INTELLIGENT CURTAIN

Aditya Sundararajan, Sagar Bhanushali, Dishant Gala, Rahul Ghuge

October 22, 2012

1 Abstract

This Project is used to control the curtains automatically depending on temperature and light conditions of the room. The system will get the ambient temperature information from LM35 and the light conditions from Light Dependent Resistor(LDR). The temperature and light conditions of the room are compared with the value stored by the stored by the user. Depending on different outputs of the comparator, the curtain is either opened or closed. The System is fully controlled by the microcontroller P89V51RD2. It is a popular 8 bit microcontroller. The circuit consists of one external switch, which will do the opposite of present condition, in case the present state of curtain is not satisfactory for the user.

2 Introduction

Intelligent Curtain is a project in which the curtain opens or closes automatically, without the involvement of any human, thus bringing convenience. This automation is done with the help of Temperature sensor and LDR. The temperature sensor, LM35, gives an analog output of the temperature and this is fed to the non-inverting terminal of comparator. The inverting terminal of the comparator is connected to a 5V, which is connected in series with a preset to vary the voltage supplied to it. The LDR in series with the resistor is connected to the non-inverting terminal of the second comparator. The input to this terminal is the voltage across the resistor. The inverting terminal of the second comparator follows the same procedure as the inverting terminal of first comparator. If the particular state of curtain(open or close) is not satisfactory to the user, there is an external switch, which will do the opposite of the present state condition of curtain, ie, if the curtain is open, the switch will make it close and vice versa. Depending on various conditions of Temperature sensor and LDR, the microcontroller will drive the dc motor, through motor driver IC L293D.

3 About Microcontroller

1. Microcontroller:
   A microcontroller is a small computer on a single integrated circuit consisting of a relatively simple CPU combined with support functions such, as a crystal oscillator, timers, watchdog timer, serial and analog I/O etc. Program memory in the form of NOR flash or OTP ROM is also often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for small or dedicated applications.

2. Uses of Microcontrollers:
   Microcontrollers are used in automatically controlled products and devices. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems.

3. AT89C51:
   We are using AT89C51 microcontroller. A particularly useful feature of the AT89C51 core is the inclusion of a boolean processing engine

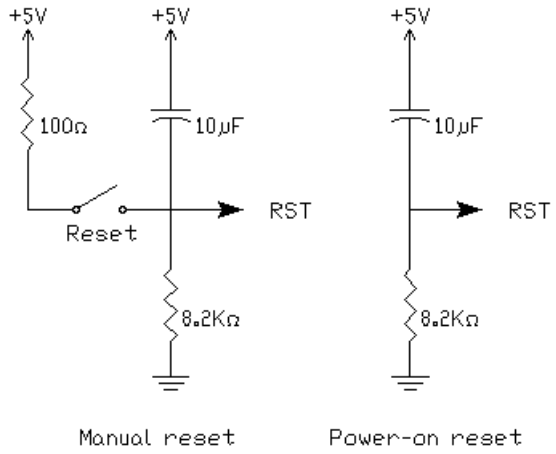
which allows bit-level boolean logic operations to be carried out directly and efficiently on internal registers and RAM. Another valued feature is that it has four separate register sets, which can be used to greatly reduce interrupt latency compared to the more common method of storing interrupt context on a stack.

AT89C51 based microcontrollers typically include UARTs, timers, 128 bytes of internal data RAM (16 bytes of which are bit-addressable),32 bits of I/O, 64 KB of internal program memory, and the AT89C51 core ran at 12 clock cycles per machine cycle, with most instructions executing in one or two machine cycles. With a 12 MHz clock frequency, the AT89C51 could thus execute 1 million one-cycle instructions per second or 500000 two-cycle instructions per second.

4. Features of AT89C51:
   (a) It provides many functions in a single package
   (b) 8-bit ALU, Accumulator and Registers
   (c) 8-bit data bus - It can access 8 bits of data in one operation
   (d) 16-bit address bus - It can access 216 memory locations - 64 kiB (65536 locations) each of RAM and ROM
   (e) On-chip RAM - 128 bytes (data memory)
   (f) On-chip ROM - 4 KB (program memory)
   (g) 32 bi-directional input/output port
   (h) UART (serial port)
   (i) Two l-bit Counter/timers
   (j) Two-level interrupt priority
   (k) Low Power Idle and Power saving mode

4 Minimum microcontroller based system

1. Reset Circuit:
   Reset is an active High input. When RESET is set to High, AT89C51 goes back to the power on state.The AT89C51 is reset by holding the RST high for at least two machine cycles and then returning it low.



