Fire Fighting Robot

Pratik. K. Palyekar, Chandan Singh, Ashish Singh, Dinesh.K.Yadav

Mechanical Engineering Department

Vishwatmak Om Gurudev College of Engineering

University of Mumbai, Maharashtra, India

Emailid-chandansingh1995@yahoo.in, dy519959@gmail.com, pratikp1711@gmail.com

Abstract: The project is designed to develop a fire fighting robot using RF technology for remote operation. The robotic vehicle is loaded with water tanker and a pump which is controlled over wireless communication to throw water. An 8051 series of microcontroller is used for the desired operation.

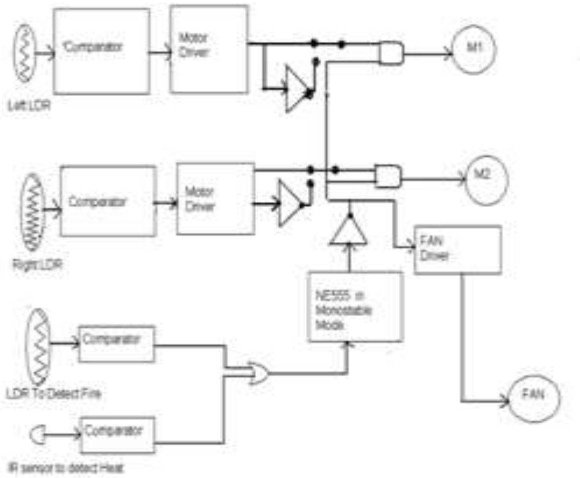
At the transmitting end using push buttons, commands are sent to the receiver to control the movement of the robot either to move forward, backward and left or right etc. At the receiving end three motors are interfaced to the microcontroller where two of them are used for the movement of the vehicle and the remaining one to position the arm of the robot. The RF transmitter acts as a RF remote control that has the advantage of adequate range (up to 200 meters) with proper antenna, while the receiver decodes before feeding it to another microcontroller to drive DC motors via motor driver IC for necessary work. A water tank along with water pump is mounted on the robot body and its operation is carried out from the microcontroller output through appropriate signal from the transmitting end. The whole operation is controlled by an 8051 series microcontroller. A motor driver IC is interfaced to the microcontroller through which the controller drives the motors.

*****

I. INTRODUCTION

The firefighting robot is intended to hunt down a flame in a little floor arrangement of a house, smother the flame (by putting a glass over the LEDs), and afterward come back to the front of the house. This mission is isolated into littler undertakings, and every errand is actualized in the most effective way. The route of the robot all through the house is accomplished by information gave by a line tracker and Ultrasound transducers. The objective procurement is accomplished by information gave by a camera. The organization of the smothering gadget is actualized with a custom arm controlled by servos. Alongside these pivotal errands were other outline imperatives, for example, the size, speed, and supply of force. Every characterizing normal for the robot is portrayed in more detail in this record.

II. BLOCK-DIAGRAM



III. Navigation

The robot is customized for a known domain that is not anticipated that would change. This permitted us to utilize a less complex programming procedure to explore the robot to each of its conceivable destinations. Rather than giving the robot a guide of the range, and coding a calculation that could decide the best way to take for any destination, the genuine known courses were customized in. While not as rich, genuine courses can be 'hard-coded' in considerably less time, furthermore requires less RAM (something fairly constrained on a Microcontroller). Be that as it may, another environment would mean recoding the greater part of the way schedules, rather than stacking another guide document. The schedules are coded to search for points of interest, for example, the nonappearance or nearness of certain dividers, lines, or separations, to decide its ebb and flow area. This is a stage above dead retribution, where the robot would indiscriminately track its separation flew out to decide its areas. Dead retribution presents issues on various surfaces and when there are slight varieties in the areas of the entryways, as could happen amid the opposition. For our coding, the main schedule that was left to dead retribution was a 90-degree turn (when not helped by a line). The level of blunder was discernible, yet insufficient to bring about issues for the route. To encourage coding the courses taking into account milestones less demanding, a guide was drawn with specific purposes of interest set apart as directions. While the directions had no intending to the Microcontroller, they gave the programming group shared belief to work from. One part could reuse a routine to travel between various coordinates, despite the fact that their last destinations were distinctive.

