

ABSTRACT

As we are aware that the number of accidents at railway crossing has been increasing lately, more precautionary measures are being undertaken. Accidents take place either due to the absence of crossing gates, or due to impatience and carelessness of pedestrians. . Whereas, there are places where the crossing gates at railroad is being controlled and managed by a guard. In such cases, threat may arise due to carelessness of the guard. So, our project aims at automating this process. Automatic Railway Gate Control (ARGC) is managed and controlled by microcontroller 8051 and is more reliable since there is no human intervention in the process. Our model, most primarily, is different from others in terms of the programming method and considering various possibilities that the actually railroad crossing system encounters. We have programmed 8051 using assembly language program (.asm). Infrared (IR) sensors are used which senses the presence of train and conveys the information to 8051. 8051 is programmed accordingly to open or close the gate whenever required.

PROJECT OBJECTIVE

This system is to manage the control system of railway gate using the microcontroller. The main purpose of this system is about railway gate control system and level crossing between railroad and highway for decreasing railroad-related accident and increasing safety

Project Components

Microcontroller Section

- AT89C51 MCU
- 11.0592 MHz QUARTZ CRYSTAL
- 2 x 33PF CERAMIC CAPACITOR
- 10 μ F / 16V ELECTROLYTIC CAPACITOR
- 10K Ω RESISTORS x 2
- AT89C51 PROGRAMMER BOARD
- LED (2)

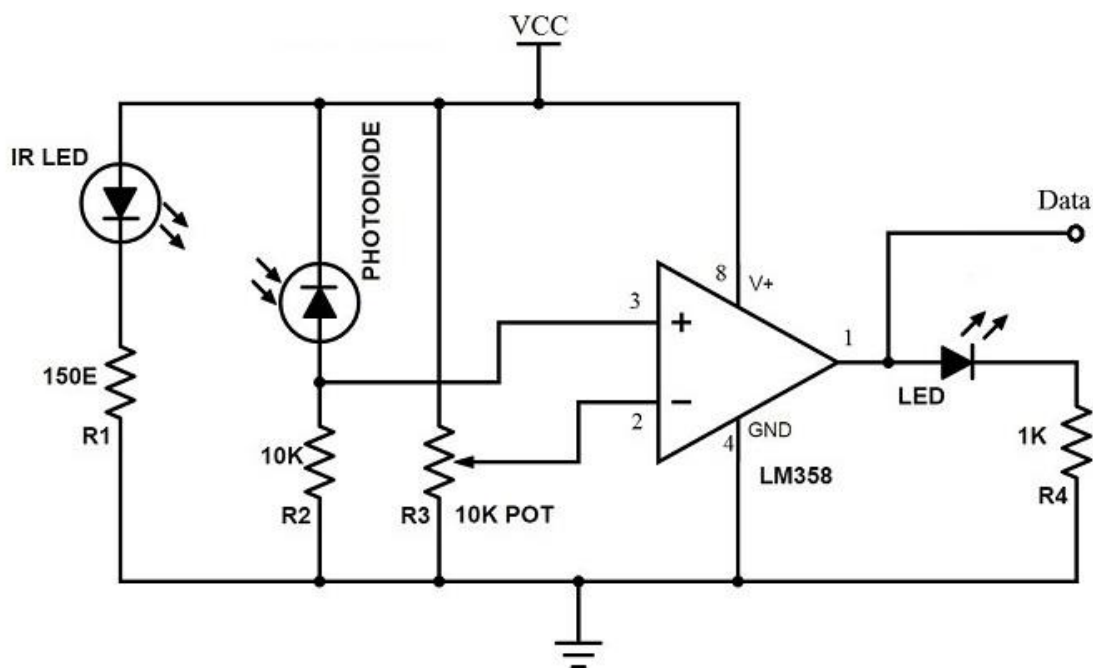
Sensor and Load Section

- 2 x REFLECTIVE TYPE IR SENSOR
- 2 x 1K Ω RESISTOR
- L293D MOTOR DRIVER IC
- MOTOR

Component Description

IR Sensor

- An IR sensor is used in this project to sense the arrival and departure of the train.
- An IR Sensor generally comprises of two components: an IR Transmitter and an IR Receiver. An IR Transmitter is a device that emits IR Rays.
- Similarly, an IR Receiver is a device that detects the IR Rays. Photo Diodes are the most commonly used IR Receivers.
- The following image shows the circuit of IR Sensor used in this project.