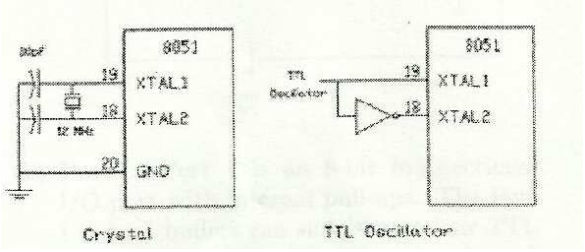
Power-On Reset:
Initially charging of capacitor makes RST High i.e. Due to transient conditions, the capacitor will act as a short circuit for a time period equal to the charging time of the capacitor. This charging time should be greater than the time taken by two machine cycles.
When capacitor charges fully, it blocks DC. As a result, Logic LOW is present at the resistor.

Manual reset:
When the switch is closed momentarily, RST will go High. This is because as the capacitor discharges, the transient conditions make the capacitor equivalent to a short circuit. As a result, the entire 5V or Logic HIGH appears across the resistor. This HIGH state, however, should be present at the RST input for atleast two clock cycles of the microcontroller. After a reset, the program counter is loaded with OOOOH. However, the content of on-chip RAM is not affected.
The following table determines the default values various SFRs will possess upon entering

the reset state.

Register	Content
Program Counter	0000 H
Accumulator	00 H
PSW	00 H
B Register	00 H
Stack Pointer	07 H
DPTR	0000 H
All Ports	FF H
IP	XXX00000 B
IX	0XX00000 B
Timer Registers	00 H
SCON	00 H
SBUF	00 H

2. Oscillator Circuit:
The ATS9C51 uses the crystal for precisely that: to synchronize its operation. Effectively, the ATS9C51 operates using what are called "machine cycles." A single machine cycle is the minimum amount of time in which a, single AT89C51 instruction can be executed. Although many instructions take multiple cycles. A89C51 has an on-chip oscillator. It needs an external crystal thats decides the operating frequency of the ATS9C51.This can be achieved in two ways:



The crystal is connected to pins 18 and 19 with stabilizing capacitors. 12 MHz (11.059MHz) crystal is often used and the capacitance ranges from 20pF to 40pF.The oscillator can also be a TTL clock source connected with a NOT gate as shown.

(a) Working of AT89C51:
A cycle is, in reality, 12 pulses of the crystal. That is to say, if an instruction takes one machine cycle to execute, it will take 12 pulses of the crystal to execute. Since we know the crystal is pulsing 11,059,000 times per second and that one machine cycle is 12 pulses, we can calculate how many instruction cycles the ATS9C51 can execute per second: 11,059,000/ 12 = 921,583.

(b) Selection of Crystal:
11.0592 MHz crystals are often used because it can be divided to give you exact clock rates for most of the common baud rates for the UART, which is compatible with IBM PCs. Despite the "oddball" value, these crystals are readily available and commonly used.

3. Power Supply Circuit:
In the rectifier circuit, we have used bridge rectifier using diodes along with IC LM7805. The IC 7805 requires the input voltage between 9V to 18V, hence if we are using mains 230V AC as input source then it must be stepped down using step down transformer.
When an ac supply is give as source input, then it gets rectified in the following manner:

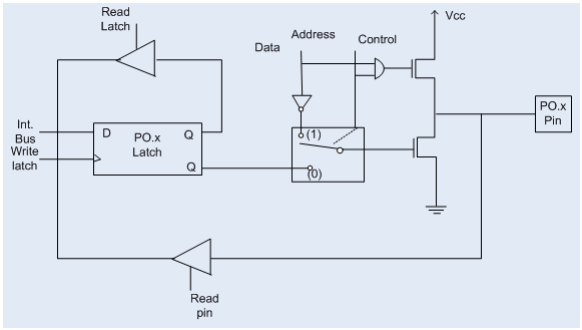
(a) During the positive half cycle diodes D1 and D3 are forward biased while diodes D2 and D4 are reverse biased. So we get positive half cycle at the output of bridge rectifier.

