

Design, Implementation and Control of a Humanoid Robot for Obstacle Avoidance using 8051 Microcontroller

Sachin M U¹, Pravin Gaonkar²

¹*Department of Electronics and Communication Engineering, Visvesvaraya Technological University, Belgaum, Karnataka, India.*

²*Department of Mechanical Engineering, Visvesvaraya Technological University, Belgaum, Karnataka, India.*

Abstract: *In this paper, the design, implementation and control of a humanoid robot, which enables human-like walk and a path planning of humanoid robot for obstacle avoidance by using infrared sensors (IRs) is proposed. As the focus is to obtain human-like walk, the robot is designed to resemble human proportions. Based on the obtained information from IR sensors, a software flow proposed to decide the behaviour of robot so that the robot avoids obstacles and goes to the destination. Furthermore the hardware and software necessary to obtain a fully autonomous system is developed and implemented. Human-like walk was not obtained on the real system, due to system limitations. If a new interface to the DC-motors in the servos was developed, and a faster on-board computer was chosen, human-like walk should be possible.*

Keywords- *Humanoid robot, Autonomous mobile robot, Obstacle avoidance, IR sensors, servo motors, 8051 Micro controller.*

I. Introduction

Traditional wheeled robots have been used to perform tasks where movement of the robot is necessary. However in the recent years more and more interest has been given toward humanoid or biped robots. An advantage of the biped robots, compared to the wheeled, is the ability to move around in human environments, where different obstacles or stairs should be surmounted. It is expected that within the coming years, the need for two legged robots will increase, and as the knowledge in the area is expanded, the number of different tasks that can be solved by robots will increase rapidly. The number of successfully developed biped robots is relatively small, but several companies have invested large amounts of money and research into developing such robots. The task of making a humanoid robot walk is however not trivial, since the code is a highly complex.

To the best of our knowledge, no one has obtained real human-like walk on a biped robot. Robot soccer games are teams organized by several robots to play soccer games under fixed restriction and rules. Two soccer alliances of the robot are the FIRA and the RoboCup. They hold the robot soccer match of world cup and international congress every year. The humanoid robots have to detect all information from the game field and decide its strategy by itself. There are many robots in the match field, so the robot must have the ability to avoid the collision with other robots and move to an appropriate destination. Obstacle avoidance is also a competition category in the FIRA Cup. The main idea of the competition category is testing the ability of obstacle avoidance of the robot. In this paper, an obstacle avoidance method based on the IR sensors is proposed for the autonomous humanoid robot. The infrared (IR) sensors are installed in the robot to detect the environment including obstacles. Based on the obtained information from IR sensors an obstacle avoidance method is proposed to determine the free path of robot so that the robot can avoid obstacles and go to a destination. The rest of this paper is organized as follows: In Section 2, Servo motor working is described. In Section 3, 8051 Microcontroller features is described. In Section 4, Micro vision Keil (IDE) is described. In Section 5, IR Sensor interfacing is described. In Section 6, Interfacing of Servo and IR sensor to the Microcontroller are described. Finally, in Section 7, some conclusions are made.

II. Servo Motor

Motors are an essential part of robots. Simple motor with error sensing negative feedback mechanism to adjust the shaft of motor on a particular position are servo motor. Radio Control (RC) hobby servos are small actuators designed for remotely operating model vehicles such as cars, airplanes, and boats. Nowadays, RC servos are become more popular in robotics, creating humanoid robot, robotic arm and etc. This is because its' ability to rotate and maintain certain location, position or angle according to control pulses from a single wire[1].

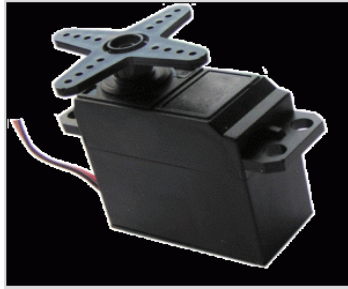


Figure 1. Standard Servo.

Currently there are two types of servo motor available on the market, the first one is called standard servo and the other one is called continuous servo. Standard servo can rotate to maximum (clockwise or counter clockwise) of 180 degrees while continuous servo can rotate up to 360 degrees in both direction. Controlling the Motor is one of interesting topics in the embedded world especially for the robotics [2].