L293D Motor Driver

L293D is a motor driver IC used in this project to control the gate motors. With the help of this motor driver IC, we can control two motors at a time with both forward and reverse direction control for individual motors.

Motor drivers are generally used to drive high current drawing devices like DC Motors, stepper motors, high intensity lights, etc. They act as simple current amplifiers as their input is a low current signal usually from a microcontroller and their output is a high current signal to drive the loads.

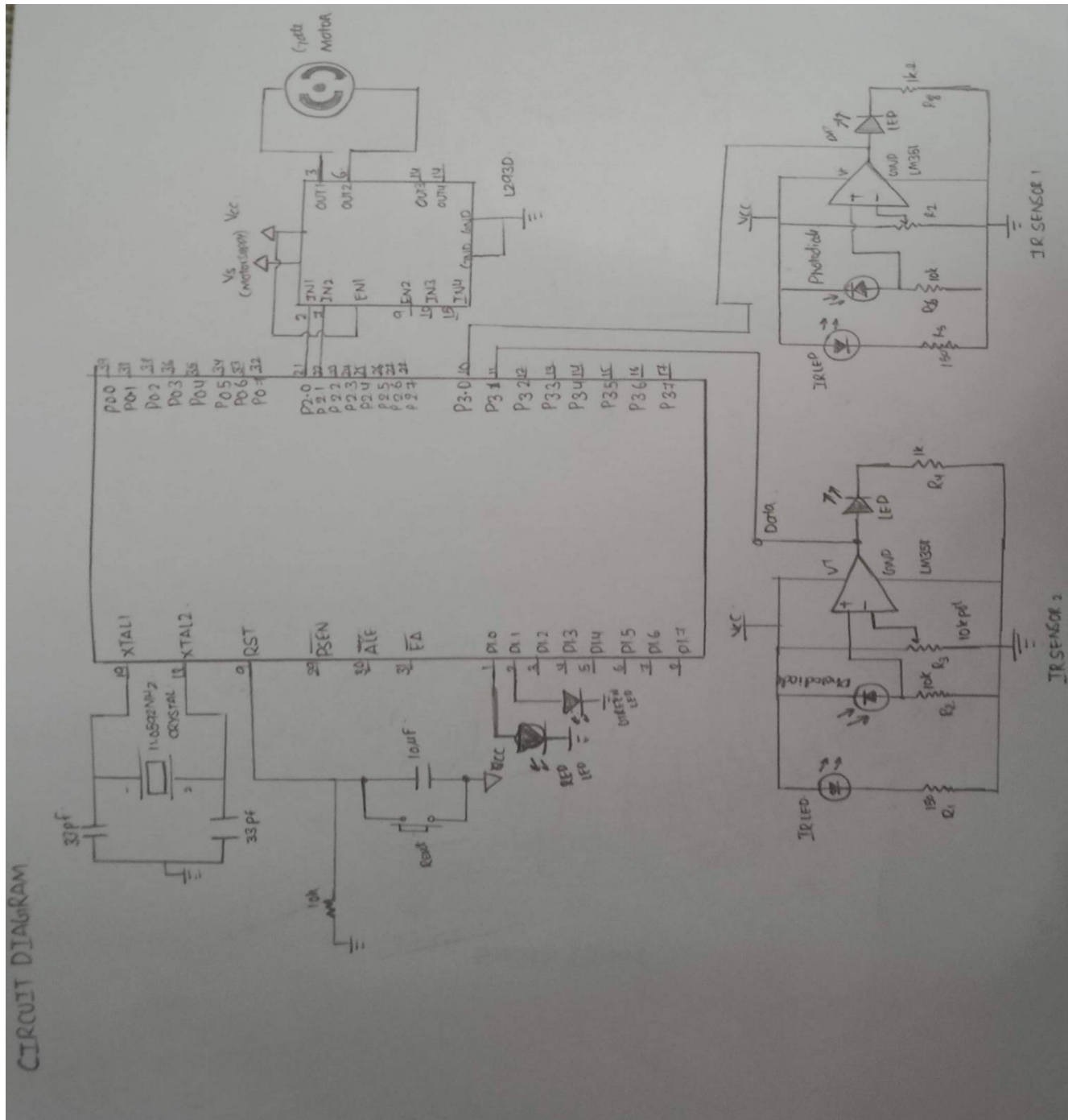
The basic operating mode of an H-bridge is fairly simple if Q1 and Q4 are turned on the left lead of the motor will be connected to the power supply, while the right lead is connected to ground. Current starts flowing through the motor which energizes the motor in (let's say) the forward direction and the motor shaft starts spinning.

