazards. The paper reviews a variety of technologies and state-of-the-art technology of fire fighting mobile robot. The paper also describes the first Malaysian designed and built fire fighting mobile robot, namely, MyBOT2000.
## TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>S.NO</th>
<th>CHAPTER</th>
<th>PAGE NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>EXISTING SYSTEM &amp; PROPOSED SYSTEM</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2.1 LUF60</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2.2 INDUSTRIAL AUTOMATION</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>2.3 TELECOMMUNICATIONS</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>2.4 SECURITY</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>HARDWARE TOOLS</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>3.1 POWER SUPPLY BLOCK</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>3.1.1 TRANSFORMER</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>3.1.2 RECTIFIER</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>3.1.3 FILTER</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>3.1.4 VOLTAGE REGULATOR</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>3.2 PANIC SWITCH</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>MICROCONTROLLER</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>4.1 DESCRIPTION</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>4.2 FEATURES</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>4.3 PIN CONFIGURATION</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>4.4 BLOCK DIAGRAM</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>4.5 MEMORY CONTROLLER BLOCK</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>4.5.1 OPERATION</td>
<td>19</td>
</tr>
<tr>
<td>5</td>
<td>COMPONENTS DESCRIPTION</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>5.1 LIGHT EMITTING DIODE</td>
<td>21</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Fig. NO</th>
<th>NAME OF THE FIGURE</th>
<th>PAGE NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>LUF60</td>
<td>4</td>
</tr>
<tr>
<td>3.1</td>
<td>POWER SUPPLY</td>
<td>7</td>
</tr>
<tr>
<td>3.2</td>
<td>CIRCUIT DIAGRAM OF POWER SUPPLY</td>
<td>8</td>
</tr>
<tr>
<td>3.3</td>
<td>STEP-DOWN TRANSFORMER</td>
<td>8</td>
</tr>
<tr>
<td>3.4</td>
<td>BRIDGE RECTIFIER</td>
<td>9</td>
</tr>
<tr>
<td>3.5</td>
<td>CAPACITIVE FILTER</td>
<td>10</td>
</tr>
<tr>
<td>3.6</td>
<td>7805 VOLTAGE REGULATOR</td>
<td>11</td>
</tr>
<tr>
<td>3.7</td>
<td>DIAGRAM OF POWER SUPPLY</td>
<td>12</td>
</tr>
<tr>
<td>4.1</td>
<td>PIN DIAGRAM</td>
<td>15</td>
</tr>
<tr>
<td>4.2</td>
<td>BLOCK DIAGRAM OF ARM MICROCONTROLLER</td>
<td>16</td>
</tr>
<tr>
<td>4.3</td>
<td>MEMORY MAPS</td>
<td>18</td>
</tr>
<tr>
<td>5.1</td>
<td>SYMBOL OF LED</td>
<td>22</td>
</tr>
<tr>
<td>5.2</td>
<td>LED CONNECTION TO MICROCONTROLLER</td>
<td>22</td>
</tr>
<tr>
<td>5.3</td>
<td>PUSH BUTTON</td>
<td>23</td>
</tr>
<tr>
<td>5.4</td>
<td>SYMBOL OF RELAY</td>
<td>25</td>
</tr>
<tr>
<td>5.5</td>
<td>RELAY OPERATION</td>
<td>26</td>
</tr>
<tr>
<td>5.6</td>
<td>CERAMIC CAPACITOR</td>
<td>29</td>
</tr>
<tr>
<td>5.7</td>
<td>ELECTROLYTIC CAPACITOR</td>
<td>30</td>
</tr>
<tr>
<td>5.8</td>
<td>TRANSISTOR</td>
<td>31</td>
</tr>
<tr>
<td>5.9</td>
<td>LOGIC DIAGRAM OF MAX232C</td>
<td>33</td>
</tr>
<tr>
<td>5.10</td>
<td>SIMPLE NULL MODEL WITHOUT HAND SHAKING</td>
<td>39</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

Fire fighting is risky profession. They are not only extinguishing fires in tall buildings but also must drag heavy hoses, climb high ladders and carry people from buildings and other situations. There are many fire fighters lost their lives in the line of duty each year throughout the world [1]. The statistics of the fire fighter fatalities are still maintain at high level every year and it may continue to increase if there is no improvement in fire fighting techniques and technology. In addition to working in long and irregular hours and unfriendly working environment such as high temperature, dusty and low humidity, firefighters are also facing with potentially life threatening situation such as explosion, collapsed building and radioactive. The common equipment used by firefighters such as flat head axe, halligan bar, turnout jacket, fire retardant or bunker pants, boots, flashlight, helmet, face mask, and gloves do not significantly reduce risk on their lives when facing those life threatening situations.

IAFF (2000) reported that there are 1.9 firefighters are killed per year in the USA, per 100,000 structure fires. However, this rate was increasing to 3.0 per 100,000 structure fires across a thirty year period which is peaking in the 1990s [12]. There are many causes for Line of Duty Deaths (LODD) such as smoke inhalation, burns, crushing injuries and related trauma [17]. The death of on duty fire fighter in US during 1977-2010 year is shown in the statistic in Figure 1. From the statistic, it shows that the death of the fire fighter almost constant every year. Therefore, there is a need of fire fighting machine in assisting the fire fighter to avoid death by handling some dangerous situations.
As a result of this, over the past few years, research and development on firefighting technology is extensively made around the world especially in US, Japan, and a number of European companies. There were many studies (Shanghai Qiangshi Firefighting Equipment co., Ltd., 2007) had emphasized on machine development to replace fire fighter to fight fire in dangerous situations and to reduce the fire fighter risk [3] [19]. The machines help the fire fighter using extinguishing agent such as water, foam or others without fire fighter having to set up or operate directly in danger areas. Amano (2002) highlighted the weaknesses of existing machine design and suggest integration of all important elements in developing fire fighting machine so that a successful rescuing process can be achieved [1]. These elements are size, weight, cost and performance.
2. EXISTING SYSTEM & PROPOSED SYSTEM

The rapid development in technology nowadays, directly improve the tools and equipment used in fire fighting. With these advance tools and equipment, fire fighting can be more effective and efficient. Besides, it also reduces the risk to minimum level. Moreover, the damages of fire incident can be reduced. Fire fighting mobile robot is one of the solution that able to reduce the fire hazards risk on fire fighter. Different type of current available fire fighting machine will be described.