2.1 Inside a servo motor

Inside a typical RC servo, a small DC motor and gearbox(metal or plastic) to do the work, a potentiometer to measure the position of the output gear, and an electronic circuit that controls the motor to make the output gear move to the desired position. Because all of these components are packaged into a compact, low-cost unit, RC servos are great actuators for robots. The control pulse is feed to a pulse width to voltage converter. This essentially produces a voltage related to the length of the applied pulse. The circuit is tuned to produce a useful voltage over a 1ms to 2ms period. The output voltage is buffered and so does not decay significantly between control pulses so the length of time between pulses is not critical[3].

The current rotational position of the servo motor output shaft is read by a sensor. This is normally a potentiometer (variable resistor) which produces a voltage that is related to the absolute angle of the output shaft. The position sensor then feeds its current value into the Error Amplifier which compares the current position with the commanded position from the pulse width to voltage converter. The error amplifier is an operational amplifier with negative feedback. It will always try to minimise the difference between the inverting (negative) and non-inverting (positive) inputs by driving its output in the correct direction[3][9].

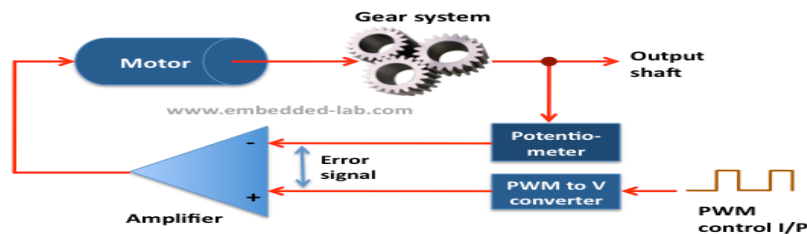


Figure 2. Functional Block Diagram of Servo.

The output of the error amplifier is either a negative or positive voltage representing the difference between its inputs. The error amplifier output is used to drive the motor, If it is positive the motor will turn in one direction, if negative the other. This allows the error amplifier to reduce the difference between its inputs (thus closing the negative feedback loop) and so make the servo go to the commanded position. The servo normally contains a single integrated circuit and a hand full of discrete components to implement the entire control system[3][7][8].

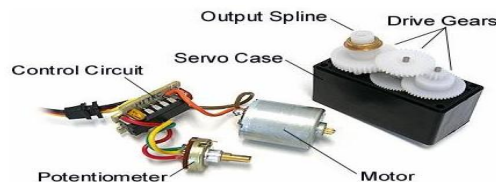


Figure 3. Structure of Actual Servo.

The servo motor use PWM signal for controlling the DC motor, unlike normal PWM usually used in ordinary DC motor, this PWM signal is not use for controlling the rotation speed, instead it is use for controlling the motor direction or position. Most servo motor will work well on 50 Hz of PWM frequency, this mean the PWM signal should have a period of 20ms. The electronic circuit inside the servo motor will response to the PWM signal width[3][4][10].

2.2 Servo control

The servo motor can be moved to a desired angular position by sending PWM (pulse width modulated) signals on the control wire. The servo understands the language of pulse position modulation. A pulse of width varying from 1 millisecond to 2 milliseconds in a repeated time frame is sent to the servo for around 50 times in a second. The width of the pulse determines the angular position. For example, a pulse of 1 millisecond moves the servo towards 0°, while a 2 milliseconds wide pulse would take it to 180°. The pulse width for in between angular positions can be interpolated accordingly. Thus a pulse of width 1.5 milliseconds will shift the servo to 90°. It must be noted that these values are only the approximations. The actual behaviour of the servos depends upon its manufacture. Pulse Widths = The Servo's Heartbeat=position[4][5].

A sequence of such pulses (50 in one second) is required to be passed to the servo to sustain a particular angular position. When the servo receives a pulse, it can retain the corresponding angular position for next 20 milliseconds. So a pulse in every 20 millisecond time frame must be fed to the servo[5].

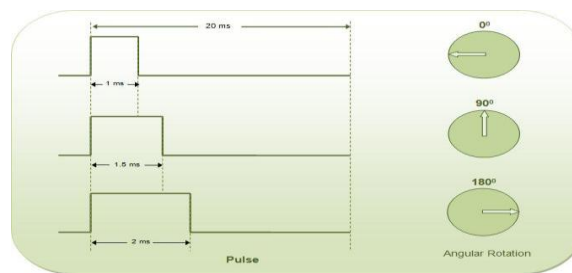


Figure 4. Controlling Angular Position of Servo Motor using Pulse Width Modulation.