The function of IN1 and IN2 pins is shown below Table

Function of IN1 and IN2 pins

IN1 / P2.0	IN2 / P2.1	Motor Status
Low	Low	Stops
Low	High	Clockwise
High	Low	Anti - Clockwise
High	High	Stop

CIRCUIT DIAGRAM



WORKING:

. Basically the circuit consists of IR LED-Photodiode pairs arranged in such a way that each pair is on either side of the track . When the train cuts the IR Sensor 1, the sensor senses its presence and notifies the 8051 microcontroller by sending an electrical signal. Usually, crossing gates are closed when train interrupts IR sensor 1 and are opened only when the train passes by the IR sensor 2. This is the basic principle of operation but there exists few other possibilities too. All the possibilities are taken under consideration while programming 8051. Various possibilities are as follows

Case 1:

Assuming the length of train is very long, the train cuts IR sensor 1 and gates are closed. Then it travels and interrupts IR sensor 2. Now both IR sensors are being blocked. As train passes further, IR sensor 1 is no longer blocked by the train but IR sensor 2 is still being blocked. Eventually train moves further and IR sensor 2 is no more interrupted and the gates are closed. Therefore, the gates will be opened only when both IR sensors are not blocked.

Case 2:

There is a possibility that train passes IR sensor 1 and waits in between. So in such cases, the gates must remain close and

open only when the train completely passes through IR sensor 2.

Case 3:

There is a possibility that there is only single engine passing through the sensors. When it blocks IR sensor 1 and moves a little further, IR sensor 1 is again unblocked. Now, the engine is somewhere between IR sensors 1 and 2 but both sensors are unblocked. As per case 1, when both IR sensors are open, the gates shall be opened. That logic is very dangerous in this case because the engine is running somewhere between the two sensor pairs but the gates are open. In order to tackle this issue, a new algorithm is designed. In the new algorithm, the gates will be opened only when there is a block-unblock transition in IR sensor 2 equivalent a similar transition in IR sensor 1.

CODE

Start: MOV P3, #0FFH ; IR connected to Port3. No block, IR's output is logic1. (Only for software simulation purpose)

CLR P2.0 ;connected to IN1 of L293D

CLR P2.1 ;connected to IN2 of L293D

CheckIR:

MOV R7, P3 ;IR (1) read.

MOV A, R7

RRC A

JNC MotorForward; ;Jump when CY=0 i.e. IR is blocked.

SJMP CheckIR

BACK1: MOV R7, P3 ;IR(2) read.

MOV A, R7

RRC A

RRC A

JNC motorReverse_Check_FirstIR

SJMP BACK1

MotorForward:

SETB P1.0

SETB P2.0 ;This loop closes the gate.