2.1 LUF60:

LUF60 (Figure 1) is a diesel powered mobile fire fighting machine is equipped with air blower and a water beam fog. The machine is able to clear the hazardous obstacles by blowing the mixture of air and water. Besides, the machine is designed to withstand the rigor operating conditions and confined spaces. The LUF60 can be used in rail tunnels, aircraft hangers, parking garages, chemical plant, etc. There are some significant features of this machine. The monitor nozzle has a flow rate up to 800 GPM and it can blow the water beam as far as 80m. In order to enhance its mobility in high temperature condition, it is equipped with rubber track system which is rated to 400 degrees Fahrenheit. The rubber track system enables the robot to climb the stair. Besides, this machine able to operate on slopes of maximum 20 degrees with the ventilation tube at a maximum 45 degree angle [15].
2.2 Industrial automation:

Today a lot of industries use embedded systems for process control. These include pharmaceutical, cement, sugar, oil exploration, nuclear energy, electricity generation and transmission [6]. The embedded systems for industrial use are designed to carry out specific tasks such as monitoring the temperature, pressure, humidity, voltage, current etc., and then take appropriate action based on the monitored levels to control other devices or to send information to a centralized monitoring station. In hazardous industrial environment, where human presence has to be avoided, robots are used, which are programmed to do specific jobs. The robots are now becoming very powerful and carry out many interesting and complicated tasks such as hardware assembly [4].

2.3 Telecommunications:

In the field of telecommunications, the embedded systems can be categorized as subscriber terminals and network equipment. The subscriber terminals such as key telephones, ISDN phones, terminal adapters, web cameras are embedded systems.
The network equipment includes multiplexers, multiple access systems, Packet Assemblers Dissemblers (PADs), satellite modems etc. IP phone, IP gateway, IP gatekeeper etc. are the latest embedded systems that provide very low-cost voice communication over the Internet.

2.4 Security:

Security of persons and information has always been a major issue. We need to protect our homes and offices; and also the information we transmit and store. Developing embedded systems for security applications is one of the most lucrative businesses nowadays. Security devices at homes, offices, airports etc. for authentication and verification are embedded systems. Encryption devices are nearly 99 per cent of the processors that are manufactured end up in embedded systems. Embedded systems find applications in every industrial segment-consumer electronics, transportation, avionics, biomedical engineering, manufacturing, process control and industrial automation, data communication, telecommunication, defense, security etc. Used to encrypt the data/voice being transmitted on communication links such as telephone lines. Biometric systems using fingerprint and face recognition are now being extensively used for user authentication in banking applications as well as for access control in high security buildings.

2.5 PROPOSED SYSTEM

We in proposed to I am using alarm control to access and control the process fast and ZigBee is new Protocol, which access wish speed data and control the process.
2.5.1 Block Diagram of Proposed System Transmission End:

**Transmission Board**

![Transmission Diagram](image1)

2.5.2 Block Diagram of Proposed System receiver End:

**Receiver Board**

![Receiver Diagram](image2)
3. HARDWARE TOOLS

3.1 POWER SUPPLY BLOCK:

The input to the circuit is applied from the regulated power supply. The a.c. input i.e., 230V from the mains supply is step down by the transformer to 12V and is fed to a rectifier. The output obtained from the rectifier is a pulsating d.c voltage. So in order to get a pure d.c voltage, the output voltage from the rectifier is fed to a filter to remove any a.c components present even after rectification. Now, this voltage is given to a voltage regulator to obtain a pure constant dc voltage.

Fig 3.1: Power supply
3.1.1 Transformer:

Usually, DC voltages are required to operate various electronic equipment and these voltages are 5V, 9V or 12V. But these voltages cannot be obtained directly. Thus the a.c input available at the mains supply i.e., 230V is to be brought down to the required voltage level. This is done by a transformer. Thus, a step down transformer is employed to decrease the voltage to a required level.
3.1.2 Rectifier:

The output from the transformer is fed to the rectifier. It converts A.C. into pulsating D.C. The rectifier may be a half wave or a full wave rectifier. In this project, a bridge rectifier is used which converts an ac voltage to dc voltage using both half cycles of the input ac voltage. The Bridge rectifier circuit is shown in the figure. The circuit has four diodes connected to form a bridge. The ac input voltage is applied to the diagonally opposite ends of the bridge. The load resistance is connected between the other two ends of the bridge. For the positive half cycle of the input ac voltage, diodes D1 and D3 conduct, whereas diodes D2 and D4 remain in the OFF state. The conducting diodes will be in series with the load resistance RL and hence the load current flows through RL.

Fig 3.4: Output Waveforms of Bridge Rectifier
For the negative half cycle of the input ac voltage, diodes D2 and D4 conduct whereas, D1 and D3 remain OFF. The conducting diodes D2 and D4 will be in series with the load resistance RL and hence the current flows through RL in the same direction as in the previous half cycle. Thus a bi-directional wave is converted into a unidirectional wave.

