A

PROJECT REPORT

On

"AUTO INTENSITY CONTROL OF STREET LIGHTS"

In partial fulfillment of the requirement for four years

B.TECH

In

ELECTRICAL and ELECTRONICS ENGINEERING

TECHNOLOGY EDUCATION & RESEARCH INTEGRATED INSTITUTIONS

SUBMITTED TO:

Er. Rajesh Kumar

(H.O.D. EEE)

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

TERII, KURUKSHETRA

PROJECT TEAM:

- **Ravi Prakash Singh**
  B.TECH EEE 4TH YEAR
  3517217
- **Vivek Tiwari**
  B.TECH EEE 4TH YEAR
  3517218
- **Rajnish Kumar**
  B.TECH EEE 4TH YEAR
  3517220
- **Prince Kumar**
  B.TECH EEE 4TH YEAR
  3517302

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to **Er. Rajesh Kumar** for his guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my gratitude towards my team members for their kind co-operation and encouragement which help me in completion of this project.

I would like to express my special gratitude and thanks to industry persons for giving me such attention and time.

My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

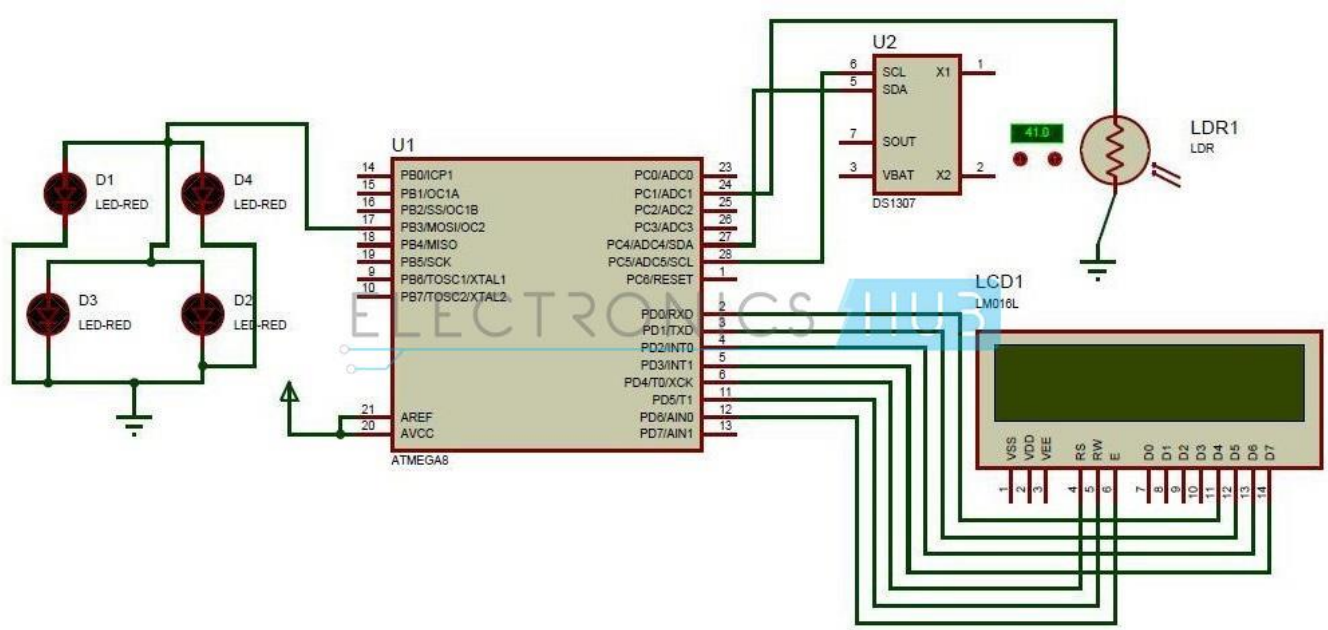
CONTENTS

Circuit Principle:

The main principle of this project is to Control the intensity of street lights using PWM. Peak hours of a particular area are calculated and accordingly PWM signal is adjusted by microcontroller to increase or decrease the intensity of street lights.

These peak hours can be calculated by considering parameters like traffic density, time, and light intensity of the environment.

Auto Intensity Control of Street Lights Circuit Diagram :



Circuit Components:

- ATmega8 micro controller
- DS1307 IC
- Light Dependent Resistor
- LED array
- LCD display.

Circuit Design:

The auto intensity control of street lights circuit is simple but it requires more coding part. This circuit consists of Atmega8 controller, DS1307, LDR, Relay and LEDs.

LDR: LDR is used for calculating the light intensity of the environment .The light dependent resistor is connected to ADC1 (PC1) pin of the micro controller. The analog light value is converted to digital value using ADC.

RTC: Current time is calculated using RTC. Real time clock has 8 pins out of which SCL and SDA are connected to PC5 and PC4 pins respectively. SCL is serial clock while SDA is serial data RTC is I2C compatible, where I2C means inter integrated circuit. One bit of data is transmitted on data bus for each clock cycle.

Data can be transferred between devices, using only two bi-directional buses. Each device can act as a slave or master. The slave devices will have one address and these devices can be accessed using this address.

LCD: LCD is the display used for displaying time which is read from RTC IC. Interfacing of LCD in 4-bit mode is shown in circuit diagram. D4-D7 pins of LCD are connected to PD0-PD3 pins of microcontroller.

RS pin of LCD is connected to PD4 pin of micro controller. RW and Enable pins are connected to PD5 and PD6 pins of controller.

LED array is number of high power LEDs connected in series. It is connected to PWM pin of the microcontroller.

ATmega8 Microcontroller:

It is an 8 bit CMOS technology based microcontroller belongs to the AVR family of microcontroller developed in 1996. It is built on RISC (Reduced Instruction Set Computer) architecture. Their main advantage is it doesn't contain any accumulator register and the result of any operation can be stored inside any register, defined by an instruction.

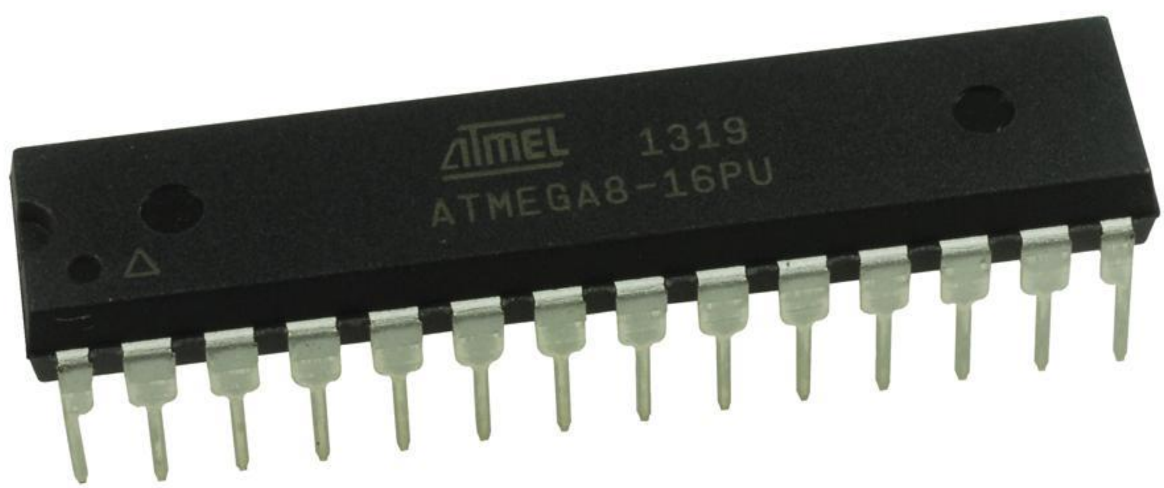


Fig. ATmega8 Microcontroller

It is an 8-bit AVR microcontroller that is based on RISC CMOS technology and comes with 28-pin interface for PDIP package. The Program memory is 8K Flash while RAM and EEPROM are 1K and 512 bytes respectively. Microchip has been the main source for producing PIC and AVR microcontrollers that are mainly used in embedded and industrial automation systems. These modules can perform a number of functions on a tiny chip, preventing you from spending too much and purchasing external components for laying out automation in the relevant project.