CLR P2.1

ACALL DELAY1

CLR P2.0

SJMP BACK1

MotorReverse_Check_FirstIR:

MOV R5,P3

MOV A,R5

RRC A

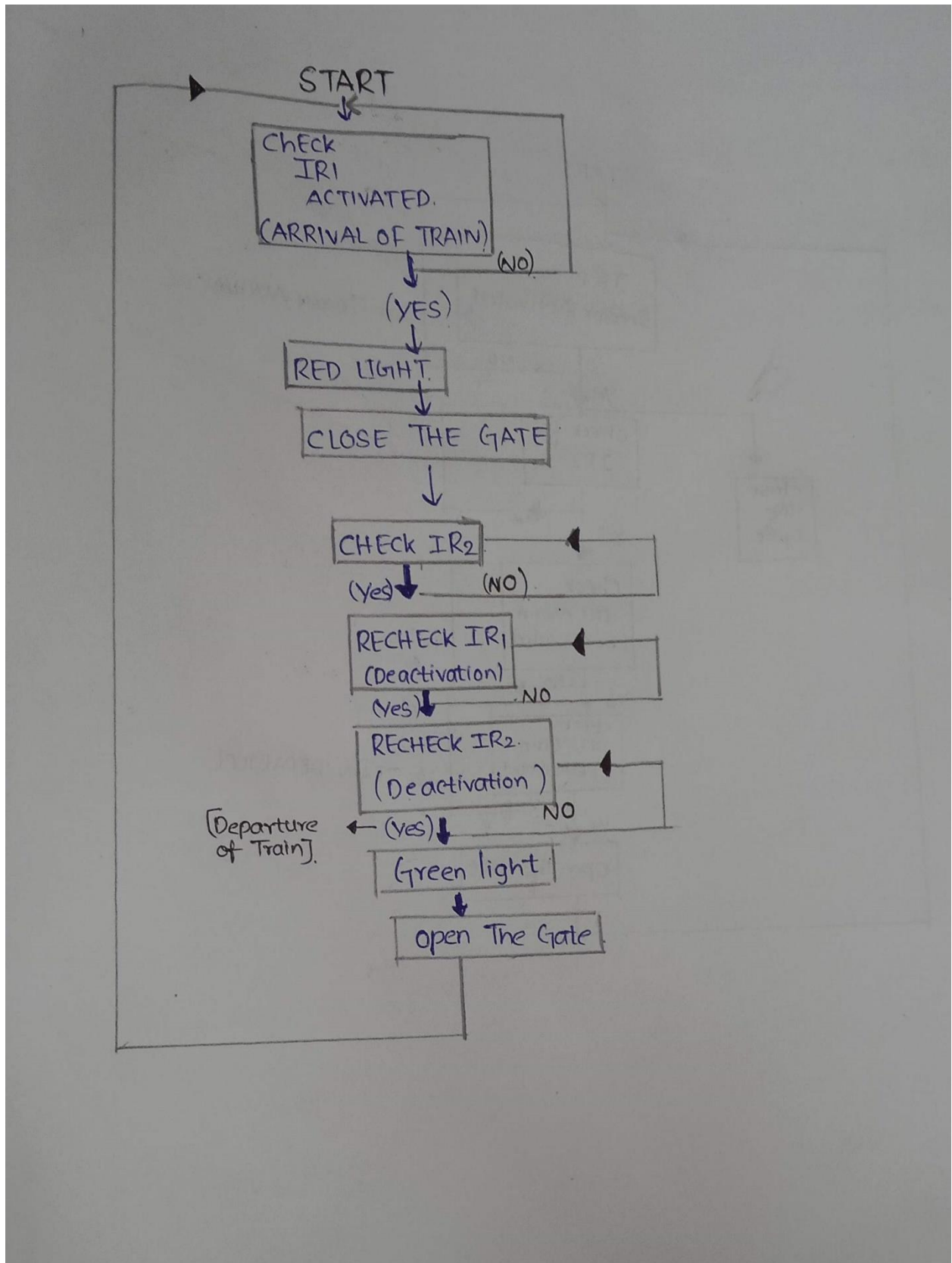
JNC MotorReverse_Check_FirstIR ;Keep looping when IR (1)
is cut. Go down once train crosses IR (1)