2.3 Servo wiring

Hobby Servo Motors have three wires, two of them (RED and BLACK) are used to give power and the third one is used to give control signals. Servo can be easily be controlled using microcontrollers using Pulse Width Modulated (PWM) signals on the control wire. Here we are using a servo whose angular rotation is limited to 0 – 180°[1][5].

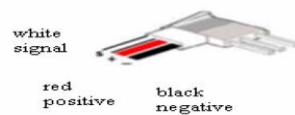


Figure 5. Connection of RC Servo Motor.

2.4 How RC servo motor works

Let's look at an example , A servo is initially situated at the 1.40ms-position when it receives a train of 1.75ms-pulses, as shown in Figure 6. The servo's control board determines that the shaft must rotate counter clockwise to reach the 1.75ms-position, and furthermore, it determines that the initial rotation speed should be high because the 1.40ms-position is far away from the 1.75ms-position. Since the first pulse does not move the shaft to the desired position, the process is repeated and the shaft continues to rotate counter clockwise with decreasing speed. After a few pulses, the shaft reaches the desired 1.75ms-position [6].

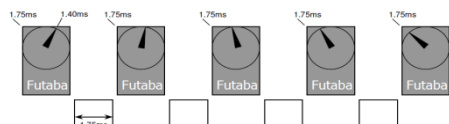


Figure 6. Servo is originally at the 1.40ms-position when it receives a train of 1.75ms-pulses. The servo's control board interprets the pulses and causes the shaft to rotate toward the 1.75ms-position.

After a few pulses, the shaft is in the desired 1.75ms-position.

If additional 1.75ms-pulses are sent to the servo, they are ignored since the shaft is already in the 1.75msposition. This is shown in Figure 7.

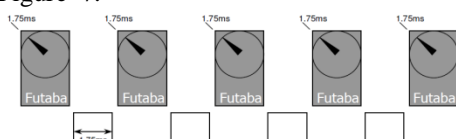


Figure 7. Additional 1.75ms-pulses are sent to the servo but the control board ignores them since the servo is already at the 1.75ms-position.

Figure 8, below shows another train of 1.75ms-pulses being sent to the servo, but this time the shaft will rotate clockwise since the desired 1.75ms-position is to the right of the original 2.0ms-position. In addition, the shaft will spin at a relatively slow speed since the 1.75ms-position is near the 2.0ms-position[6].

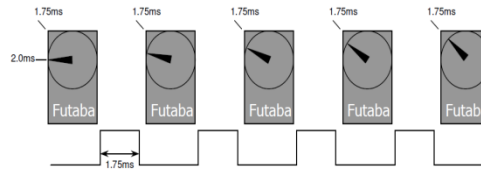


Figure 8. The servo rotates clockwise to reach the 1.75ms-mark.

Following are general rules when programming a servo:

- A train of pulses whose pulse widths are approximately 1.0ms will always rotate the shaft fully to its left-most position.
- A train of pulses whose pulse widths are approximately 1.5ms will rotate the shaft to its centre position.
- A train of pulses whose pulse widths are approximately 2.0ms will always rotate the shaft fully to its right-most position[6].
-

III. 8051 Microcontroller

A microcontroller is an economical computer-on-a-chip built for dealing with specific tasks. The most commonly used set of microcontrollers belong to 8051 Family(AT89S52,AT89C52). Through 8051, the world became witness to the most revolutionary set of microcontrollers. Atmel fabricated the flash ROM version of 8051 which is popularly known as AT89S52. The flash memory can erase the contents within seconds which is best for fast growth. To build up a microcontroller based system using AT89S52, it is essential to have ROM burner that supports flash memory. Note that in Flash memory, entire contents must be erased to program it again. The contents are erased by the ROM burner. Atmel is working on a newer version of microcontroller that can be programmed using the serial COM port of IBM PC in order to get rid of the ROM burner[5][11].