ATmega8 Microcontroller Architecture:

The block diagram representation of internal architecture configuration of ATmega8 microcontroller is:

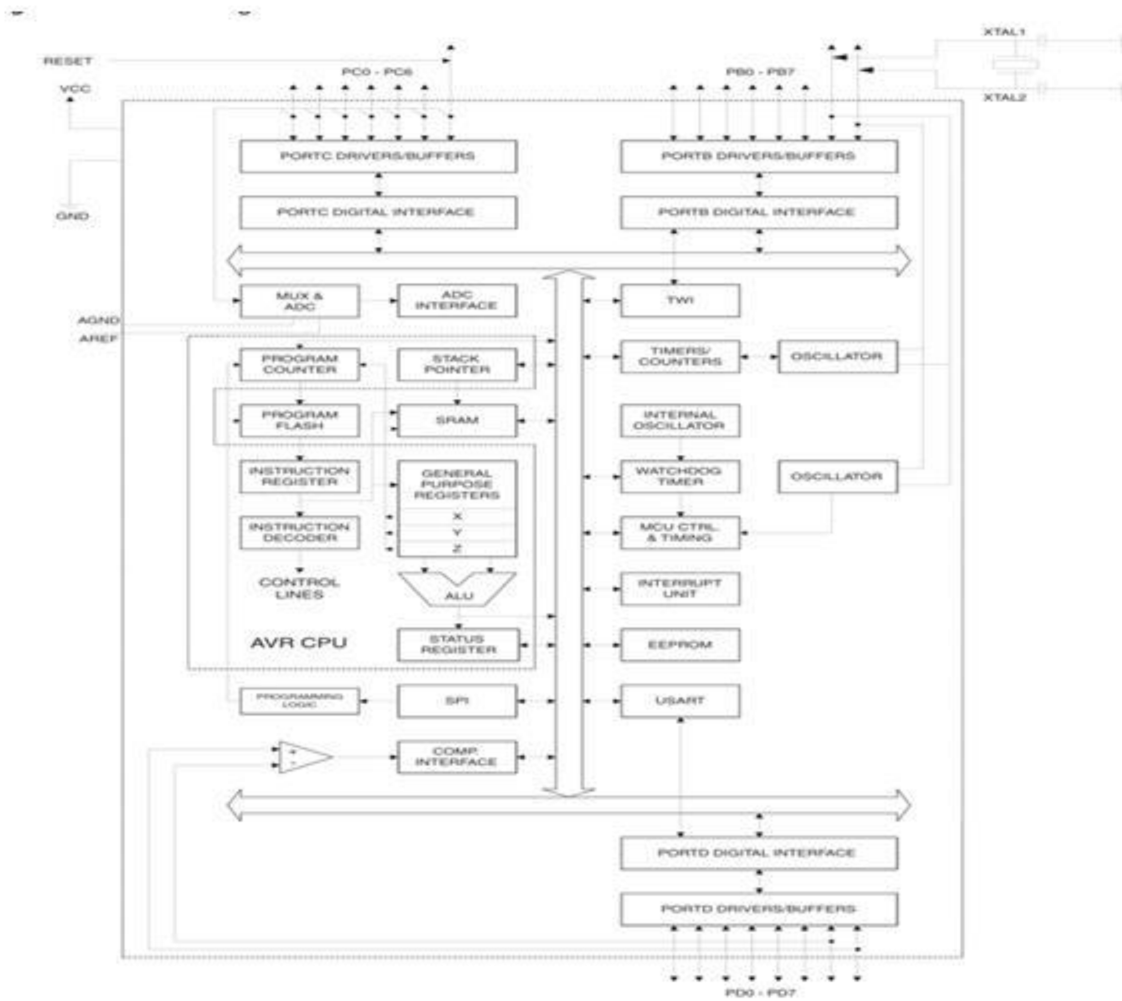


Fig. ATmega8 Microcontroller Architecture

Pin Diagram of ATmega8 Microcontroller:

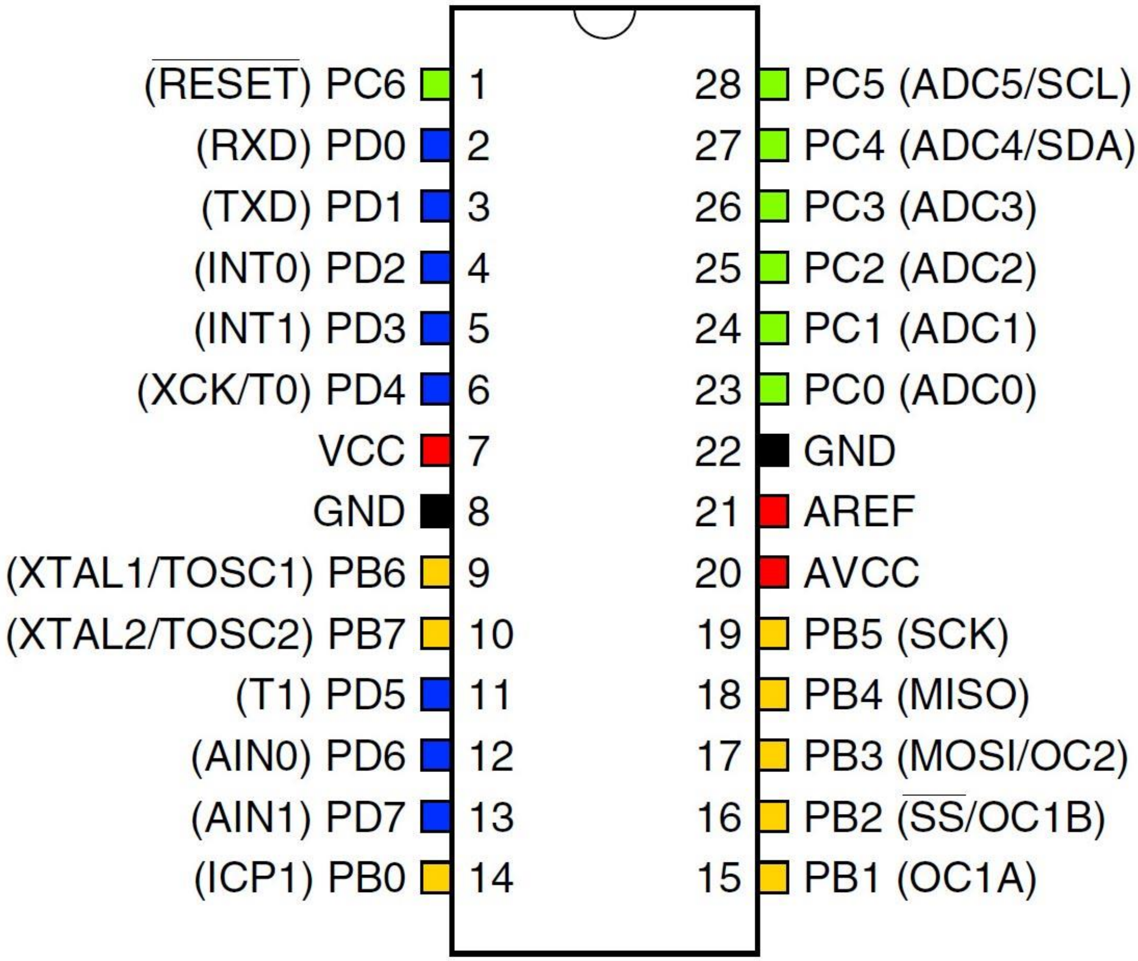


Fig. ATmega8 Microcontroller Pin Diagram

One of the most important features of ATmega8 microcontroller is that except 5 pins, all other pins can be used for supporting two signals.