3.1.3 Filter:

Capacitive filter is used in this project. It removes the ripples from the output of rectifier and smoothenes the D.C. Output received from this filter is constant until the mains voltage and load is maintained constant. However, if either of the two is varied, D.C. voltage received at this point changes. Therefore a regulator is applied at the output stage.

![Capacitive Filter](image)

Fig 3.5 Capacitive Filter

3.1.4 Voltage regulator:

As the name itself implies, it regulates the input applied to it. A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. In this project, power supply of 5V and 12V are required. In order to obtain these voltage levels, 7805 and 7812 voltage regulators are to be used.
The first number 78 represents positive supply and the numbers 05, 12 represent the required output voltage levels. If adequate heat sinking is provided, they can deliver over 1 A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltage and currents.

This module is basically designed to achieve 12V, 1A and 5V, 500mA and 3.3V. The design consists of a transformer which is used to step down the AC voltage, IN4007 diodes used to form a bridge rectifier to convert AC to DC, capacitor 1000uF which used as a filter circuit, 7812 regulator to obtain a 12V DC and
followed by 7805 regulator to obtain a 5V DC, at the output of the regulator a 330 ohm resistance and LED is connected as Power ON indicator. LT1086CT (3.3V) regulator is used to generate 3.3V which is required for Ethernet Controller is as shown in figure 2.

![Diagram of Power Supply section](image)

**Fig 3.7:** Diagram of Power Supply section

### 3.2 Panic Switch:

This is nothing but a simple switch which is connected in the sensor board. The arrangement of this is as shown in the figure below.

The response of this switch is monitored by the microcontroller and the corresponding action takes place. This can be activated by pressing this switch which will lead to the activation of buzzer and auto-dialer.
4. MICROCONTROLLER

Microprocessors and microcontrollers are widely used in embedded systems products. Microcontroller is a programmable device. A microcontroller has a CPU in addition to a fixed amount of RAM, ROM, I/O ports and a timer embedded all on a single chip. The fixed amount of on-chip ROM, RAM and number of I/O ports in microcontrollers makes them ideal for many applications in which cost and space are critical.

The present project is implemented on Keil uVision. In order to program the device, flash magic has been used to burn the program onto the microcontroller. The pin description, features of the microcontroller and the software tools used are discussed in the following sections.

4.1 DESCRIPTION:

The LPC2141/2/4/6/8 microcontrollers are based on a 32/16 bit ARM7TDMI-S CPU with real-time emulation and embedded trace support, that combines the microcontroller with embedded high speed flash memory ranging from 32 kB to 512 kB. A 128-bit wide memory interface and a unique accelerator architecture enable 32-bit code execution at the maximum clock rate. For critical code size applications, the alternative 16-bit Thumb mode reduces code by more than 30 % with minimal performance penalty. Due to their tiny size and low power consumption, LPC2141/2/4/6/8 are ideal for applications where miniaturization is a key requirement, such as access control and point-of-sale. A blend of serial communications interfaces ranging from a USB 2.0 Full Speed device, multiple UARTs, SPI, SSP to I2Cs, and on-chip SRAM of 8 kB up to 40 kB, make these devices very well suited for communication gateways and protocol converters, soft
modems, voice recognition and low end imaging, providing both large buffer size and high processing power. Various 32-bit timers, single or dual 10-bit ADC(s), 10-bit DAC, PWM channels and 45 fast GPIO lines with up to nine edge or level sensitive external interrupt pins make these microcontrollers particularly suitable for industrial control and medical systems.

4.2 FEATURES

• 16/32-bit ARM7TDMI-S microcontroller in a tiny LQFP64 package.
• 8 to 40 kB of on-chip static RAM and 32 to 512 kB of on-chip flash program memory.
• 128 bit wide interface/accelerator enables high speed 60 MHz operation.
• In-System/In-Application Programming (ISP/IAP) via on-chip boot-loader software.
• Single flash sector or full chip erase in 400 ms and programming of 256 bytes in 1 ms.
• Embedded ICE RT and Embedded Trace interfaces offer real-time debugging with the on-chip Real Monitor software and high speed tracing of instruction execution.
• USB 2.0 Full Speed compliant Device Controller with 2 kB of endpoint RAM.
• In addition, the LPC2146/8 provide 8 kB of on-chip RAM accessible to USB by DMA.
• One or two (LPC2141/2 vs. LPC2144/6/8) 10-bit A/D converters provide a total of 6/14 analog inputs, with conversion times as low as 2.44 s per channel.
• Single 10-bit D/A converter provides variable analog output.
• Two 32-bit timers/external event counters (with four capture and four compare channels each), PWM unit (six outputs) and watchdog.
- Low power real-time clock with independent power and dedicated 32 kHz clock input.
- Multiple serial interfaces including two UARTs (16C550), two Fast I2C-bus (400 kbit/s), SPI and SSP with buffering and variable data length capabilities.
- Vectored interrupt controller with configurable priorities and vector addresses.
- Up to 45 of 5 V tolerant fast general purpose I/O pins in a tiny LQFP64 package.

4.3 PIN CONFIGURATION:

Fig 4.1: Pin Diagram
4.4 Block Diagram:

Fig 4.2: Block Diagram of ARM Microcontroller
DESCRIPTION:

LPC2141/2/4/6/8 has two 32-bit General Purpose I/O ports. Total of 30 input/output and a single output only pin out of 32 pins are available on PORT0. PORT1 has up to 16 pins available for GPIO functions. PORT0 and PORT1 are controlled via two groups of 4 registers. Legacy registers allow backward compatibility with earlier family devices, using existing code.

The functions and relative timing of older GPIO implementations is preserved.