(b) During negative half cycle diodes D1 and D3 are reverse biased and hence they do not conduct. While diodes D2 and D4 are forward biased giving us again the positive half cycle output at the output of the bridge rectifier.

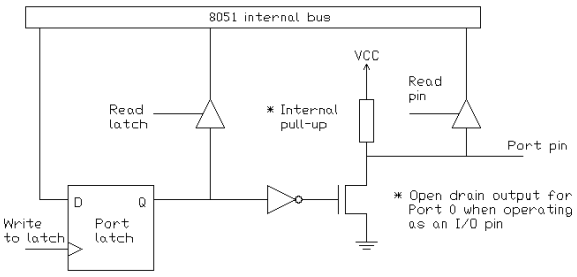
If we use directly DC voltage source then the bridge rectifier is bypassed directly and the input is given to the IC LM7805.
The IC LM7805 is a positive voltage regulator IC with internal thermal shutdown. This IC provides constant dc supply of 5V at its output pin. Hence we get a constant DC supply voltage as Vcc to, Microcontroller IC.

4. Ports of AT89C51:

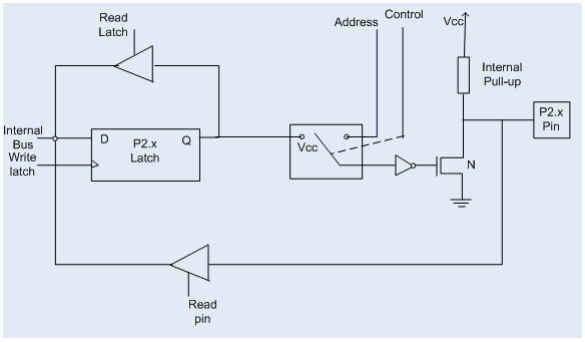
(a) PortO: Port 0 is an 8-bit open-drain bi-directional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port a pins, the pins can be used as high impedance inputs. Port 0 may also be configured to be the multiplexed low order address/data bus during accesses to external program and data memory. In this mode PO has internal pull-ups because on port 0 I/O lines data and address bus are multiplexed. Port 0 receives the code bytes during Flash programming, and outputs the code bytes during program verification and hence External pull-ups are required during program verification.

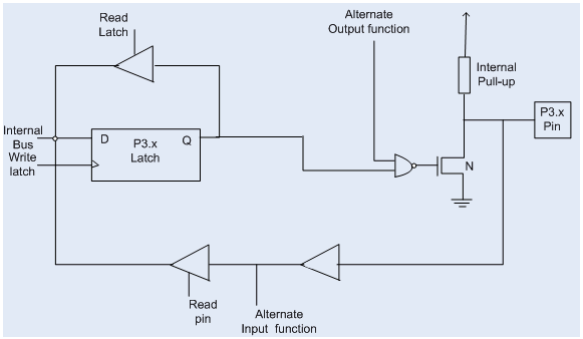


(b) Port1: Port 1 is an 8-bit bi-directional I/O port-with internal pull-ups. The Port 1 Output buffers can Sink/source four TTL inputs. When 1s are written to Port 1 pins they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL) because of the internal pull-ups. Port 1 also receives the low-order address bytes during Flash programming and verification.



(c) Port2: Port 2 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pull-ups and can be used as inputs. As inputs Port 2 pins that are externally being pulled low will source current (IIL) because of the internal pull-ups. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that uses 16-bit addresses (MOVX @ DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that uses 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register. Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.





(d)Port3: Port 3 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 3 output buffers call sink/source four TTL inputs. When 1s are written to Port 3 pins they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL) because of the pull-ups. Port 3 also serves the functions of various special features of the AT89C51 as listed below:

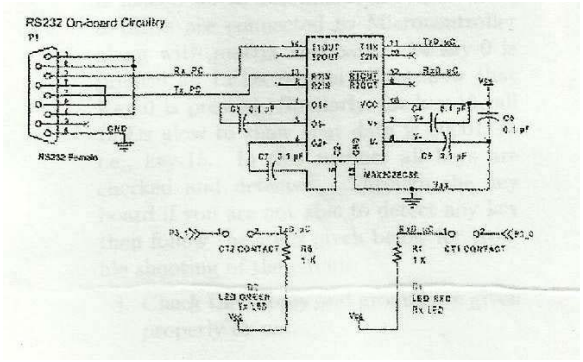
Port Pin	Function
P3.0	RxD
P3.1	TxD
P3.2	INT0
P3.3	INT1
P3.4	T0
P3.5	T1
P3.6	WR
P3.7	RD

This port does not have external pull-ups as the data present at this port as mentioned above are important for timer interrupt and serial communication and so we do not want to change it easily. Port 3 also receives some control signals for Flash programming and verification.