3.1 Features of 8051

The main features of 8051 microcontroller are,

- RAM – 128 Bytes (Data memory).
- ROM – 4Kbytes (ROM signify the on – chip program space).
- Serial Port – Using UART makes it simpler to interface for serial communication.
- Two 16 bit Timer/ Counter.
- Input/output Pins – 4 Ports of 8 bits each on a single chip.
- 6 Interrupt Sources.
- 8 – bit ALU (Arithmetic Logic Unit).
- Harvard Memory Architecture – It has 16 bit Address bus (each of RAM and ROM) and 8 bit Data Bus.
- 8051 can execute 1 million one-cycle instructions per second with a clock frequency of 12MHz.
- This microcontroller is also called as “System on a chip” because it has all the features on a single chip[5][11].

3.2 Memory Architecture

The 4 discrete types of memory in 8051 are,

- Internal RAM – This memory is located from address 0 to 0xff. The memory locations from 0x00 to 0x7F are accessed directly. The bytes from 0x20 to 0x2F are bit-addressable. Loading R0 and R1 the memory location from 0x80 to 0xFF can easily accessed.
- Special Function Registers (SFR) – Located from address 0x80 to 0xFF of the memory location. The same instructions used for lower half of Internal RAM can be used to access SFR’s. The SFR’s are bit addressable too.
- Program Memory – This is read only memory which is located at address 0. With the help of 16 bit Special Function Register DPTR, this memory can also save the tables of constants.
- External Data Memory – Located at address 0. The Instruction MOVX (Move External) should be used to access the external data memory[5].

3.3 AT89S52 Microcontroller

AT89S52 is an 8-bit microcontroller and belongs to Atmel's 8051 family. It has 8K Bytes of In-System Programmable (ISP) Flash Memory and 256 x 8-bit Internal RAM. It can be erased and program to a maximum of 1000 times. In 40 pin AT89S52, there are four ports designated as P₁, P₂, P₃ and P₀. All these ports

are 8-bit bi-directional ports, *i.e.*, they can be used as both input and output ports. Except P₀ which needs external pull-ups, rest of the ports have internal pull-ups. When 1s are written to these port pins, they are pulled high by the internal pull-ups and can be used as inputs. These ports are also bit addressable and so their bits can also be accessed individually[5][12].

3.4 Programming Timers

The 8051 has two timers/counters, they can be used either as Timers to generate a time delay or as Event counters to count events happening outside the microcontroller. Both Timer 0 and Timer 1 are 16 bits Wide, is accessed as two separate registers of low byte and high byte. The low byte register is called TLO/TL1 and The high byte register is called TH0/TH1. Both timers 0 and 1 use the same register, called TMOD (timer mode), to set the various timer operation modes. TMOD is a 8-bit register. The lower 4 bits are for Timer 0 and the upper 4 bits are for Timer 1, mode configuration. In each case, The lower 2 bits to set the timer mode and The upper 2 bits to specify the operation. TCON is a 8 bit timer control register[11].

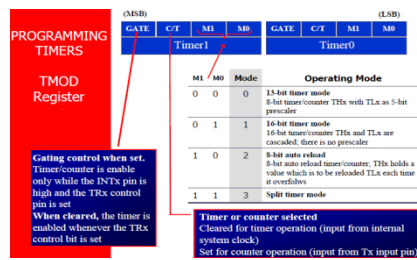


Figure 9. TMOD Register .

3.5 Mode 1 Operation of Timers

It is a 16-bit timer, therefore, it allows value of 0000 to FFFFH to be loaded into the timer’s register TL and TH. After TH and TL are loaded with a 16-bit initial value, the timer must be started, this is done by SETB TR0 for timer 0 and SETB TR1 for timer 1. After the timer is started, it starts to count up until it reaches its limit of FFFFH. When it rolls over from FFFFH to 0000, it sets high a flag bit called TF (timer flag) Each timer has its own timer flag, TF0 for timer 0, and TF1 for timer 1. This timer flag can be monitored When this timer flag is raised, one option would be to stop the timer with the instructions CLR TR0 or CLR TR1, for timer 0 and timer 1, respectively. After the timer reaches its limit and rolls over, in order to repeat the process, TH and TL must be reloaded with the original value, and TF must be reloaded to 0.[11].

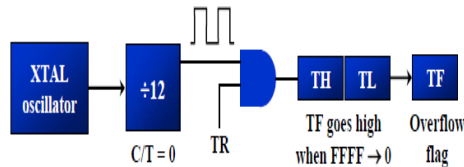


Figure 10. Functional block diagram of mode 1 operation of Timers.