SecondIR_Check:

MOV R4,P3 ;Check IR (2) after making sure that
train has crossed IR(1)


```
MOV A,R4
RRC A
RRC A
JNC SecondIR_Check ;Keep looping when IR(2) is cut. Go
CLR P1.0 down when train crosses IR(2)
CLR P2.0 ;This loop opens the gate.
SETB P2.1
ACALL DELAY1
CLR P2.1
SETB P1.1
ACALL DELAY1
CLR P1.1
SJMP Start
DELAY1:MOV R4,#14H ;Delay program
OneSec:MOV TMOD,#01H
MOV TL0,#0AFH
MOV TH0,#3CH
SETB TR0
WAIT:JNB TF0,WAIT
CLR TR0
CLR TF0
DJNZ R4,OneSec
RET
END
```

FLOW CHART



EDSIM51 SIMULATIONS

EdSim51DI - Version 2.1.32 | MINI PROJECT CODE.asm

The screenshot displays the EdSim51DI simulation environment. The top section shows the system clock at 11.0592 MHz and a frequency of 50000. The register window on the left lists various registers including R0-R7, B, ACC, PSW, IP, IE, PCON, DPH, DPL, and SP. The PC register is highlighted at 8051. The central code window shows assembly instructions such as `START:`, `MOV P2, #0FFH`, `CHECK_IR1:`, `MOV R7, P2`, `MOV A, R7`, `RRC A`, `JNC CLOSEGATE`, `SJMP CHECK_IR1`, `CHECK_IR2:`, `MOV R7, P2`, `MOV A, R7`, `RRC A`, `JNC RECHECK_IR1`, `SJMP CHECK_IR2`, `CLOSEGATE:`, `MOV P1, #00H`, `SETB P3.0`, `CLR P3.1`, and `ACALL DELAY`. The right panel shows a list of I/O devices and their status, including Keypad Column 0-2, Keypad Row 0-2, LED 0-7, SW 0-7, ADC RD/Comparator Output, ADC WR, Motor Sensor, Display-select Input 1, AND Gate Output/Display-select 0, ADC INTR, Motor Control Bit 1/Ext. UART Rx, and Motor Control Bit 0/Ext. UART Tx. The bottom section features a control panel with buttons for DI, LD, AND Gate Disabled, Key Bounce Disabled, Standard, Scope, DAC, and a status bar indicating 'Error! Function set not called'. A large 7-segment display shows '8.8.8.8'.

IR sensor 1 detected the train .so the gate is closed with the help of the motor.

2)

System Clock (MHz) 11.0592 50000 Update Freq.

SBUF

R/O	W/O	TH0	TL0	R7	0xFC	B	0x00
0x00	0x00	0x00	0x00	R6	0x00	ACC	0xFC
RxD	TxD	TMOD	0x00	R5	0xFC	PSW	0x00
0	0	TCOD	0x00	R4	0xFF	IP	0x00
SCON	0x00	TCOD	0x00	R3	0x00	IE	0x00
				R2	0x00	PCON	0x00
pins	bits	TH1	TL1	R1	0x00	DPH	0x00
0xFC	0xFC	P3	0x00	R0	0x00	DPL	0x00
0xFC	0xFF	P2				SP	0x07
0x00	0x00	P1					
0xFF	0xFF	P0					

PC 8051 0x0026 PSW 0 0 0 0 0 0 0 0

Data Memory

addr	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	00	00	00	00	FF	FC	00	FC	1D	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Copyright ©2005-2021 James Rogers Remove All Breakpoints