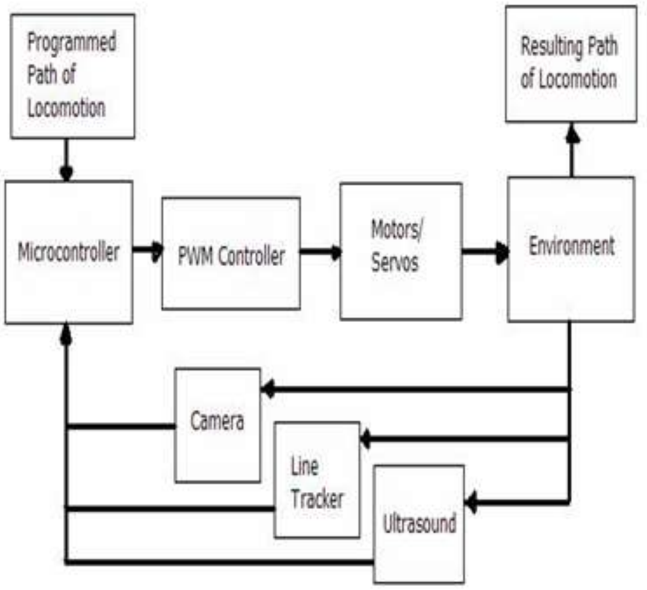
IV. WORKING OF THE ROBOT

The circuit implemented consists mainly of two different sub-circuits.

The principal part involves making the robot take after a dark strip. This was done utilizing a comparator circuit utilizing the LDR whose reference voltage was altered utilizing the potentiometer. This depended on the marvel that the resistance of the LDR diminishes as the force of light falling on it increments. For our situation, the light reflected from the white surface is more than that from the dark surface. Thusly, the voltage in positive terminal of the comparator stays high the length of the robot is proceeding onward white surface. In the event that the dark surface go under one of the two LDR's the engine relating to that LDR stops, the other making it move far from the dark line. Be that as it may, as the other LDR crosses the dark line, its engine stops, the other driving it in other bearing, bringing about its taking after the dark line. In this way, the yield voltage of the Op-Amp has been shifted by need, but the present coursing through the yield of Op-Amp is in some mA .Thus there was requirement for current intensification. The Darlington –pair was utilized for the same. The resistance of the engine was 5-10ohms,thus source supporter circuit was utilized. The second part involved utilizing LDR's and IR-recipients to identify fire .The double Op-Amp LM358 was utilized for the same. The two comparator circuits were utilized .For LDR, the working was same, however for IR collector the voltage and not the resistance shift as indicated by the power of light. Along these lines the two finishes of IR-beneficiary were associated with ground and positive terminal of the comparator .The yield of these two were ORed and given to a monostable 555 clock with time of 4 sec.555 was utilized on the grounds that when the robot sees the flame ,the fan begins and the engines stop yet when fan begins the force of light abatements and subsequently the two engines begin again and there is a chance that the engine begins before the flame is quenched .The 555 aides in producing a high beat of around 4 seconds which will stay high and won't rely on upon the power of light for the same span .Thus , this will ensure that the flame is stifled before pushing forward. Once the flame is quenched it will hold its unique movement.

V. MOTOR CONTROL SYSTEM

Feedback from ultrasound and line tracker Control system block diagram



The robot is customized for a known domain that is not anticipated that would change. This permitted us to utilize a more straightforward programming procedure to explore the robot to each of its conceivable destinations. Rather than giving the robot a guide of the region, and coding a calculation that could decide the best way to take for any destination, the real known courses were customized in. While not as rich, real courses can be 'hard-coded' in significantly less time, furthermore requires less RAM (something fairly restricted on a Microcontroller). Nonetheless, another environment would mean recoding the greater part of the way schedules, rather than stacking another guide document. The schedules are coded to search for milestones, for example, the nonattendance or nearness of certain dividers, lines, or separations, to decide its ebb and flow area. This is a stage above dead retribution, where the robot would indiscriminately track its separation flew out to decide its areas. Dead retribution presents issues on various surfaces and when there are slight varieties in the areas of the entryways, as could happen amid the opposition. For our coding, the main schedule that was left to dead retribution was a 90-degree turn (when not helped by a line). The level of blunder was perceptible, however insufficient to bring about issues for the route. To encourage coding the courses in view of milestones simpler, a guide was drawn with specific purposes of interest set apart as directions. While the directions had no intending to the Microcontroller, they gave the programming group shared view to work from. One part could reuse a routine to travel between various coordinates, despite the fact that their last destinations were distinctive.