Algorithm to generate time delay in mode 1 operation of timers are,

1. Load the TMOD value register indicating which timer (timer 0 or timer 1) is to be used and which timer mode (mode1) is selected.
2. Load registers TL and TH with initial count value.
3. Start the timer.
4. Keep monitoring the timer flag (TF) with the JNB TFx, target instruction to see if it is raised, Get out of the loop when TF becomes high.
5. Stop the timer.
6. Clear the TF flag for the next round.
7. Go back to Step 2 to load TH and TL again[11].

To calculate the values to be loaded into the TL and TH registers, look at the following example, Assume XTAL = 11.0592 MHz, we can use the following steps for finding the TH, TL registers’ values.

- i. Divide the desired time delay by 1.085 us.
- ii. Perform 65536 – n, where n is the decimal value we got in Step i.
- iii. Convert the result of Step ii to hex, where yyxx is the initial hex value to be loaded into the timer’s register.
- iv. Set TL = xx and TH = yy[11].

3.6 Mode 2 Operation of Timers

It is an 8-bit timer, therefore, it allows only values of 00 to FFH to be loaded into the timer's register TH. After TH is loaded with the 8-bit value, the 8051 gives a copy of it to TL. Then the timer must be started. This is done by the instruction SETB TR0 for timer 0 and SETB TR1 for timer 1. After the timer is started, it starts to count up by incrementing the TL register, It counts up until it reaches its limit of FFH. When it rolls over from FFH to 00, it sets high the TF (timer flag). When the TL register rolls from FFH to 0 and TF is set to 1, TL is reloaded automatically with the original value kept by the TH register. To repeat the process, we must simply clear TF and let it go without any need by the programmer to reload the original value. This makes mode 2 an auto-reload, in contrast with mode 1 in which the programmer has to reload TH and TL[11].

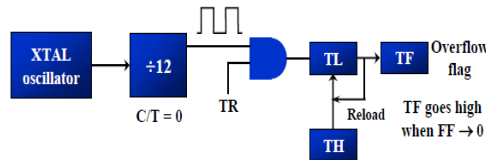


Figure 11. Functional block diagram of mode 2 operation of Timers.

Algorithm to generate time delay in mode 2 operation of timers are,

1. Load the TMOD value register indicating which timer (timer 0 or timer 1) is to be used, and the timer mode (mode 2) is selected.
2. Load the TH registers with the initial count value.
3. Start timer.
4. Keep monitoring the timer flag (TF) with the JNB TFx, target instruction to see whether it is raised, Get out of the loop when TF goes high.
5. Clear the TF flag.
6. Go back to Step4, since mode 2 is auto reload[11].

To calculate the values to be loaded into the TL and TH registers, look at the following example, Assume XTAL = 11.0592 MHz, we can use the following steps for finding the TH register values.

- i. Divide the desired time delay by 1.085 us.
- ii. Perform 256 – n, where n is the decimal value we got in Step i.
- iii. Convert the result of Step ii to hex, where xx is the initial hex value to be loaded into the timer's register.
- iv. Set TH = xx and TL is reloaded automatically[11].

IV. Software - Micro vision Keil (IDE)

Keil is a German based Software development company. It provides several development tools like

- IDE (Integrated Development environment).
- Project Manager.
- Simulator.
- Debugger.
- C Cross Compiler , Cross Assembler, Locator/Linker.

Keil Software provides you with software development tools for the 8051 family of microcontrollers. With these tools, you can generate embedded applications for the multitude of 8051 derivatives. Keil provides following tools for 8051 development[5][13].

1. C51 Optimizing C Cross Compiler,
2. A51 Macro Assembler,
3. 8051 Utilities (linker, object file converter, library manager),
4. Source-Level Debugger/Simulator,
5. µVision for Windows Integrated Development Environment.

The keil 8051 tool kit includes three main tools, assembler, compiler and linker. An assembler is used to assemble your 8051 assembly program, A compiler is used to compile your C source code into an object file. A linker is used to create an absolute object module suitable for your in-circuit emulator. The software proload is used to dump the code into the 8051 micro controller.8051 programmer is used to burn code[5][13].