The registers represent the enhanced GPIO features available on the LPC2141/2/4/6/8. All of these registers are located directly on the local bus of the CPU for the fastest possible read and write timing. An additional feature has been added that provides byte addressability of all GPIO registers. A mask register allows treating groups of bits in a single GPIO port separately from other bits on the same port. The user must select whether a GPIO will be accessed via registers that provide enhanced features or a legacy set of registers.

While both of a port’s fast and legacy GPIO registers are controlling the same physical pins, these two port control branches are mutually exclusive and operate independently. For example, changing a pin’s output via a fast register will not be observable via the corresponding legacy register.
Memory maps:

![Memory Map Diagram]

Fig 4.3: Memory Maps

4.5 MEMORY CONTROLL BLOCK:

The MAM block in the LPC2141/2/4/6/8 maximizes the performance of the ARM processor when it is running code in Flash memory, but does so using a single Flash bank.
4.5.1 Operation:

Simply put, the Memory Accelerator Module (MAM) attempts to have the next ARM instruction that will be needed in its latches in time to prevent CPU fetch stalls. The LPC2141/2/4/6/8 uses one bank of Flash memory, compared to the two banks used on predecessor devices. It includes three 128-bit buffers called the Prefetch Buffer, the Branch Trail Buffer and the data buffer. When an Instruction Fetch is not satisfied by either the Prefetch or Branch Trail Buffer, nor has a prefetch been initiated for that line, the ARM is stalled while a fetch is initiated for the 128-bit line. If a prefetch has been initiated but not yet completed, the ARM is stalled for a shorter time. Unless aborted by a data access, a prefetch is initiated as soon as the Flash has completed the previous access. The prefetched line is latched by the Flash module, but the MAM does not capture the line in its prefetch buffer until the ARM core presents the address from which the prefetch has been made. If the core presents a different address from the one from which the prefetch has been made, the prefetched line is discarded. The Prefetch and Branch Trail buffers each include four 32-bit ARM instructions or eight 16-bit Thumb instructions. During sequential code execution, typically the Prefetch Buffer contains the current instruction and the entire Flash line that contains it.

The MAM differentiates between instruction and data accesses. Code and data accesses use separate 128-bit buffers. 3 of every 4 sequential 32-bit code or data accesses "hit" in the buffer without requiring a Flash access (7 of 8 sequential 16-bit accesses, 15 of every 16 sequential byte accesses). The fourth (eighth, 16th) sequential data access must access Flash, aborting any prefetch in progress. When a Flash data access is concluded, any prefetch that had been in progress is re-initiated. Timing of Flash read operations is programmable and is described later in this section. In this manner, there is no code fetch penalty for sequential instruction
execution when the CPU clock period is greater than or equal to one fourth of the Flash access time. The average amount of time spent doing program branches is relatively small (less than 25%) and may be minimized in ARM (rather than Thumb) code through the use of the conditional execution feature present in all ARM instructions. This conditional execution may often be used to avoid small forward branches that would otherwise be necessary.

Branches and other program flow changes because a break in the sequential flow of instruction fetches described above. The Branch Trail Buffer captures the line to which such a non-sequential break occurs. If the same branch is taken again, the next instruction is taken from the Branch Trail Buffer. When a branch outside the contents of the Prefetch and Branch Trail Buffer is taken, a stall of several clocks is needed to load the Branch Trail buffer. Subsequently, there will typically be no further instruction fetch delays until a new and different branch occurs.
5. COMPONENTS DESCRIPTION

5.1 Light-emitting diode (LED):

Light-emitting diodes are elements for light signalization in electronics. They are manufactured in different shapes, colors and sizes. For their low price, low consumption and simple use, they have almost completely pushed aside other light sources—bulbs at first place. They perform similar to common diodes with the difference that they emit light when current flows through them. Light emitting diodes (LEDs) are semiconductor light sources. The light emitted from LEDs varies from visible to infrared and ultraviolet regions. They operate on low voltage and power. LEDs are one of the most common electronic components and are mostly used as indicators in circuits. They are also used for luminance and optoelectronic applications.

Based on semiconductor diode, LEDs emit photons when electrons recombine with holes on forward biasing. The two terminals of LEDs are anode (+) and cathode (-) and can be identified by their size. The longer leg is the positive terminal or anode and shorter one is negative terminal. The forward voltage of LED (1.7V-2.2V) is lower than the voltage supplied (5V) to drive it in a circuit. Using an LED as such would burn it because a high current would destroy its p-n gate. Therefore a current limiting resistor is used in series with LED. Without this resistor, either low input voltage (equal to forward voltage) or PWM (pulse width modulation) is used to drive the LED.
It is important to know that each diode will be immediately destroyed unless its current is limited. This means that a conductor must be connected in parallel to a diode. In order to correctly determine value of this conductor, it is necessary to know diode’s voltage drop in forward direction, which depends on what material a diode is made of and what color it is. Values typical for the most frequently used diodes are shown in table below: There are three main types of LEDs. *Standard* ones get full brightness at current of 20mA. Low Current diodes get full brightness at ten time’s lower current while Super Bright diodes produce more intensive light than Standard ones.
Since the 8051 microcontrollers can provide only low input current and since their pins are configured as outputs when voltage level on them is equal to 0, direct connecting to LEDs is carried out as it is shown on figure (Low current LED, cathode is connected to output pin).

5.2 Switches and Pushbuttons:

There is nothing simpler than this! This is the simplest way of controlling appearance of some voltage on microcontroller’s input pin. There is also no need for additional explanation of how these components operate. Nevertheless, it is not so simple in practice... This is about something commonly unnoticeable when using these components in everyday life. It is about contact bounce- a common problem with mechanical switches. If contact switching does not happen so quickly, several consecutive bounces can be noticed prior to maintain stable state.