- Pins 9,10,14,15,16,17,18,19 are used for port B, Whereas Pins 23,24,25,26,27,28 and 1 are used for port C and Pins 2,3,4,5,6,11,12 are used for port D.
- Pin 1 is used as Reset pin and on applying low level signal for time longer than minimum pulse length will generate a reset signal.
- Pins 3 and 2 can also be used in serial communication for USART (Universal Synchronous and Asynchronous Receiver Transmitter).
- Pin 5 and 4 are used as external interrupts.
- Pins 10 and 9 are used as timer counter oscillators as well as external oscillator where the crystal is connected directly between the pins.
- Pin 19 is used as slave clock input or master clock output for Serial Peripheral Interface (SPI) channel.
- Pin 18 is used as slave clock output or master clock input
- Pins 23 to 28 are used for analog to digital conversion (ADC) channels.
- Pin 12 and 13 are used as Analog Comparator inputs.
- Pins 6 and 11 are used as counter/timer sources.

Pin No.	**Pin name**	**Description**	**Alternate Function**
1	PC6 (RESET)	Pin6 of PORTC	Pin by default is used as RESET pin. If the RSTDISBL Fuse is programmed, PC6 can be used as an I/O pin. (Pulled HIGH to RESET controller)
2	PD0 (RXD)	Pin0 of PORTD	RXD (USART Input Pin) USART Serial Communication Interface [Can be used for programming]
3	PD1 (TXD)	Pin1 of PORTD	TXD (USART Output Pin)

			USART Serial Communication Interface [Can be used for programming] INT2(External Interrupt 2 Input)
4	PD2 (INT0)	Pin2 of PORTD	External Interrupt INT0
5	PD3 (INT1)	Pin3 of PORTD	External Interrupt INT1
6	PD4 (XCK/T0)	Pin4 of PORTD	T0(Timer0 External Counter Input) XCK (USART External Clock I/O)
7	VCC		
8	GND		
9	PB6 (XTAL1/TOSC1)	Pin6 of PORTB	XTAL1 (Chip Clock Oscillator pin 1 or External clock input) TOSC1 (Timer Oscillator pin 1)
10	PB7 (XTAL2/TOSC2)	Pin7 of PORTB	XTAL2 (Chip Clock Oscillator pin 2) TOSC2 (Timer Oscillator pin 2)
11	PD5 (T1)	Pin5 of PORTD	T1(Timer1 External Counter Input)
12	PD6 (AIN0)	Pin6 of PORTD	AIN0(Analog Comparator Positive I/P)
13	PD7 (AIN1)	Pin7 of PORTD	AIN1(Analog Comparator Negative I/P)

14	PB0 (ICP1)	Pin0 of PORTB	ICP1(Timer/Counter1 Input Capture Pin)
15	PB1 (OC1A)	Pin1 of PORTB	OC1A (Timer/Counter1 Output Compare Match A Output)
16	PB2 (SS/OC1B)	Pin2 of PORTB	SS (SPI Slave Select Input). This pin is low when controller acts as slave. [Serial Peripheral Interface (SPI) for programming] OC1B (Timer/Counter1 Output Compare Match B Output)
17	PB3 (MOSI/OC2)	Pin3 of PORTB	MOSI (Master Output Slave Input). When controller acts as slave, the data is received by this pin. [Serial Peripheral Interface (SPI) for programming] OC2 (Timer/Counter2 Output Compare Match Output)
18	PB4 (MISO)	Pin4 of PORTB	MISO (Master Input Slave Output). When controller acts as slave, the data is sent to master by this controller through this pin. [Serial Peripheral Interface (SPI) for programming]
19	PB5 (SCK)	Pin5 of PORTB	SCK (SPI Bus Serial Clock). This is the clock shared between this controller and other system for accurate data transfer. [Serial Peripheral Interface (SPI) for programming]
20	AVCC		Vcc for Internal ADC Converter
21	AREF		Analog Reference Pin for ADC
22	GND		GROUND
23	PC0 (ADC0)	Pin0 of PORTC	ADC0 (ADC Input Channel 0)

24	PC1 (ADC1)	Pin1 of PORTC	ADC1 (ADC Input Channel 1)
25	PC2 (ADC2)	Pin2 of PORTC	ADC2 (ADC Input Channel 2)
26	PC3 (ADC3)	Pin3 of PORTC	ADC3 (ADC Input Channel 3)
27	PC4 (ADC4/SDA)	Pin4 of PORTC	ADC4 (ADC Input Channel 4) SDA (Two-wire Serial Bus Data Input/Output Line)
28	PC5 (ADC5/SCL)	Pin5 of PORTC	ADC5 (ADC Input Channel 5) SCL (Two-wire Serial Bus Clock Line)

ATmega8 Microcontroller Sleep Modes:

The Microcontroller operates in 5 sleep modes as given below:

- **Power save Mode:** It is used when Counter/Timer is clocked asynchronously. In general this mode used for saving the operational power requirement of microcontroller.
- **Idle Mode:** It stops the functioning of CPU, but allows operation of ADC, TWI, SPI, and interrupts system and Watchdog. It is achieved by setting SM0 to SM2 bits of Microcontroller Unit register flag at zero.
- **Power down Mode:** It enables external interrupts, the 2-wire serial interface, and watchdog while disabling the external oscillator. It stops all generated clocks.
- **ADC Noise Reduction Mode:** It stops the central processing unit but allows the functioning of ADC, timer/counter and external interrupts.

- **Stand By mode:** In this mode, only oscillator is allowed to operate by slowing all other operation of microcontroller.

ATMEGA8 Features:

ATMEGA8 –Simplified Features	
CPU	8-bit AVR
Number of Pins	28
Operating Voltage (V)	+2.7 V TO +5.5 V (ATmega8L) +4.5 V TO +5.5 V (ATmega8) (+5.5V being absolute maximum)
Number of I/O pins	23
Communication Interface	Master/Slave SPI Serial Interface(16,17,18,19 PINS) [Can be used for programming this controller] Programmable Serial USART(2,3 PINS) [Can be used for programming this controller] Two-wire Serial Interface(27,28 PINS)[Can be used to connect peripheral devices like sensors and LCDs]
JTAG Interface	Not available
ADC Module	6 channels , 10-bit resolution ADC
Timer Module	Two 8-bit counters, One 16-bit counter [Total three]
Analog Comparators	1

DAC Module	Nil
PWM channels	3
External Oscillator	0-8MHz for ATMEGA8L 0-16MHz for ATMEGA8
Internal Oscillator	0-8MHz Calibrated Internal Oscillator
Program Memory Type	Flash
Program Memory or Flash memory	8Kbytes[10000 write/erase cycles]
CPU Speed (MIPS)	16 MIPS
RAM	1KBytes
EEPROM	512
Watchdog Timer	Programmable Watchdog Timer with Separate On-chip Oscillator
Program Lock	Yes
Power Save Modes	Six Modes[Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby]
Operating Temperature	-55°C to +125°C(+125 being absolute maximum, -55 being absolute minimum)

Applications:

There are hundreds of applications for ATMEGA8.