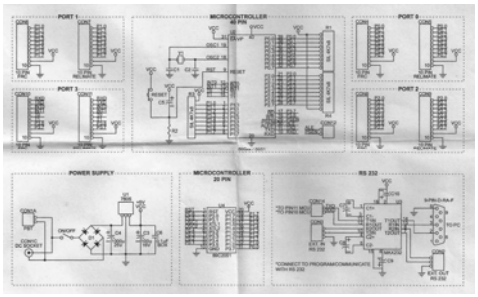
5. Pull-up resistances:

The output current from the microcontroller is 18 mA. If load connected is like dc motor or stepper motor which require a large amount of current then this excess current can be drawn using power supply and external pull ups. In the absence of this external pull ups the excess amount of current which is required will be drawn from microcontroller ic and this will damage the microcontroller.

6. Circuit diagram of serial block:



7. Circuit diagram of minimum system:



5 Troubleshooting in minimum circuit

1. Track Testing:
   The first step in checking if any system is working or not is to check the continuity of the tracks etched and the soldered parts. The Vee and GND are checked for any shorting. If any such discrepancy exists, then it should be removed. The steps given below imply how to troubleshoot the rest of the circuit.

2. Power supply checking:
   First step is to check the supply voltage by DMM with the input voltage range from 9V to l8V. Check bridge rectifier by checking 165 internal diodes of bridge. Then check input of 7805 regulator which is coming from DC supply, and check whether we are getting 5V at the output of it. Check connectivity of tracks.

3. Power on reset checking:
   Reset pin should go low for atleast two cycles so that it gets initialized properly i.e.

   All the SFRs gets initialize to 00.
   Program counter initialize to OOOOh.
   Stack pointer initialize to 07h.
   All the interrupt gets disable.
   To ensure that the peripherals connected to the microcontroller contain no bugs, the output of the ALE pin is checked. A square wave will be obtained at the ALE pin.

4. Component Testing Of Minimum Kit:
   The following steps are taken to ensure that the minimum system kit is working properly and is suited for the desired application:

   (a) To Ensure that all ports are properly connected:
   A program is loaded onto the microcontroller to give a square wave output of any suitable frequency at all the ports. The working of the ports is then checked by connecting an LED one by one to all the ports. Blinking of the LED throughout the operation governs the correct working of all the ports. If any error is reported, then the following steps are to be taken.
   i.Check the supply and ground are given properly or not.
   ii. Check connectivity of tracks.
   iii. Check the anode and cathode connection. Ideally anode is connected with Vee and cathode with the ground.
   iv. Check the IC AT89C5I on IC tester.
   v. Check the led and pull up resistor.
   vi. Check the software.

   (b)Eight LEDS and 4X4 Matrix Key- board:
   Eight LEDs and Matrix Keyboard program is loaded on to the microcontroller. Here 8 LEDs are connected to Microcontroller along with matrix keyboard. As key-O is pressed all LEDs remain off to show that Key-O is pressed. Similarly, for key-15, all LEDs glow to show that data is 00001111 i.e., key-15. In this manner all keys are checked and detected. Through the keyboard if you are not able to detect any key then follow the steps give below for trouble shooting of the circuit:
   i. Check the supply and ground are given properly or not.
   ii.Check the connections of rows and columns.
   iii. Check the switches.
   iv. Check the IC AT89C51.
   v. Check the other devices which are connected with the keyboard.
   vi. Check connectivity of tracks.

vii. Check the software.