![Fig 5.3: Push Button](image)

The reasons for this are: vibrations, slight rough spots and dirt. Anyway, whole this process does not last long (a few micro- or milliseconds), but long enough to be registered by the microcontroller. Concerning pulse counter, error occurs in almost 100% of cases!
The simplest solution is to connect simple RC circuit which will “suppress” each quick voltage change. Since the bouncing time is not defined, the values of elements are not strictly determined. In the most cases, the values shown on figure are sufficient.

If complete safety is needed, radical measures should be taken! The circuit, shown on the figure (RS flip-flop), changes logic state on its output with the first pulse triggered by contact bounce. Even though this is more expensive solution (SPDT switch), the problem is definitely resolved! Besides, since the compensator is not used, very short pulses can be also registered in this way.

In addition to these hardware solutions, a simple software solution is commonly applied too: when a program tests the state of some input pin and finds changes, the check should be done one more time after certain time delay. If the change is confirmed it means that switch (or pushbutton) has changed its position. The advantages of such solution are obvious: it is free of charge, effects of disturbances are eliminated too and it can be adjusted to the worst-quality contacts.
5.3 RELAYS:

A relay is an electrically controllable switch widely used in industrial controls, automobiles and appliances. The relay allows the isolation of two separate sections of a system with two different voltage sources i.e., a small amount of voltage/current on one side can handle a large amount of voltage/current on the other side but there is no chance that these two voltages mix up.

![Circuit symbol of a relay](image)

**Fig 5.4: Circuit symbol of a relay**

5.3.1 Operation:

When current flows through the coil, a magnetic field are created around the coil i.e., the coil is energized. This causes the armature to be attracted to the coil. The armature’s contact acts like a switch and closes or opens the circuit. When the coil is not energized, a spring pulls the armature to its normal state of open or closed. There are all types of relays for all kinds of applications.
Transistors and ICs must be protected from the brief high voltage 'spike' produced when the relay coil is switched off. The above diagram shows how a signal diode (eg 1N4148) is connected across the relay coil to provide this protection. The diode is connected 'backwards' so that it will normally not conduct. Conduction occurs only when the relay coil is switched off, at this moment the current tries to flow continuously through the coil and it is safely diverted through the diode. Without the diode no current could flow and the coil would produce a damaging high voltage 'spike' in its attempt to keep the current flowing.

In choosing a relay, the following characteristics need to be considered:

1. The contacts can be normally open (NO) or normally closed (NC). In the NC type, the contacts are closed when the coil is not energized. In the NO type, the contacts are closed when the coil is energized.

2. There can be one or more contacts. i.e., a different type like SPST (single pole single throw), SPDT (single pole double throw) and DPDT (double pole double throw) relays.
3. The voltage and current required to energize the coil. The voltage can vary from a few volts to 50 volts, while the current can be from a few milliamps to 20 milliamps. The relay has a minimum voltage, below which the coil will not be energized. This minimum voltage is called the “pull-in” voltage.

4. The minimum DC/AC voltage and current that can be handled by the contacts. This is in the range of a few volts to hundreds of volts, while the current can be from a few amps to 40A or more, depending on the relay.

5.4 RESISTORS:

Resistors (R) are the most commonly used of all electronic components, to the point where they are almost taken for granted. There are many different resistor types available with their principal job being to "resist" the flow of current through an electrical circuit, or to act as voltage droppers or voltage dividers. They are "Passive Devices", that is they contain no source of power or amplification but only attenuate or reduce the voltage signal passing through them. When used in DC circuits the voltage drop produced is measured across their terminals as the circuit current flows through them while in AC circuits the voltage and current are both in-phase producing 0° phase shift. Resistors produce a voltage drop across themselves when an electrical current flow through them because they obey Ohm’s Law and different values of resistance produces different values of current or voltage.

Ohm’s law:

To make a current flow through a resistance there must be a voltage across that resistance.
Ohm's Law shows the relationship between the voltage (V), current (I) and resistance (R).

It can be written in three ways:

\[
V = I \times R \quad I = \frac{V}{R} \quad R = \frac{V}{I}
\]

Where:  
- \( V \) = voltage in volts (V)  
- \( I \) = current in amps (A)  
- \( R \) = resistance in ohms (\( \Omega \))

Some of the common characteristics associated with the humble resistor are; Temperature Coefficient, Voltage Coefficient, Noise, Frequency Response, Power as well as Temperature Rating, Physical Size and Reliability.

**Resistor symbol:**

![Resistor Symbol](image)

**Fig: Resistor symbol**

**5.5 Capacitors:**

The capacitor’s function is to store electricity, or electrical energy. The capacitor also functions as a filter, passing alternating current (AC), and blocking direct current (DC). This symbol ‘F’ is used to indicate a capacitor in a circuit diagram. The capacitor is constructed with two electrode plates facing each other, but separated by an insulator. When DC voltage is applied to the capacitor, an electric charge is stored on each electrode. While the capacitor is charging up, current flows. The current will stop flowing when the capacitor has fully charged.
Symbol:

\[ + \quad \bigg| \quad - \]

Symbol of Capacitor

The symbol of capacitor is shown and the two terminals both positive and negative terminals are also shown in the diagram

5.5.1 Ceramic Capacitors:

Ceramic capacitors are constructed with materials such as titanium acid barium used as the dielectric. Internally, these capacitors are not constructed as a coil, so they can be used in high frequency applications. Typically, they are used in circuits which bypass high frequency signals to ground. These capacitors have the shape of a disk. Their capacitance is comparatively small. The capacitor on the left is a 100pF capacitor with a diameter of about 3 mm. The capacitor on the right side is printed with 103, so \(10 \times 10^3\) pF becomes 0.01 µF. The diameter of the disk is about 6 mm. Ceramic capacitors have no polarity. Ceramic capacitors should not be used for analog circuits, because they can distort the signal. Different types of ceramic capacitors used as shown

![Fig 5.6: Ceramic capacitors](image-url)
Ceramic capacitors are normally used for radio frequency and some audio applications. Ceramic capacitors range in value from figures as low as a few picofarads to around 0.1 microfarads.