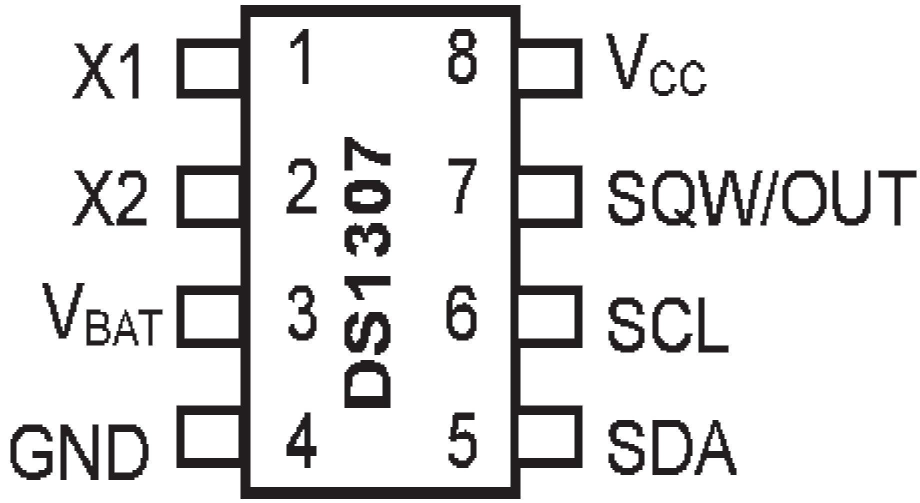
- Industrial control systems.
- SMPS and Power Regulation systems.
- Analog signal measuring and manipulations.
- Embedded systems like coffee machine, vending machine.
- Motor control systems.
- Display units.
- Peripheral Interface system.

DS1307 IC:

The **DS1307** is a low power Full Binary (BCD) Real Time Clock (RTC) IC with 56 bytes of SVRAM that communicates through I2C Protocol. The IC can work from directly supply on Vcc and switch to Battery automatically when required.

DS1307 IC

Pin Diagram of DS1307 IC:



Pin Diagram of DS1307 IC

Pin Configuration:

Pin Number	Pin Name	Description
1,2	X1 , X2	Crystal Oscillator should be connected to these pins
3	V-Bat	Connected to Positive terminal of the battery
4	Ground	Ground pin of the IC
5,6	SCL and SDA	Pins for I2C communication with CPU
7	SQW / Out	Square wave output driver pin to obtain square wave frequencies.
8	Vcc	Powers the IC typically 5V

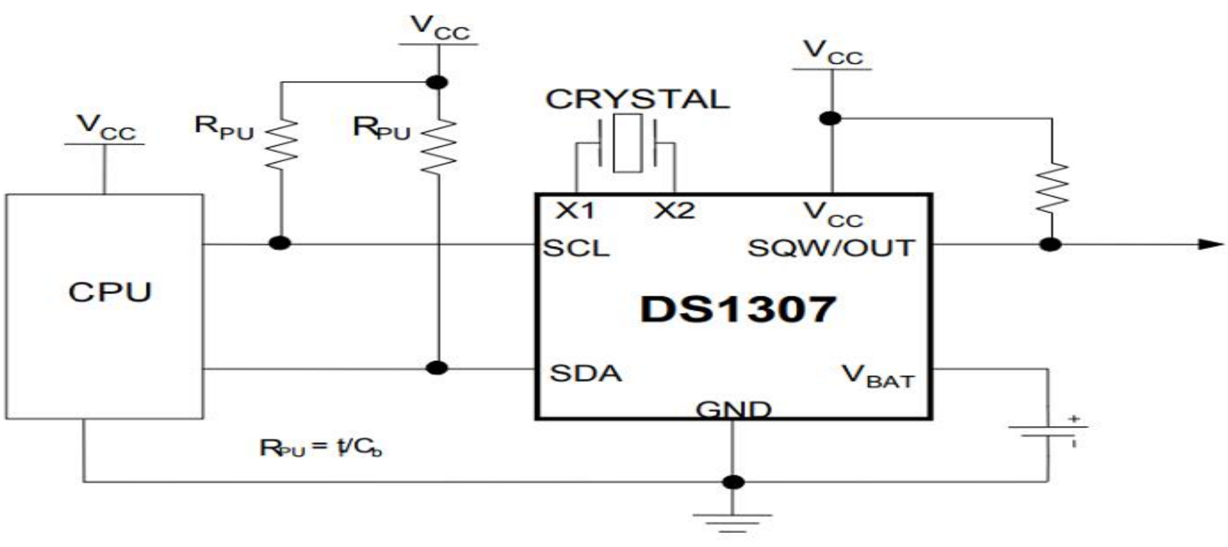
DS1307 Specifications:

- I2C Interface RTC IC
- Operating Voltage: 5V
- Less than 500nA current when operating with battery
- 56bytes SVRAM
- Operates in power or battery mode
- Programmable square wave output pin
- Available in PDIP and SO package.

Equivalent for DS1307: PCF8523, DS1820, DS1338

How to use DS1307:

The DS1307 is a 8-pin IC which operates on 5V and communicates with CPU though I2C protocol. A typical application circuit for the DS1307 from **DS1307 datasheet** is shown below.



As you can see the IC has SCL (Serial Clock) and SDA (Serial Data) pins using which it communicates to the CPU, both these pins has to be pulled high using a resistor. The IC can be powered by providing 5V on the Vcc pin, when the power fails it will automatically switch to battery operated mode in which it will obtain power with Lithium cell connected to pin Vbat and ground.

The pins X1 and X2 are used to connect the crystal oscillator; typically a 32.7KHz Quartz crystal is used. The SQW pin is used to provide a PWM square wave of programmable frequencies in the range of 1Hz, 4KHz, 8KHZ or 32KHz. This pin also requires a pull up resistor. Data between the CPU and RTC IC is exchanged only though the I2C protocol. Both Read and write happens through this communication. The IC can provide details like Real-Time Clock Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the Week, and Year with Leap-Year Compensation Valid Up to 2100.

Applications of DS1307 IC:

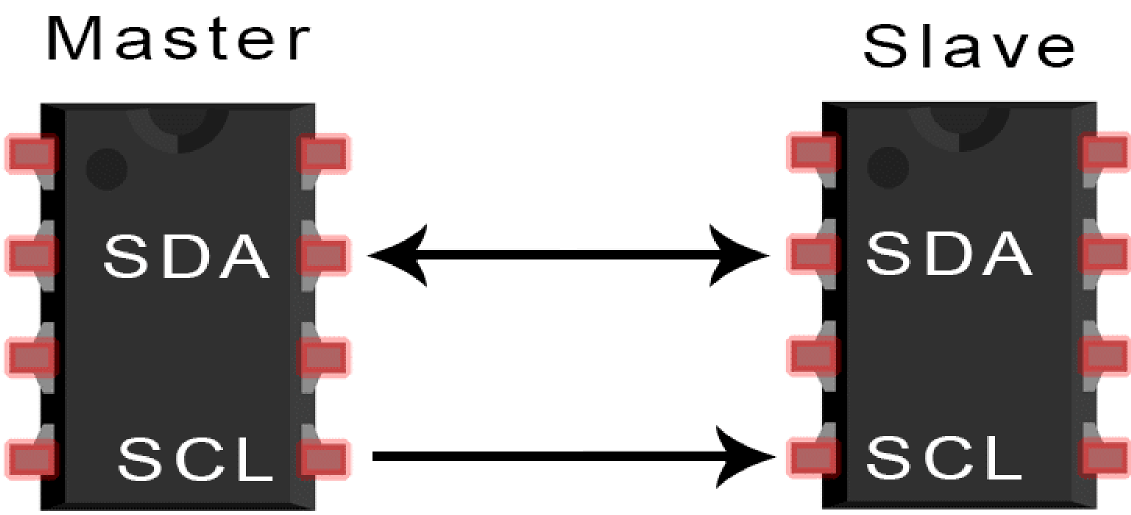
- Robotics
- Gaming
- Servers
- Computer Peripherals
- GPS
- Utility power meters

I2C Protocol:

I2c is a communication protocol invented by Philips Company. This is well suited for communication between integrated circuits and peripherals. This uses two lines to transfer data.