(c) LCD:

The LCD program is loaded onto the microcontroller and the LCD is interfaced with the controller. Upon power on, the LCD will start to glow which determines that it is receiving power from the controller. Correct presentation of the text on the LCD determines that its properly working. If you won't get desired display on LCD then follow the steps given below for troubleshooting of the circuit.

i.Check the supply and ground are given properly or not.
ii. Check the LCD and its pin connection.
iii. Check the other devices which are connected with it.
iv. Check the IC AT89C51.
v. Check the software.

(4) Serial communication:

If you are not able to transmit or receive the data through transmitter and receiver then follow the steps given below for the troubleshooting of this circuit.

i. Check the supply and ground are given properly or not.
ii. Check if connections are made between the RX and TX pins and microcontroller.
iii. Check the transmitter and receiver connections.
iv. Check the other devices which are connected with it.

6 LCD Interfacing

Liquid crystal display (LCD) has material which combines the properties of both liquid and crystals. They have a temperature range within which the molecules are almost as mobile as they would be in a liquid, but are grouped together in an order form similar to a crystal.

This display contains two internal byte-wide registers, one for command and second for characters to be displayed. There are three control signals called R/W, DI/RS and En. The table given below will tell you what the use of these three signals is.

Control Signals	It's function
R/W	= 0 Writes character in display
	= 1 Reads from display
RS/DI	= 0 Selects command register
	= 1 Selects Data register to display character
En	= 0 Disables the display
	= 1 Enables the displa

By making RS/DI signal 0 you can send different commands to display. These commands are used to initialize the LCD, to select display pattern, to shift cursor or screen etc. The different commands and their functions are as given in the datasheet.

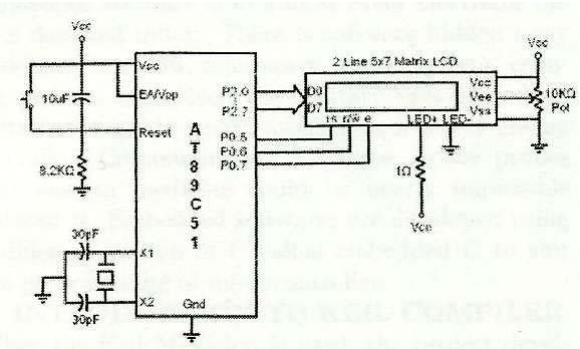
In order to avoid improper transfer of data to the LCD, the respective data bits should be present at the data bus for a specific amount of time. One way is to check the BUSY bit found on data line D7. This is not the best method because LCD's can get stuck, and program will then stay forever in a loop checking the BUSY bit. The other way is to introduce a delay in the program. The delay has to be long enough for the LCD to finish the operation in process. Instructions for writing to and reading from an LCD memory are shown in the previous table. At the beginning we mentioned that we needed 11 I/O lines to communicate with an LCD. However, we can communicate with an LCD through a 4-bit data bus. Thus we can reduce the total number of communication lines to seven. The wiring for connection via a 4-bit data bus is shown in the diagram below. In this example we use an LCD display with 2x16 characters.

1. Features

(a) 16 Characters x 2 Lines.
(b) 5x7 Dot latrix Character + Cursor.
(c) HD44780 Equivalent LCD Controller/driver Built-In.

(d) 4-bit or 8-bit MPU Interface.

Item Interfacing Circuit[Only LCD and uC]



2. Pin Description[connection of all 16 pins of LCD]

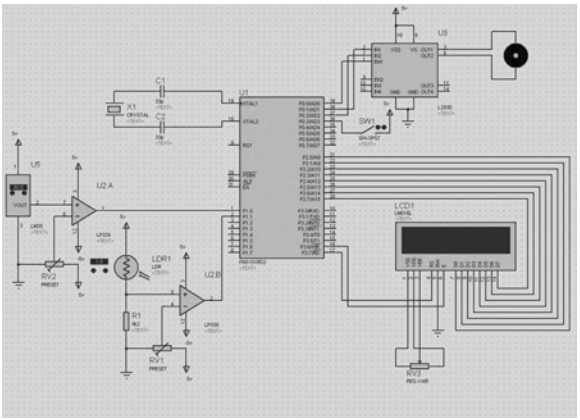
Pins	Descrpition
D0-D7	Data Lines
RS	Register Select
R/W	Read or Write
E	Enable Display Logic
LED+	Supply to LCD
LED-	Gnd to LCD
Vcc	Source Voltage
Gnd	ground