8051 project development cycle: - these are the steps to develop 8051 project using keil

1. Create source files in C or assembly.
2. Compile or assemble source files.
3. Correct errors in source files.

4. Link object files from compiler and assembler.
5. Test linked application.

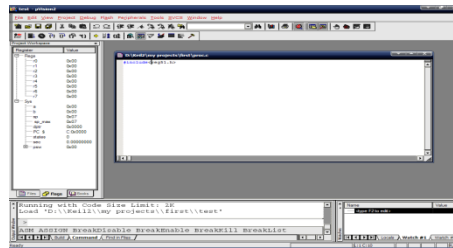


Figure 12. KEIL software.

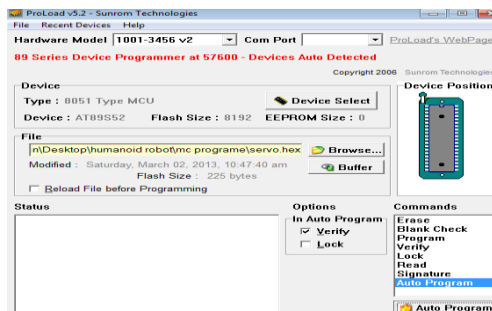


Figure 13. Proload software.

4.1 Programming environment and programmer

Formerly, programmers used machine language for coding. A machine language is a program that consists of 0s and 1s which was very dreary for the humans to program any computer. In due course of time, assembly language was developed in order to speed up the programming and make it error-free. Assembly language is a low level language which uses an assembler to translate the program into machine code. The high level programming languages such as BASIC, Pascal, Forth, C, C++, and Java are available to code the program for 8051. These high level languages make use of a Compiler to translate into machine code. For example, when a program is written in C, the program needs to be translated into machine language using C compiler. Usually, Assembly and C language is widely used for 8051 programs as compared to the other high level languages. Here we are using C and Assembly language to generate codes. The 8051 provides a total of four ports for I/O operations. 8051 has 40 pins, of which 32 pins are set aside for the four ports. PO, P1, P2, and P3 each have 8 pins and can be used for either input or output. 8051 allows you to manipulate one or all of the bits of a port, thus providing programmers with a unique and powerful feature. 8051 provides the programmer with the ability to read, write and modify each port to customize applications as much as possible[5].

V. IR Sensor for Object Detections

The interaction of the robot with the environment set ups needs mechanisms known as Sensors , that can perform the following functions :motion control variables ,detection, robot guidance without obstruction, object identification tasks, handling the objects[18].

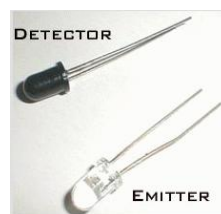


Figure 14. IR Emitter and Photodiode.

An infrared emitter is an LED made from gallium arsenide, which emits near-infrared energy at about 880nm. The infrared phototransistor acts as a transistor with the base voltage determined by the amount of light hitting the transistor. Hence it acts as a variable current source. Greater amount of IR light cause greater

currents to flow through the collector-emitter leads. The variable current travelling through the resistor causes a voltage drop in the pull-up resistor[15].

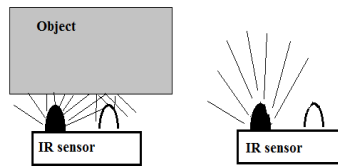


Figure 15. Sensor Behaviour if there is an object and there is no object.

In the Figure 15, the Black led is generating IR light but that IR light getting reflected by the gray or white surface. The black surface absorbs all the light and reflects nothing . So the photo diode in this case ,senses no IR light .The case of black surface is similar to that of there being no object, because in both the cases there is no reflection sensed by the photodiode. Any color other then black will reflect IR light from it. So, ultimately , if there is a reflection it means that there is an object and if there is no reflection, it means there is no object[16].

5.1 IR Sensor Circuit

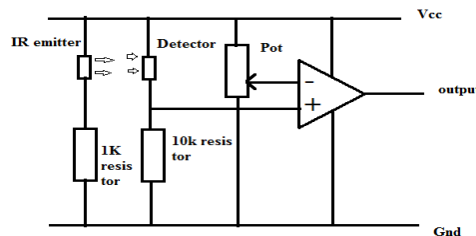


Figure 16. IR Sensor circuit.