I2C combines the best features of SPI and UARTs. With I2C, you can connect multiple slaves to a single master (like SPI) and you can have multiple masters controlling single, or multiple slaves. This is really useful when you want to have more than one microcontroller logging data to a single memory card or displaying text to a single LCD.

Like UART communication, I2C only uses two wires to transmit data between devices:



- **SDA (Serial Data)** – The line for the master and slave to send and receive data.
- **SCL (Serial Clock)** – The line that carries the clock signal.

HOW I2C WORKS:

With I2C, data is transferred in *messages.* Messages are broken up into *frames* of data. Each message has an address frame that contains the binary address of the slave, and one or more data frames that contain the data being transmitted. The message also includes start and stop conditions, read/write bits, and ACK/NACK bits between each data frame

Start Condition: The SDA line switches from a high voltage level to a low voltage level *before* the SCL line switches from high to low.

Stop Condition: The SDA line switches from a low voltage level to a high voltage level *after* the SCL line switches from low to high.

Address Frame: A 7 or 10 bit sequence unique to each slave that identifies the slave when the master wants to talk to it.

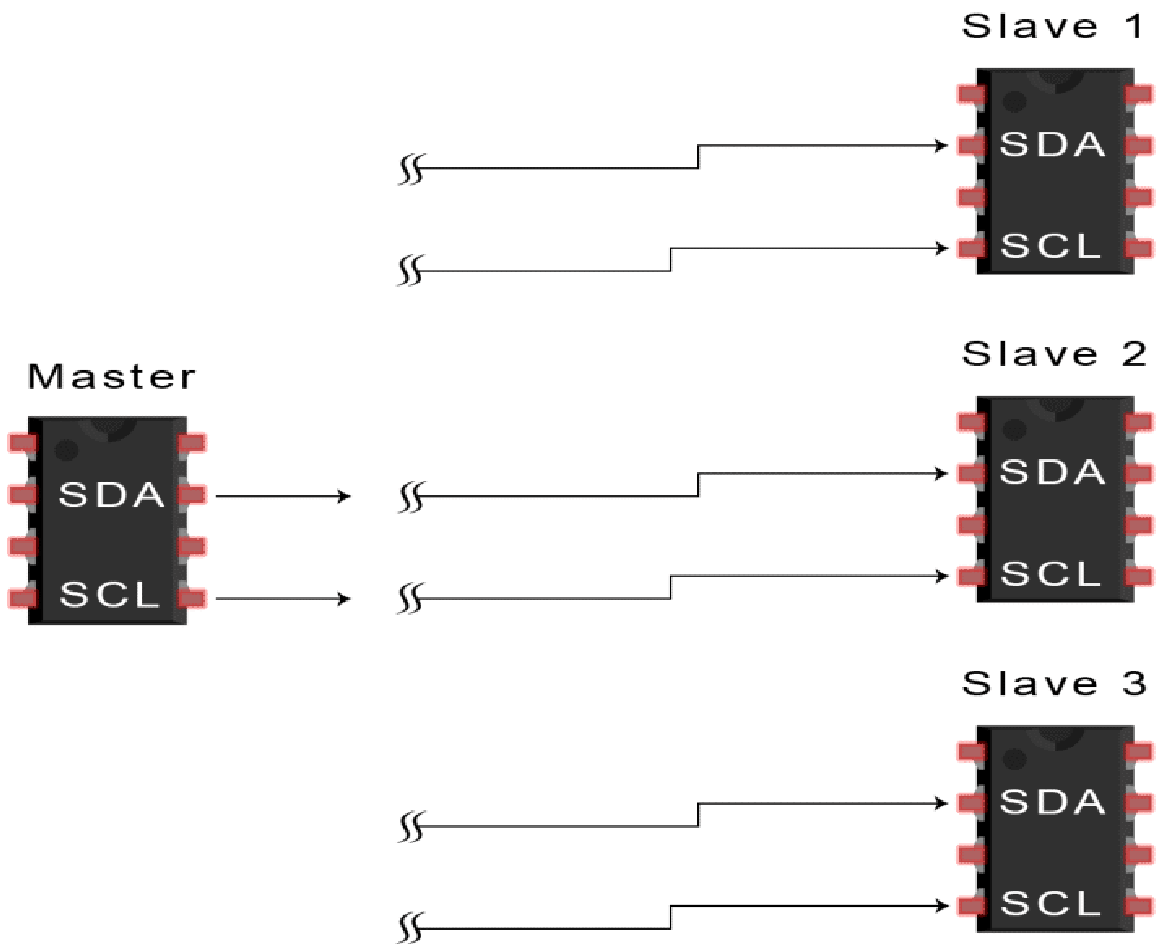
Read/Write Bit: A single bit specifying whether the master is sending data to the slave (low voltage level) or requesting data from it (high voltage level).

ACK/NACK Bit: Each frame in a message is followed by an acknowledge/no-acknowledge bit. If an address frame or data frame was successfully received, an ACK bit is returned to the sender from the receiving device.