VI. IC 8051

In 1981 Intel Corporation Introduced A 8 bit Microcontroller The Intel 8051 is a 8-bit microcontroller which implies that most accessible operations are restricted to 8 bits. There are 3 essential "sizes" of the 8051: Short, Standard, and Extended. The Short and Standard chips are regularly accessible in DIP (double in-line bundle) structure, yet the Extended 8051 models frequently have an alternate structure figure, and are not "drop-in good". Every one of these things are called 8051 in light of the fact that they would all be able to be customized utilizing 8051 low level computing construct, and they all share certain components (in spite of the fact that the diverse models all have their own extraordinary elements).

Some of the features that have made the 8051 popular are:

- 4 KB on chip program memory.
- 128 bytes on chip data memory(RAM)

[32 bank reg + 16 bit addressable reg + 80 general purpose reg]

- 4 reg banks.
- 128 user defined software flags.
- 8-bit data bus
- 16-bit address bus
- 16 bit timers (usually 2, but may have more, or less).
- 3 internal and 2 external interrupts.
- Bit as well as byte addressable RAM area of 16 bytes.
- Four 8-bit ports, (short models have two 8-bit ports).
- 16-bit program counter and data pointer.
- 1 Microsecond instruction cycle with 12 MHz Crystal.

8051 models may also have a number of special, model-specific features, such as UART, ADC, Op_Amps, etc... It is a very powerful micro controller.

Basic Pins

PIN 9: PIN 9 is the reset pin which is used to reset the microcontroller's internal registers and ports upon starting up. (Pin should be held high for 2 machine cycles.)

PINS 18 & 19: The 8051 has a built-in oscillator amplifier hence we need to only connect a crystal at these pins to provide clock pulses to the circuit.

PIN 40 and 20: Pins 40 and 20 are VCC and ground respectively. The 8051 chip needs +5V 500mA to function properly, although there are lower powered versions like the Atmel 2051 which is a scaled down version of the 8051 which runs on +3V.

PINS 29, 30 & 31: As described in the features of the 8051, this chip contains a built-in flash memory. In order to program this we need to supply a voltage of +12V at pin 31. If external memory is connected then PIN 31, also called EA/VPP, should be connected to ground to indicate the presence of external memory. PIN 30 is called ALE (address latch enable), which is used when multiple memory chips are connected to the controller and only one of them needs to be selected.We will deal with this in depth in the later chapters. PIN 29 is called PSEN. This is "program store enable". In order to use the external memory it is required to provide the low voltage (0) on both PSEN and EA pins.

Ports

There are 4 8-bit ports: P0, P1, P2 and P3.

PORT P1 (Pins 1 to 8): The port P1 is a general purpose input/output port which can be used for a variety of interfacing tasks. The other ports P0, P2 and P3 have dual roles or additional functions associated with them based upon the context of their usage.The port 1 output buffers can sink/source four TTL inputs. When 1s are written to portn1 pins are pulled high by the internal pull-ups and can be used as inputs.

PORT P3 (Pins 10 to 17): PORT P3 acts as a normal IO port, but Port P3 has additional functions such as, serial transmit and receive pins, 2 external interrupt pins, 2 external counter inputs, read and write pins for memory access.

PORT P2 (pins 21 to 28): PORT P2 can also be used as a general purpose 8 bit port when no external memory is present, but if external memory access is required then PORT P2 will act as an address bus in conjunction with PORT P0 to access external memory. PORT P2 acts as A8-A15, as can be seen from fig 1.1

PORT P0 (pins 32 to 39) PORT P0 can be used as a general purpose 8 bit port when no external memory is present, but if external memory access is required then PORT P0 acts as a multiplexed address and data bus that can be used to access external memory in conjunction with PORT P2. P0 acts as AD0-AD7, as can be seen from fig 1.1

PORT P10: asynchronous communication input or Serial synchronous communication output.

VII. ARM DESIGN

A HiTec HS-55 MicroLite servo was utilized as our height servo. The capacity of this servo is to augment and withdraw our arm. This servo not just must be sufficiently solid to withstand the additional weight that is included by torque, however the option of the quenching operators (the container). We likewise required this servo to be smaller and lightweight. Two HS-50 HiTec Feather servos were likewise utilized. These give a large portion of the torque of the HS-55 servos,

Be that as it may we didn't require an incredible arrangement torque since the lower furthest point of the arm wouldn't be focused as much. The primary HS-50 was utilized as a wrist for our arm. It was composed so that the whole arm when collapsed up hides flawlessly in the robot making it minimized yet practical. The gripper comprises of another HS-50 servo connected to two U-molded metal bars as fingers. The bars are molded and mounted on a custom edge so that when the servo pivots counter clockwise it makes a torque pushing the fingers open.