When the supply is given to the IR sensor, the led starts emitting light radiations. If the surface is of color other than black then it reflect all the radiations. As these radiations start falling on the photo diode(detector), which is connected in reverse bias, the resistance of the Photo diode starts decreasing rapidly and the voltage drop across the diode also decreases. The voltage at pin + starts increasing and as it reaches just beyond the voltage of pin -, the comparator gives a high output . In case of the black surface , the led emits light but it is not reflected by the surface, so the photo diode detects nothing and its resistance remain infinite. Hence the comparator gives low output[16].

5.2 IC LM324

LM324 is a 14pin IC consisting of four independent operational amplifiers (op-amps) compensated in a single package. Op-amps are high gain electronic voltage amplifier with differential input and, usually, a single-ended output. The output voltage is many times higher than the voltage difference between input terminals of an op-amp. These op-amps are operated by a single power supply, LM324 and need for a dual supply is eliminated. The conventional op-amp applications can be more easily implemented with LM324 [5][14].

VI. Interfacing of Servo and IR sensor to the Microcontroller

The interfacing of servo to the microcontroller is shown in below, Figure 17. Here we are using 16 servo to model the robot as shown in Figure 18, therefore, 2 port of micro controller is used to interface servo motors because each port is of 8 bit . IR sensor or sensor array is interfaced to remaining port as input configuration. Robot is designed using CATIA Versio V5R20.

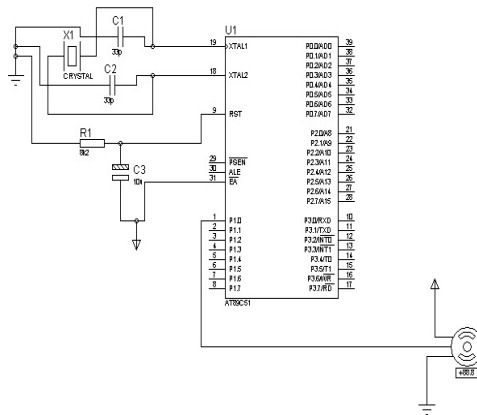


Figure 17. Interfacing of servo to Micro controller[10] .

6.1 Controlling of Autonomous Walking Humanoid robot for Obstacle Avoidance

Control strategy of Humanoid robot is described by the software flow diagram as shown in Figure 20 , for a specific path as shown in Figure 21. Table 1 , indicates the position of the respected servos shaft as numbered in Figure 18. Each servo motor is held by 3 types of C arm as shown in Figure 19. Initially all servos are in neutral position i.e 90 degree(90°). When position as mentioned in Table 1, is maintained for servo gives movement to robot. Col 1 shows servo serial numbers. Col 2 and col 3 shows initialization of the movement. Col 4 and col 5 shows right leg movement. Col 6 and col 7 shows left leg movement. Col 8 and col 9 shows termination of the movement. Col 10 to col 13 shows left side turning of the robot. Col 14 to col 17 shows right side turning of the robot[17][18].According to software flow robot will detect obstacles .

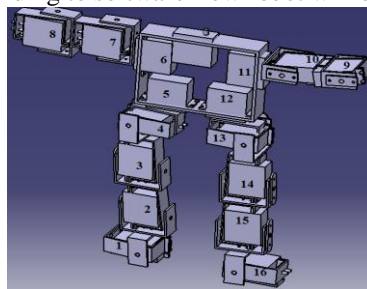


Figure 18. 16 servo motors Humanoid Robot.

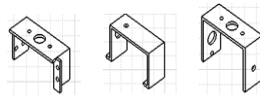


Figure 19. 3 Types of C arm.

Table 1.Position of servo for walking and object detections.

Col 1	Col 2	Col 3	Col 4	Col 5	Col 6	Col 7	Col 8	Col 9	Col 10	Col 11	Col 12	Col 13	Col 14	Col 15	Col 16	Col 17
1	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°
2	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°
3	90°	90°	135°	135°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°
4	110°	90°	90°	90°	110°	90°	90°	90°	110°	90°	90°	90°	90°	90°	110°	90°
5	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	180°	180°	90°	90°
6	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°
7	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°
8	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°
9	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°
10	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°
11	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90
12	90°	90°	90°	90°	90°	90°	90°	90°	180°	180°	90°	90°	90°	90°	90°	90°

13	90°	90°	110°	90°	90°	90°	110°	90°	90°	90°	110°	90°	110°	90°	90°	90°
14	135°	135°	90°	90°	135°	135°	90°	90°	90°	90°	110°	90°	110°	90°	90°	90°
15	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°
16	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°	90°