5.5.2 Electrolytic Capacitors (Electrochemical type capacitors):

Aluminum is used for the electrodes by using a thin oxidization membrane. Large values of capacitance can be obtained in comparison with the size of the capacitor, because the dielectric used is very thin. The most important characteristic of electrolytic capacitors is that they have polarity. They have a positive and a negative electrode. [Polarized] This means that it is very important which way round they are connected.

If the capacitor is subjected to voltage exceeding its working voltage, or if it is connected with incorrect polarity, it may burst. It is extremely dangerous, because it can quite literally explode. Make absolutely no mistakes. Generally, in the circuit diagram, the positive side is indicated by a "+" (plus) symbol. Electrolytic capacitors range in value from about 1\(\mu\)F to thousands of \(\mu\)F. Mainly this type of capacitor is used as a ripple filter in a power supply circuit, or as a filter to bypass low frequency signals, etc. Because this type of capacitor is
comparatively similar to the nature of a coil in construction, it isn't possible to use for high-frequency circuits. Different types of electrolytic capacitors used as shown.

5.6 TRANSISTOR:

A Transistor is a semiconductor device used to amplify and switch electronic signals. It is made of a solid piece of semiconductor material, with at least three terminals for connection to an external circuit. A voltage or current applied to one pair of the transistor's terminals changes the current flowing through another pair of terminals. Because the controlled (output) power can be much more than the controlling (input) power, the transistor provides amplification of a signal. The transistor is the fundamental building block of modern electronic devices. A transistor can control its output in proportion to the input signal; that is, it can act as an amplifier. Alternatively, the transistor can be used to turn current on or off in a circuit as an electrically controlled switch, where the amount of current is determined by other circuit elements. Modern transistor audio amplifiers of up to a few hundred watts are common and relatively inexpensive.

![Fig 5.8: Transistors](image_url)
Transistors amplify current, for example they can be used to amplify the small output current from a logic IC so that it can operate a lamp, relay or other high current device.

In many circuits a resistor is used to convert the changing current to a changing voltage, so the transistor is being used to amplify voltage. A transistor may be used as a switch (either fully on with maximum current, or fully off with no current) and as an amplifier (always partly on).

5.7 MAX 232C:

The TTL signals output by a USART are not suitable for transmission over long distances, so these signals are converted to some other form to be transmitted. With a jumper between the points numbered 7 and 8, a high on the TX output of the 8251A produces a high on the base of the transistor, which turns it off. With points numbered 9 and 10 numbered, the CR TX line will then be pulled to -12V, which is a legal high or marking condition for RS-232C. A low on the TXD output of the 8251A will turn on the transistor and pull the CR TX line to +5V, which is legal low or space condition for RS-232C.
5.8 Introduction To RS232 Serial Communication

Serial communication is basically the transmission or reception of data one bit at a time. Today's computers generally address data in bytes or some multiple thereof. A byte contains 8 bits. A bit is basically either a logical 1 or zero. Every character on this page is actually expressed internally as one byte. The serial port is used to convert each byte to a stream of ones and zeroes as well as to convert a streams of ones and zeroes to bytes. The serial port contains a electronic chip called a Universal Asynchronous Receiver/Transmitter (UART) that actually does the conversion.

The serial port has many pins. We will discuss the transmit and receive pin first. Electrically speaking, whenever the serial port sends a logical one (1) a negative voltage is effected on the transmit pin. Whenever the serial port sends a logical zero (0) a positive voltage is affected. When no data is being sent, the serial port's
transmit pin's voltage is negative (1) and is said to be in a **MARK** state. Note that the serial port can also be forced to keep the transmit pin at a positive voltage (0) and is said to be the **SPACE** or **BREAK** state. (The terms **MARK** and **SPACE** are also used to simply denote a negative voltage (1) or a positive voltage (0) at the transmit pin respectively).

When transmitting a byte, the UART (serial port) first sends a **START BIT** which is a positive voltage (0), followed by the data (general 8 bits, but could be 5, 6, 7, or 8 bits) followed by one or two **STOP BITs** which is a negative (1) voltage. The sequence is repeated for each byte sent.

At this point you may want to know what the duration of a bit is. In other words, how long does the signal stay in a particular state to define a bit? The answer is simple. It is dependent on the baud rate. The baud rate is the number of times the signal can switch states in one second. Therefore, if the line is operating at 9600 baud, the line can switch states 9,600 times per second. This means each bit has the duration of $\frac{1}{9600}$ of a second or about 100 $\mu$sec.

When transmitting a character there are other characteristics other than the baud rate that must be known or that must be setup. These characteristics define the entire interpretation of the data stream.

The first characteristic is the length of the byte that will be transmitted. This length
in general can be anywhere from 5 to 8 bits. The second characteristic is parity. The parity characteristic can be even, odd, mark, space, or none. If even parity, then the last data bit transmitted will be a logical 1 if the data transmitted had an even amount of 0 bits. If odd parity, then the last data bit transmitted will be a logical 1 if the data transmitted had an odd amount of 0 bits. If \textbf{MARK} parity, then the last transmitted data bit will always be a logical 1. If \textbf{SPACE} parity, then the last transmitted data bit will always be a logical 0. If no parity then there is no parity bit transmitted.