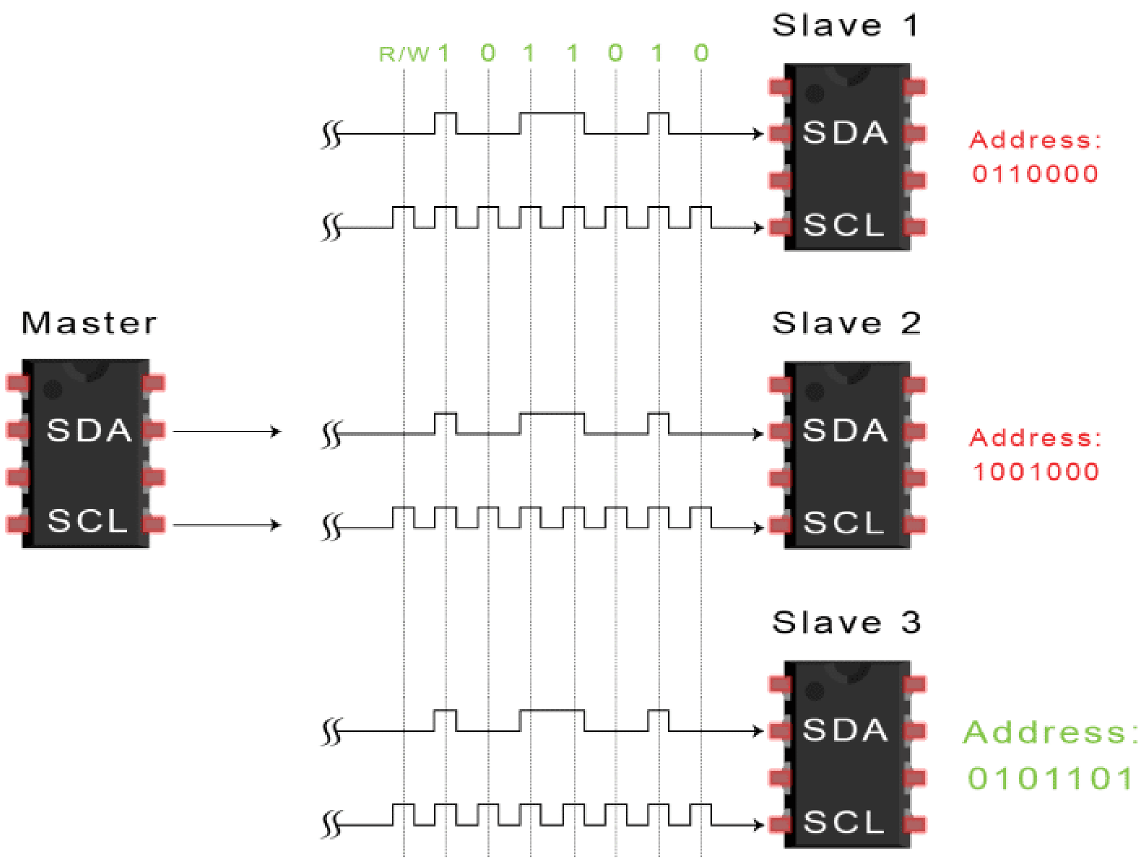
1. This can connect up to 128 devices using two wires. Each device connected will have an address. The device which initiates the data transfer is called Master.
2. Every device will have 7 bit address.
3. Master initially sends the START bit on the data line.
4. Then it sends the address of the device with which wants to communicate and the mode of operation i.e. read or write.
5. The slave devices listen to the incoming data and checks if its address matches to the received data. The device whose address matches send an acknowledgement signal.
6. Then master starts transmitting or receiving the data from the slave.
7. After completion of the transmission, Master sends a STOP bit.
8. Data on SDA can be changed only if SCL pin is low.

Steps for I2C Data Transmission:

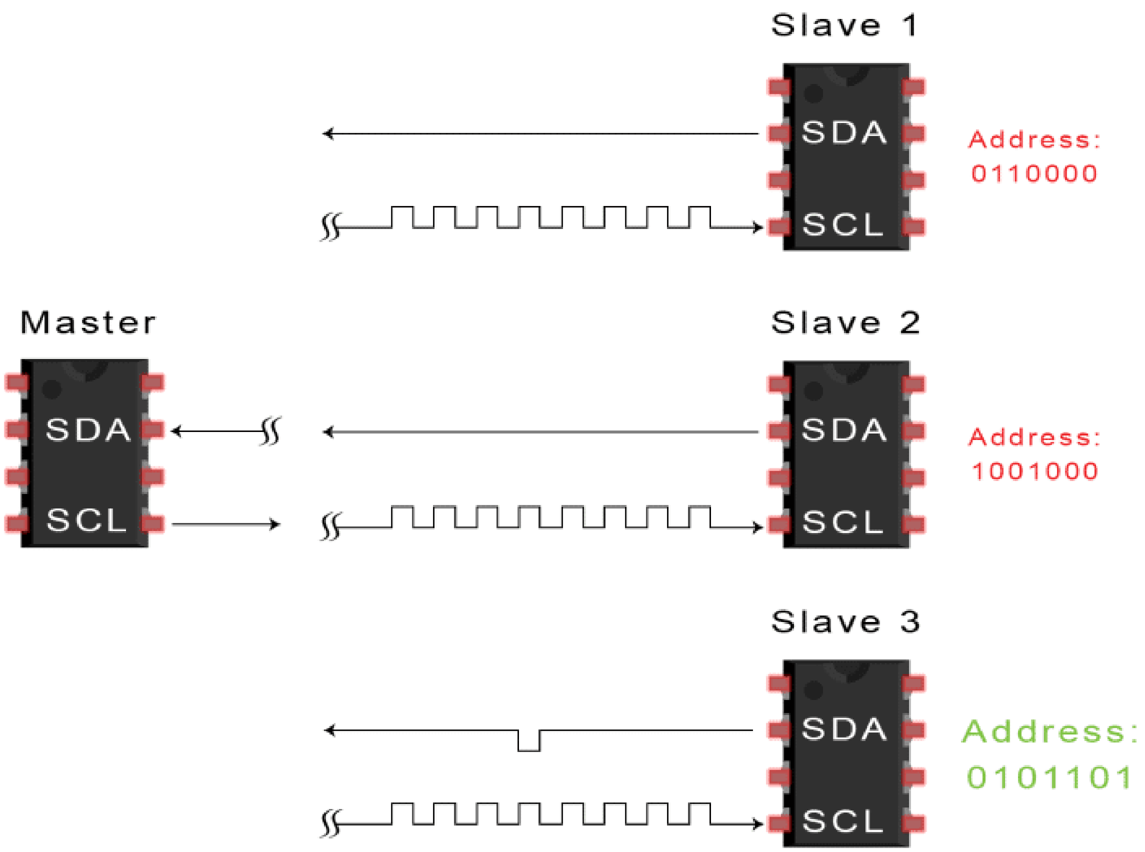
1. The master sends the start condition to every connected slave by switching the SDA line from a high voltage level to a low voltage level *before* switching the SCL line from high to low:



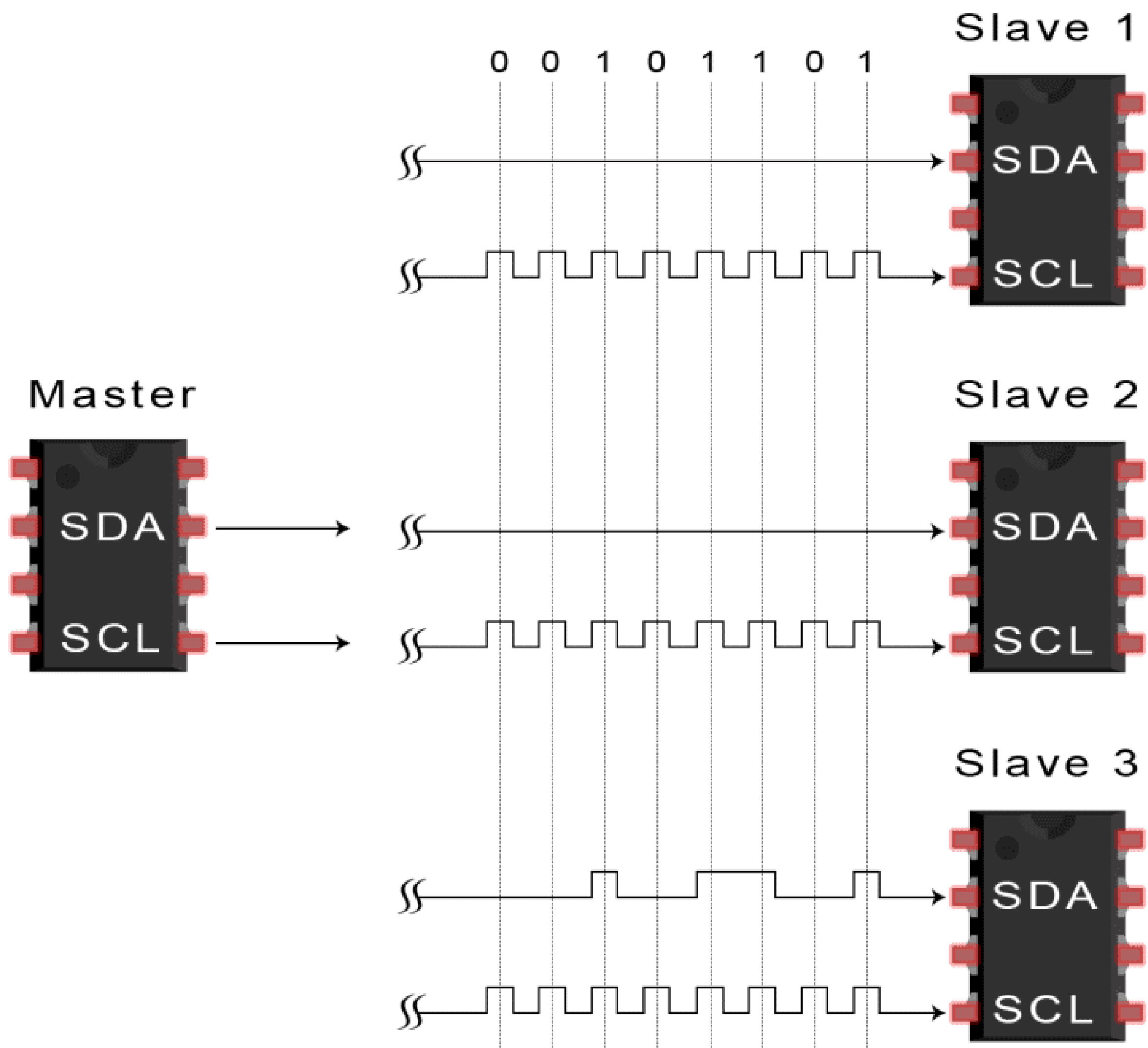
2. The master sends each slave the 7 or 10 bit address of the slave it wants to communicate with, along with the read/write bit