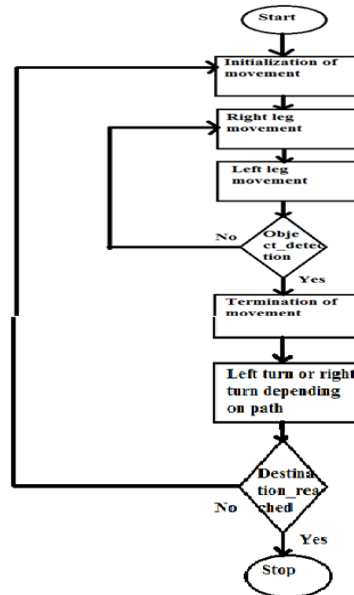


Figure 20. A software flow diagram of the movement of humanoid for obstacles detection .

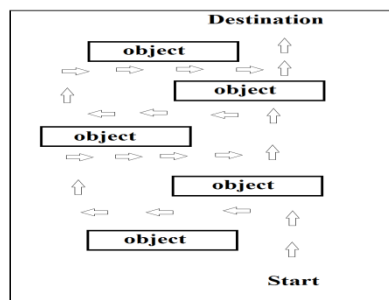


Figure 21. Specific path for humanoid robot.

VII. Conclusion and Future Work

Infrared sensors are installed in a humanoid robot to detect obstacles. Based on the obtained information from these sensors, a software flow is proposed. From the software flow , the robot can avoid obstacles and go to the destination. The advantage of these two-legged robots is, that they are often capable of performing more versatile and demanding tasks, than the traditional robots. Biped robots could be used to assist humans in carrying heavy materials around, entering high risk areas such as an atomic power plant, aid in the household, etc. An advantage of the biped robots, compared to the wheeled, is the ability to move around in human environments, where different obstacles or stairs should be surmounted. The number of different tasks that can be solved by robots will increase rapidly. This is also that makes the area very interesting. To the best of our knowledge, no one has obtained real human-like walk on a biped robot. Robot soccer games are teams organized by several robots to play soccer games under fixed restriction and rules. The humanoid robots have to detect all information from the game field and decide its strategy by itself. There are many robots in the match field, so the robot must have the ability to avoid the collision with other robots and move to an appropriate destination. The main idea of the competition category is testing the ability of obstacle avoidance of the robot. Hardware and software , walking generation and artificial intelligence are the main research field of humanoid robot.

Acknowledgement

The authors would like to thank the Principal / Director, HOD, Staff of S.D.M College of Engineering and Technology, Dharwad, Karnataka, India, for encouraging us for this research work. Our special thanks to

Prof. M. Vijay Kumar, Faculty of Department of E & C E ,S. D. M. C. E. T., Dharwad, Karnataka, India, for their timely suggestion and support.

References

- [1] RC Servo user manual-7,Cytron technology.
- [2] www.ermicro.com.
- [3] www.digitalnemesiis.com.
- [4] www.electrosome.com.
- [5] www.engineersgarage.com.
- [6] Introduction to Servomotor Programming manual pdf.
- [7] www.embedded-lab.com.
- [8] www.mcu-programming.blogspot.in.
- [9] www.umangjain25.blogspot.in.
- [10] www.vshamu.wordpress.com.
- [11] Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. McKinlay, the 8051 Microcontroller and Embedded Systems Using assembly and C.
- [12] Atmel 8051 Microcontrollers Hardware Manual(AT89S52).
- [13] Keil and proload quick start software flow manual.
- [14] LM324 datasheet.
- [15] www.irbasic.blogspot.in.
- [16] Toshendra k. Sharma, Robotics with AVR, A cook book for engineering project in Robotics with AVR microcontrollers.
- [17] Jens Christensen ,Jesper L. Nielsen, Mads S. Svendsen, Peter F. Ørts, Aalborg University Department of Electronic System Section of Automation and control, Development , Modelling and control of a humanoid robot.
- [18] Ching-Chang Wong, Chi-Tai Cheng, Kai-Hsiang Huang, Yu-Ting Yang, Department of Electrical Engineering, Tamkang University Tamsui, Taipei Hsien, 25137, Taiwan. Design and Implementation of Humanoid Robot for Obstacle Avoidance.