The third characteristic is the amount of stop bits. This value in general is 1 or 2. Assume we want to send the letter 'A' over the serial port. The binary representation of the letter 'A' is 01000001. Remembering that bits are transmitted from least significant bit (LSB) to most significant bit (MSB), the bit stream transmitted would be as follows for the line characteristics 8 bits, no parity, 1 stop bit, and 9600 baud.

\textbf{LSB (0 1 0 0 0 0 0 1 0 1) MSB}

The above represents (Start Bit) (Data Bits) (Stop Bit)

To calculate the actual byte transfer rate simply divide the baud rate by the number of bits that must be transferred for each byte of data. In the case of the above example, each character requires 10 bits to be transmitted for each character. As such, at 9600 baud, up to 960 bytes can be transferred in one second.

The above discussion was concerned with the "electrical/logical" characteristics of the data stream. We will expand the discussion to line protocol. Serial communication can be half duplex or full duplex. Full duplex communication means that a device can receive and transmit data at the same time. Half duplex means that
the device cannot send and receive at the same time. It can do them both, but not at the same time. Half duplex communication is all but outdated except for a very small focused set of applications.

Half duplex serial communication needs at a minimum two wires, signal ground and the data line. Full duplex serial communication needs at a minimum three wires, signal ground, transmit data line, and receive data line. The RS232 specification governs the physical and electrical characteristics of serial communications. This specification defines several additional signals that are asserted (set to logical 1) for information and control beyond the data signals and signal ground.

These signals are the Carrier Detect Signal (CD), asserted by modems to signal a successful connection to another modem, Ring Indicator (RI), asserted by modems to signal the phone ringing, Data Set Ready (DSR), asserted by modems to show their presence, Clear To Send (CTS), asserted by modems if they can receive data, Data Terminal Ready (DTR), asserted by terminals to show their presence, Request To Send (RTS), asserted by terminals if they can receive data. The section RS232 Cabling describes these signals and how they are connected.

The above paragraph eluded to hardware flow control. Hardware flow control is a method that two connected devices use to tell each other electronically when to send or when not to send data. A modem in general drops (logical 0) its CTS line when it can no longer receive characters. It re-asserts it when it can receive again. A terminal does the same thing instead with the RTS signal.
Note that hardware flow control requires the use of additional wires. The benefit to this however is crisp and reliable flow control. Another method of flow control used is known as software flow control. This method requires a simple 3 wire serial communication link, transmit data, receive data, and signal ground. If using this method, when a device can no longer receive, it will transmit a character that the two devices agreed on. This character is known as the XOFF character. This character is generally a hexadecimal 13. When a device can receive again it transmits an XON character that both devices agreed to. This character is generally a hexadecimal 11.

**NULL MODEM, AN INTRODUCTION**

Serial communications with RS232. One of the oldest and most widely spread communication methods in computer world. The way this type of communication can be performed is pretty well defined in standards. I.e. with one exception. The standards show the use of DTE/DCE communication, the way a computer should communicate with a peripheral device like a modem. For your information, **DTE** means *data terminal equipment* (computers etc.) where **DCE** is the abbreviation of *data communication equipment* (modems). One of the main uses of serial communication today where no modem is involved—a *serial null modem* configuration with DTE/DTE communication—is not so well defined, especially when it comes to flow control. The terminology *null modem* for the situation where two computers communicate directly is so often used nowadays, that most people don't realize anymore the origin of the phrase and that a null modem connection is an exception, not the rule.

In history, practical solutions were developed to let two computers talk with each other using a null modem serial communication line. In most situations, the original
modem signal lines are reused to perform some sort of handshaking. Handshaking can increase the maximum allowed communication speed because it gives the computers the ability to control the flow of information. High amounts of incoming data is allowed if the computer is capable to handle it, but not if it is busy performing other tasks. If no flow control is implemented in the null modem connection, communication is only possible at speeds at which it is sure the receiving side can handle the amount information even under worst case conditions.

5.8.1 Original use of RS232

When we look at the connector pinout of the RS232 port, we see two pins which are certainly used for flow control. These two pins are RTS, request to send and CTS, clear to send. With DTE/DCE communication (i.e. a computer communicating with a modem device) RTS is an output on the DTE and input on the DCE. CTS is the answering signal coming from the DCE.

Before sending a character, the DTE asks permission by setting its RTS output. No information will be sent until the DCE grants permission by using the CTS line. If the DCE cannot handle new requests, the CTS signal will go low. A simple but useful mechanism allowing flow control in one direction. The assumption is, that the DTE can always handle incoming information faster than the DCE can send it. In the past, this was true. Modem speeds of 300 baud were common and 1200 baud was seen as a high speed connection.

For further control of the information flow, both devices have the ability to signal their status to the other side. For this purpose, the DTR data terminal ready and DSR data set ready signals are present. The DTE uses the DTR signal to signal that
it is ready to accept information, whereas the DCE uses the DSR signal for the same purpose. Using these signals involves not a small protocol of requesting and answering as with the RTS/CTS handshaking. These signals are in one direction only.

The last flow control signal present in DTE/DCE communication is the CD carrier detect. It is not used directly for flow control, but mainly an indication of the ability of the modem device to communicate with its counterpart. This signal indicates the existence of a communication link between two modem devices.

**Null modem without handshaking**

How to use the handshaking lines in a null modem configuration? The simplest way is to don't use them at all. In that situation, only the data lines and signal ground are cross connected in the null modem communication cable. All other pins have no connection. An example of such a null modem cable without handshaking can be seen in the figure below.

![Null Modem Diagram](image)

Fig 5.10 Simple null modem without handshaking

<table>
<thead>
<tr>
<th>Connector 1</th>
<th>Connector 2</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>Rx</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>Tx → Rx</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>Signal ground</td>
</tr>
</tbody>
</table>
5.8.2 Compatibility issues

If you read about null modems, this three wire null modem cable is often talked about. Yes, it is simple but can we use it in all circumstances? There is a problem, if either of the two devices checks the DSR or CD inputs. These signals normaly define the ability of the other side to communicate. As they are not connected, their signal level will never go high. This might cause a problem.