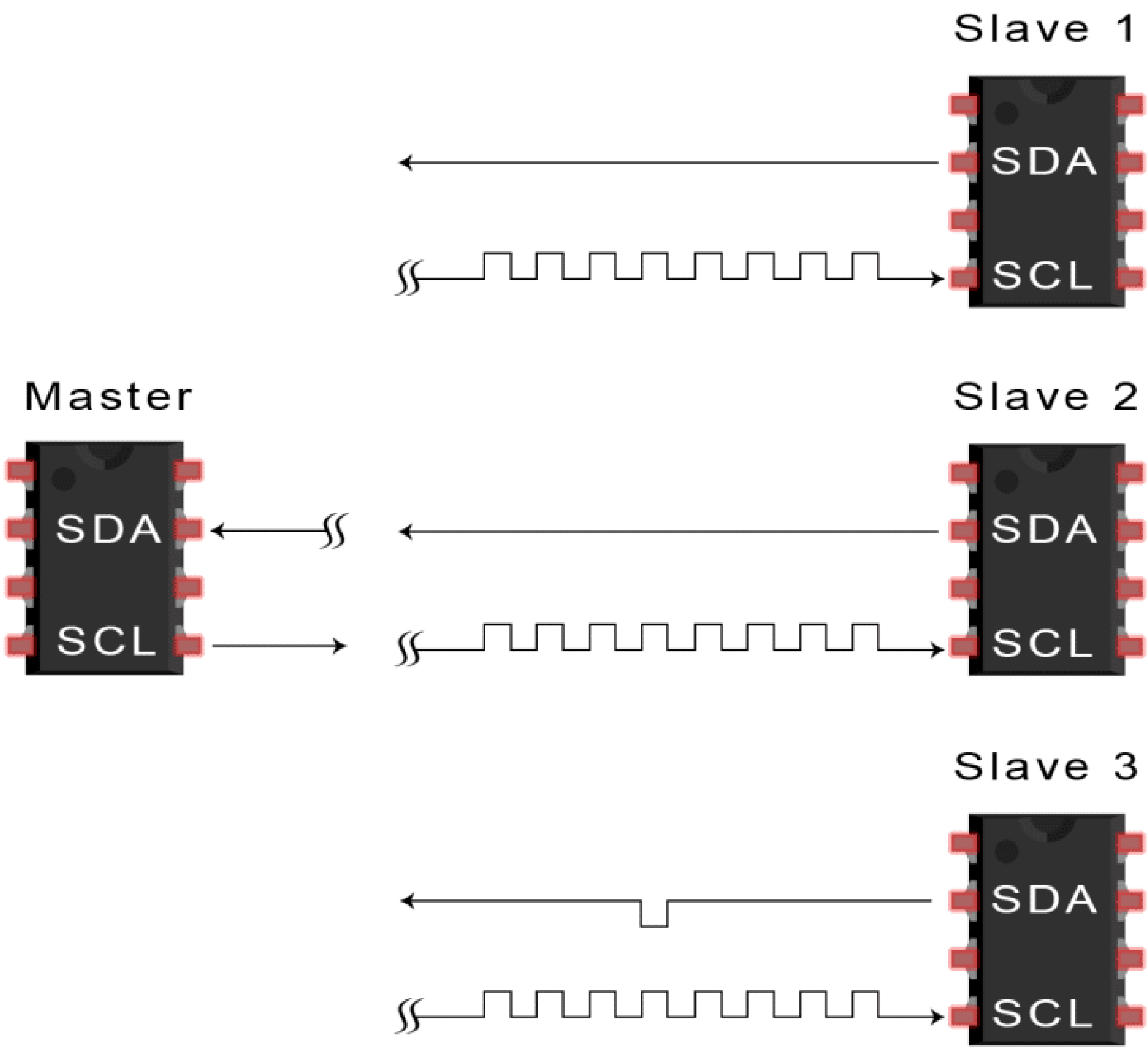
3. Each slave compares the address sent from the master to its own address. If the address matches, the slave returns an ACK bit by pulling the SDA line low for one bit. If the address from the master does not match the slave's own address, the slave leaves the SDA line high.