3. Checking and initializing:
(a) adjust contrast using pot
(b) connections of each pin with controller
(c) checking enabling signal

7 Circuit Working Details

The automation in done with the help of Temperature sensor and LDR. The temperature sensor, LM35, gives an analog output of the temperature and this is fed to the non-inverting terminal of comparator. The inverting terminal of the comparator is connected to a 5V, which is connected in series with a preset to vary the voltage supplied to it. The output of the comparator is connected to pin P1.0 of microcontroller. The LDR in series with the resistor is connected to the non-inverting terminal of the second comparator. The input to this terminal is the voltage across the resistor. The inverting terminal of the second comparator follows the same procedure as the inverting terminal of first comparator. The output of the comparator is connected to pin P1.1 of microcontroller. If the particular state of curtain(open or close) is not satisfactory to the user, there is an external switch, which will do the opposite of the present state condition of curtain, ie, if the curtain is open, the switch will make it close and vice versa. This external switch is connected to pin P0.3 of the microcontroller. Depending on various conditions of Temperature sensor and LDR, the microcontroller will drive the dc motor, through motor driver IC L293D. Pin P0.0 is used for enabling the motor driver IC L293D. Pin P0.1 and P0.2 are the inputs to the L293D which will rotate the dc motor accordingly.

Circuit Diagram



8 Keil-an overview

1. EMBEDDED LANGUAGE:
Embedded software is in almost every electronic device designed today. There is software hidden away inside our watches, microwave, music sys-

tem, cellular phones, etc. Even the military uses embedded software to guide smart missiles and detect enemy aircraft. Communication satellites, space probes and modern medicine could be nearly impossible without it. Embedded softwares are developed using a different version of C called embedded C to suit the programming of microcontroller.

2. INTRODUCTION TO KEIL COMPILER:
   When the Keil Micro-Vision is used, the project development cycle is roughly the same as it is for any other software development project.
   (a) Create source file in C or assembly.
   (b) Build application with the project manager.
   (c) simulation of the project manager.
   (d) Correct errors in source file.
   (e) Test the linked application.

3. MICRO-VISION IDE:
   The MICRO-Vision IDE combines project managements, a rich featured editor with interactive error correction, option setup make facility, and online help into a single entity. Using MVision, one can create source files and organize them into a project that defines one's target application. M-Vision automatically compiles, assembles and links one's embedded application and provides a single focal point for one's development efforts.

4. C51 COMPILER AND A51 MACRO ASSEMBLER:
   The source file is created by M-Vision IDE and passed to the C51 compiler macro assembler. The compiler and assembler process the source files and create relocatable object files. The Keil C51 compiler is a full ANSI implementation of the C programming language that supports all standard features of the C language.

5. LIB51 LIBRARY MANAGER:
   The LIB 51 lib manager allows you to create object library from the object file created by the compiler and assembler. Libraries are specially, ordered collection of object modules that may be used by the linker at a later time. When the linker processes a library, only those object modules in the library that are necessary to create the program are used.

6. BL 51 LINKER/LOCATOR:
   The BL 51 linker/locator creates an absolute ELF/DWARF files using the object module extracted from libraries and those created by the compilers and assembler. An absolute object file or module contains no relocatable code and data reside at a fixed memory location. The absolute ELF/DWARF file is used:
   (a) To program an flash ROM or other memory devices with M-Vision debugger for simulation and target debugging.
   (b) With an in-circuit emulator for the program testing.

7. MICRO-VISION DEBUGGER:
   M-Vision symbolic source level debugger is ideally suited for fast, reliable program debugging. The debugger includes a high-speed simulator that can simulate an entire AT89C51 system including on-chip peripherals and external hardware. The attributes of the chip used are automatically configured when device is selected from device database. The M-Vision debugger provides several ways for testing programs on real target hardware.

9 Conclusion

The circuit is successful in working as an Intelligent Curtain. It also has various technically sound and precise applications in day-to-day systems. It utilizes the basic concepts behind the working of air conditioning and heating systems.

10 References

1. Bookname:The 8051 Microcontroller And Embedded Systems
   Authors:Muhammad Ali Mazidi, Janice Gille spie Mazidi, Rolin D. Mckinlay
   Publication: Pearson Education.

2. Datasheets