VIII. MOTOR DRIVE SYSTEM

The motor drive system consists of a PWM Controller, a motor controller, and geared drive DC motors. Using a motor controller and geared drive motors is a great improvement over modified hobby servo solutions in reliability, speed, noise (both electrical and audible), and power consumption. This solution is about twice the initial cost of using servos, since a separate motor controller is used. However, the longevity of geared DC motor over a modified hobby servo pays for the difference after about 100 hours of use. Hobby servos that are designed for intermittent use typically fail after about 30 hours of continuous use.

8.1PWM Controller

The PWM controller handles the task of outputting a constant pulse width modulation signal to the servos and motors, freeing the Microcontroller of this task and sparing I/O lines. The Microcontroller used on this robot could handle two PWM outputs in hardware, which is enough only for the main drive motors. To provide additional PWM signals for all of the required servos and drive motors, a separate PWM controller was serially interfaced to the Microcontroller. This allows the Microcontroller to send updates only as necessary to the PWM controller. The PWM controller chosen handles up to 8 servos, and receives commands at 9600 bps.

8.2Motor Controller

The robot uses a dual H-bridge motor controller, controlled by a PWM signal, allowing it to be interfaced as a standard servo. Pulse widths greater than 1.5 ms produce forward motion on the attached motors, pulse widths less than 1.5 ms produce reverse motion. A 1.5 ms pulse width stops the motors.

8.3Drive Motors

The drive motors are two geared DC motors, with a max speed of 200 RPM at 7.2 volts. Using 5.5 cm diameter wheels, this translates to a max speed of about 57 cm/s. This is too fast for operation inside the maze, so the robot is operated at less than maxspeed.

IX. GEARS

A **gear** or **cogwheel** is a rotating machine part having cut *teeth*, or cogs, which mesh with another toothed part to transmit torque. Geared devices can change the speed, torque, and direction of a power source. Gears almost always produce a change in torque, creating a mechanical, through their gear ratio, and thus may be considered a simple machine. The teeth on the two meshing gears all have the same shape.[1] Two or more meshing gears, working in a sequence, are called a gear train or a *transmission*. A gear can mesh with a linear toothed part, called a rack, thereby producing translation instead of rotation.

The gears in a transmission are analogous to the wheels in a crossed belt pulley system. An advantage of gears is that the teeth of a gear prevent slippage.

When two gears mesh, if one gear is bigger than the other, a mechanical advantage is produced, with the rotational speeds, and the torques, of the two gears differing in proportion to their diameters.

$$\frac{900mAh}{750mA} \; 1.2h$$

This provides an estimated run time of 1.2 hours. It would be safe to subtract some percentage from that time to account for motor stalls and other unforeseen requirements. The same calculation for 2000mAh NiMH provides a run time of 2.6 hours. Both NiCad and NiMH provide a reasonable run time, and we chose 1500 mAh NiMH batteries for the robot, which provides about 2 hours of run time.

Battery Life Calculation for NiMH:

$$\frac{1500mAh}{750mA} = 2.0h$$

X. LINE REGULATION

To protect the Microcontroller from power starvation and excessive noise, two regulators were used to provide power to the robot. One regulator provided power to the servos, and another precision low dropout regulator supplied power to the Microcontroller and associated electronics. The motor controller was powered directly from the battery, unregulated. To make this work we had to make sure that the stall current of the motor did not exceed the maximum current supply of the battery minus the current demands of the Microcontroller. Decoupling capacitors were added in appropriate places to ensure noise free operation. Before separating the regulated supplies, the Microcontroller could occasionally reset. With the split regulator design, the Microcontroller no longer resets. Had the current requirements

XI. CONCLUSIONS

This autonomous robot successfully performs the task of a firefighter in a simulated house fire. The robot accurately and efficiently finds the fire within the allotted time after the fire alarm is heard and returns to a safe place (Home), before the five minute time allocated for each trial is met. There are seldom incidents where therobot will veer off track however the use of preventative programming allows for the robots correction to return to its desired path.

of the motors been less predictable in stall conditions, they could have been powered through their own current limiting supply, or share the regulator of the servos, at a slight loss in speed and efficiency.Returns to a safe place (Home), before the five minute time allocated for each trial is met. There are seldom incidents where the robot will veer off track however the use of preventative programming allows for the robots correction to return to its desired path or location. Warehouses may be the first to benefit from this technology, since the expense of activating other types of fire extinguishers may outweigh that of a robot, where product stock could be damaged by imprecise fire control methods.

REFERENCES

[1] http://www.ewh.ieee.org/r3/jamaica/southeastcon/robot.html

[2] http://www.wikipedia.com/