The same holds for the RTS/CTS handshaking sequence. If the software on both sides is well structured, the RTS output is set high and then a waiting cycle is started until a ready signal is received on the CTS line. This causes the software to hang because no physical connection is present to either CTS line to make this possible. The only type of communication which is allowed on such a null modem line is data-only traffic on the cross connected Rx/Tx lines.

This does however not mean, that this null modem cable is useless. Communication links like present in the Norton Commander program can use this null modem cable. This null modem cable can also be used when communicating with devices which do not have modem control signals like electronic measuring equipment etc. As you can imagine, with this simple null modem cable no hardware flow control can be implemented. The only way to perform flow control is with software flow control using the XOFF and XON characters.
6. SOFTWARE TOOLS

6.1 KEIL SOFTWARE:

Keil compiler is a software used where the machine language code is written and compiled. After compilation, the machine source code is converted into hex code which is to be dumped into the microcontroller for further processing. Keil compiler also supports C language code.

6.1.1 STEPS TO WRITE AN ASSEMBLY LANGUAGE PROGRAM IN KEIL AND HOW TO COMPILE IT:

1. Install the Keil Software in the PC in any of the drives.
2. After installation, an icon will be created with the name “Keil uVision3”.
   Just drag this icon onto the desktop so that it becomes easy whenever you try to write programs in keil.
3. Double click on this icon to start the keil compiler.
4. A page opens with different options in it showing the project workspace at the leftmost corner side, output window in the bottom and an ash coloured space for the program to be written.
5. Now to start using the keil, click on the option “project”.
6. A small window opens showing the options like new project, import project, open project etc. Click on “New project”.
7. Now click on “File” and in that “New”. A new page opens and you can start writing program in it.
8. After the program is completed, save it with any name but with the .asm extension. Save the program in the file you have created earlier.
9. You can notice that after you save the program, the predefined keywords will be highlighted in bold letters.
10. Now add this file to the target by giving a right click on the source group. A list of options open and in that select “Add files to the source group”. Check for this file where you have saved and add it.

11. Right click on the target and select the first option “Options for target”. A window opens with different options like device, target, output etc. First click on “target”.

12. Since the set frequency of the microcontroller is 11.0592 MHz to interface with the PC, just enter this frequency value in the Xtal (MHz) text area and put a tick on the Use on-chip ROM. This is because the program what we write here in the keil will later be dumped into the microcontroller and will be stored in the inbuilt ROM in the microcontroller.

13. Now click the option “Output” and give any name to the hex file to be created in the “Name of executable” text area and put a tick to the “Create HEX file” option present in the same window. The hex file can be created in any of the drives. You can change the folder by clicking on “Select folder for Objects”.

14. Now to check whether the program you have written is errorless or not, click on the icon exactly below the “Open file” icon which is nothing but Build Target icon. You can even use the shortcut key F7 to compile the program written.

15. To check for the output, there are several windows like serial window, memory window, project window etc. Depending on the program you have written, select the appropriate window to see the output by entering into debug mode.

16. The icon with the letter “d” indicates the debug mode.

17. Click on this icon and now click on the option “View” and select the appropriate window to check for the output.
18. After this is done, click the icon “debug” again to come out of the debug mode.

19. The hex file created as shown earlier will be dumped into the microcontroller with the help of another software called Proload.

A small window with the title bar “Create new project” opens. The window asks the user to give

6.2 WORKING WITH KEIL:

1. Click on the Keil u Vision Icon on Desktop

2. The following fig will appear
3. Click on the Project menu from the title bar

4. Then Click on New Project.

5. Save the Project by typing suitable project name with no extension in your own folder sited in either C:\ or D:\
6. Then Click on Save button above.
7. **Select** the component for your project. i.e. Atmel……
8. Click on the + Symbol beside of Atmel
9. Select AT89C51 as shown below
10. Then Click on “OK”

11. The Following fig will appear

![Image of the interface showing a dialog box with options Yes and No]

12. Then Click either YES or NO………mostly “NO”

13. Now your project is ready to USE

14. Now double click on the Target1, you would get another option “Source group 1” as shown in next page.
15. Click on the file option from menu bar and select “new”
16. The next screen will be as shown in next page, and just maximize it by double clicking on its blue boarder.
17. Now start writing program in either in “C” or “ASM”

18. For a program written in Assembly, then save it with extension “.asm” and for “C” based program save it with extension “.C”

19. Now right click on Source group 1 and click on “Add files to Group Source”
20. Now you will get another window, on which by default “C” files will appear.

21. Click only one time on option “ADD”
22. Now Press function key F7 to compile. Any error will appear if so happen.

23. If the file contains no error, then press Control+F5 simultaneously.

24. The new window is as follows
25. Then Click “OK”

26. Now Click on the Peripherals from menu bar, and check your required port as shown in fig below
27. Drag the port a side and click in the program file.

29. You are running your program successfully
7. CONCLUSION AND FUTURE SCOPE

The fire fighter is still facing the risk of fire hazard because they still need to go near to fire scene. To reduce the fire fighter risk during fire incident, a fire fighting mobile robot is designed. Fire fighting mobile robot is a remote controlled machine that replace fire fighter to perform fire fighting task. The application of fire fighting mobile robot is able to reduce the direct contact of fire fighter to the fire hazards. The paper reviews a variety of technologies and state-of-the-art technology of fire fighting mobile robot. The paper also describes the first Malaysian designed and built fire fighting mobile robot, namely, MyBOT2000.
8. REFERENCES