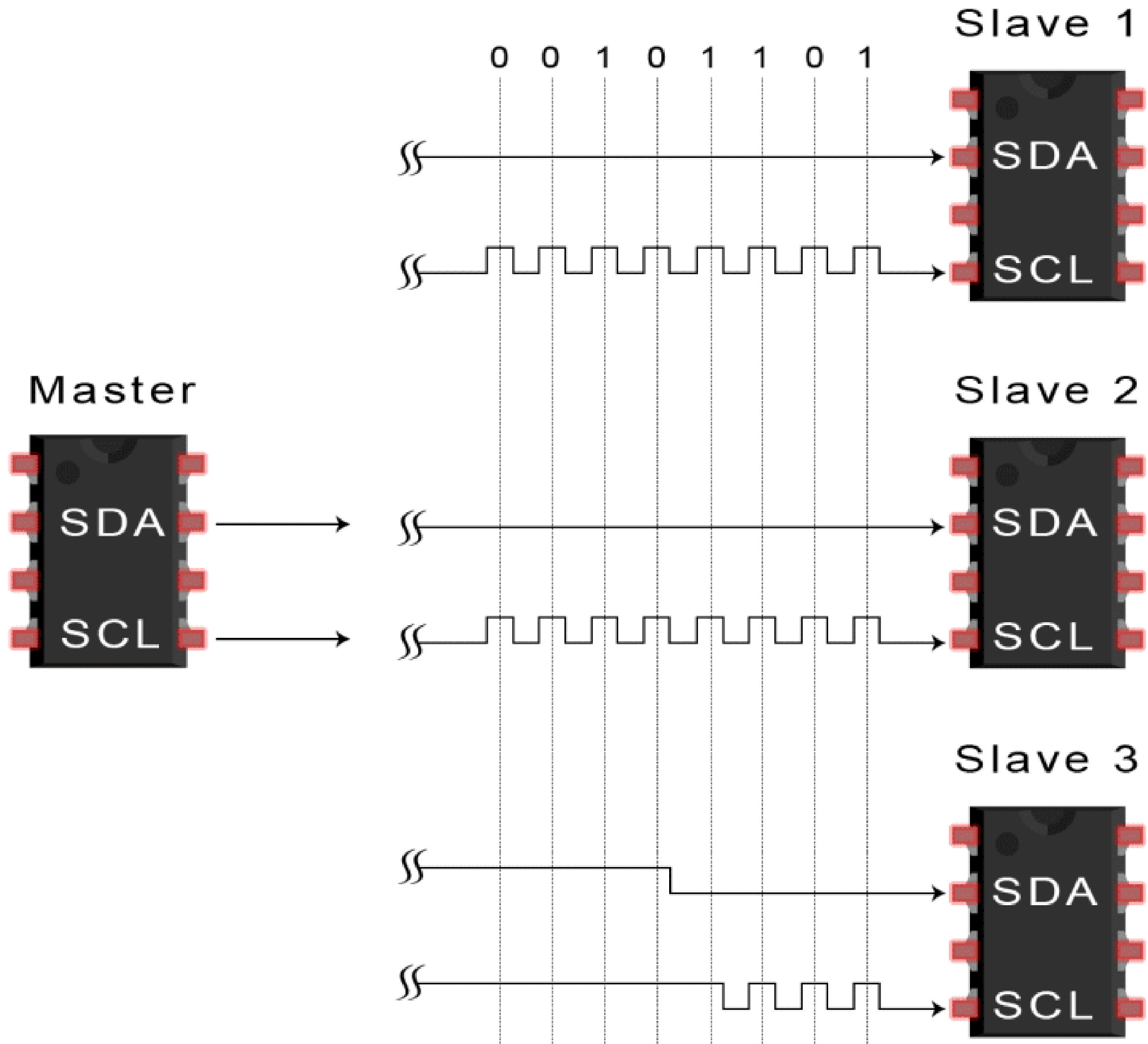
4. The master sends or receives the data frame:



5. After each data frame has been transferred, the receiving device returns another ACK bit to the sender to acknowledge successful receipt of the frame:



6. To stop the data transmission, the master sends a stop condition to the slave by switching SCL high before switching SDA high:



Advantages of I2C Protocol:

- Only uses two wires
- Supports multiple masters and multiple slaves
- ACK/NACK bit gives confirmation that each frame is transferred successfully
- Hardware is less complicated than with UARTs
- Well known and widely used protocol.

Disadvantages of I2C Protocol:

- Slower data transfer rate than SPI
- The size of the data frame is limited to 8 bits
- More complicated hardware needed to implement than SPI

Working of Intensity Control of Street Lights Circuit:

1. Initially power the circuit.
2. Time is displayed on the LCD display.
3. Place the LDR in darkness as the street lights switches on only when there is no light on LDR.
4. Now check the time if the time is between 9 pm to 2 am street light glows with full intensity.
5. From 2 pm intensity of the lights slowly starts decreasing and finally in early morning it glows with least intensity. When the light is sensed by the LDR lights are switched off automatically.

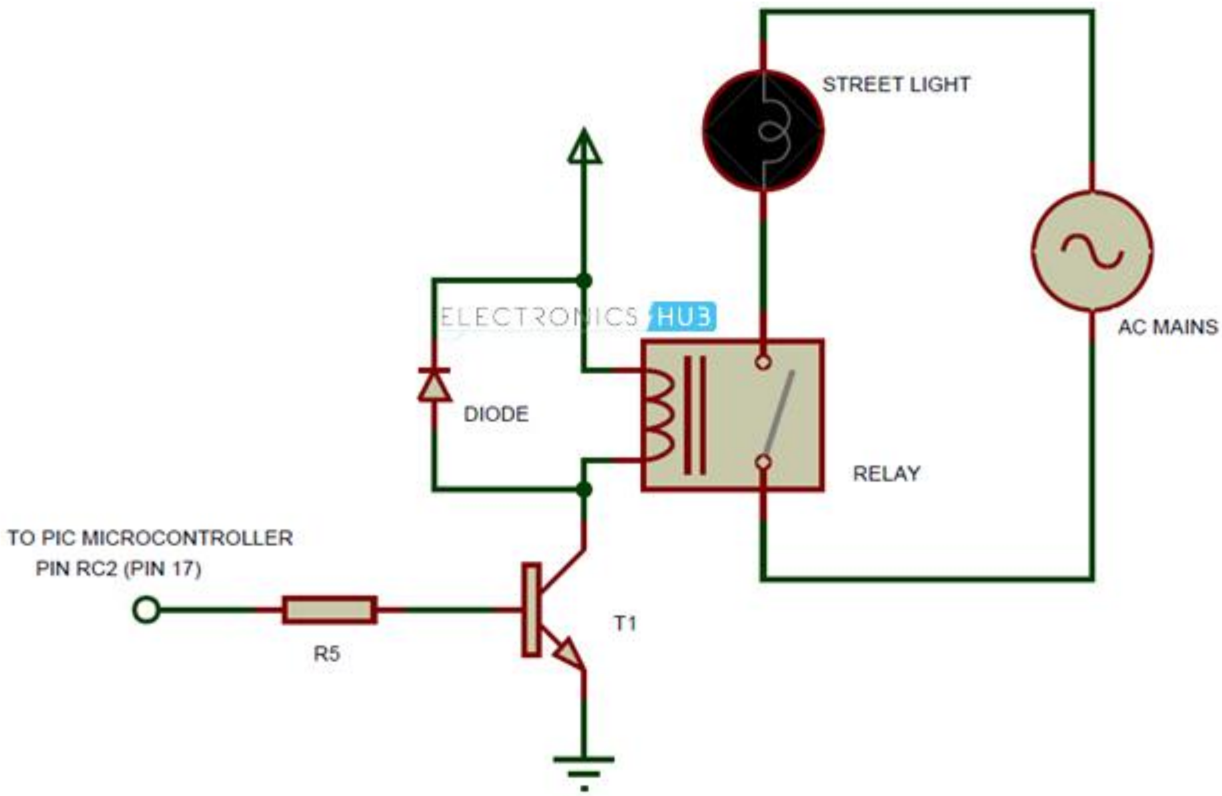
Code is written in such a way that up to 2 am lights will glow with full intensity. From then it slowly starts decreasing and finally it drops to zero in the morning.

Limitations of this Circuit:

- Even though energy is saved if there are any vehicles after fixed time, intensity of the light is low.
- Maximum energy cannot be saved.

Alternative Circuit:

The circuit shown above uses array of LEDs as street light in order to save power. But the same circuit can also be used to fire a normal HID street lamp. The circuit for auto intensity control of HID street light is as follows.



The above circuit shows only the interface to the street light and the rest of the circuit is same. It consists of a relay, a high intensity discharge street light that is connected to the mains supply and a diode.

The relay contact is made only when the intensity of light on the LDR is low and the street light glows.

Auto Intensity Control of Street Lights Circuit Advantages:

- Power wastage can be reduced.
- Using LED array reduces the cost.
- Using of RTC and LDR produces accurate results.

The End