Time: 1s 160ms 926us - Instructions: 650008

```

CHECK_IR2:
##B MOV R7,P2
##D MOV A,R7
##E RRC A
##F RRC A
##1 JNC RECHECK_IR1
##12 SJMP CHECK_IR2
CLOSEGATE:
##14 MOV P1,##5H
##17 SETB P3.5
##19 CLR P3.1
##1B ACALL DELAY
##1D CLR P3.5
##1F CLR P3.1
##21 SJMP CHECK_IR2
RECHECK_IR1:
##23 MOV R5,P2
##25 MOV A,R5
##26 RRC A
##27 JNC RECHECK_IR1
  
```

P0.7 1 Display-select Decoder CS/DAC WR
P0.6 1 Keypad Column 2
P0.5 1 Keypad Column 1
P0.4 1 Keypad Column 0
P0.3 1 Keypad Row 3
P0.2 1 Keypad Row 2
P0.1 1 Keypad Row 1
P0.0 1 Keypad Row 0
P1.7 0 LED 7(Seg. dp)DAC DB7/LCD DB7
P1.6 0 LED 6(Seg. g)DAC DB6/LCD DB6
P1.5 0 LED 5(Seg. f)DAC DB5/LCD DB5
P1.4 0 LED 4(Seg. e)DAC DB4/LCD DB4
P1.3 0 LED 3(Seg. d)DAC DB3/LCD DB3
P1.2 0 LED 2(Seg. c)DAC DB2/LCD DB2
P1.1 0 LED 1(Seg. b)DAC DB1/LCD DB1
P1.0 0 LED 0(Seg. a)DAC DB0/LCD DB0
P2.7 1 SW 7(ADC DB7)
P2.6 1 SW 6(ADC DB6)
P2.5 1 SW 5(ADC DB5)
P2.4 1 SW 4(ADC DB4)
P2.3 1 SW 3(ADC DB3)
P2.2 1 SW 2(ADC DB2)
P2.1 0 SW 1(ADC DB1)
P2.0 0 SW 0(ADC DB0)
P3.7 1 ADC RD/Comparator Output
P3.6 1 ADC WR
P3.5 1 Motor Sensor
P3.4 1 Display-select Input 1
P3.3 1 AND Gate Output(Display-se...t 0
P3.2 1 ADC INTR
P3.1 0 Motor Control Bit 1/Ext. UART Rx
P3.0 0 Motor Control Bit 0/Ext. UART Tx

DI LD AND Gate Disabled Key Bounce Disabled Standard

U No Parity 8-bit UART @ 4800 Baud Rx Reset Tx Send

0.0 V output Scope DAC

Error Function set not called

11111111 ADC

MAX MIN Motor Enabled

3)

System Clock (MHz) 11.0592 50000 Update Freq.

SBUF

R/O	W/O	TH0	TL0	R7	0xFC	B	0x00
0x00	0x00	0x00	0x00	R6	0x00	ACC	0x7E
RxD	TxD	TMOD	0x00	R5	0xFD	PSW	0x01
0	0	TCOD	0x00	R4	0xFD	IP	0x00
SCON	0x00	TCOD	0x00	R3	0x00	IE	0x00
				R2	0x00	PCON	0x00
pins	bits	TH1	TL1	R1	0x00	DPH	0x00
0xFC	0xFC	P3	0x00	R0	0x00	DPL	0x00
0xFD	0xFF	P2				SP	0x07
0x00	0x00	P1					
0xFF	0xFF	P0					

PC 8051 0x0029 PSW 1 0 0 0 0 0 0 0

Data Memory

addr	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	00	00	00	00	FD	FD	00	FC	1D	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Copyright ©2005-2021 James Rogers Remove All Breakpoints

Time: 4s 177ms 839us - Instructions: 255118

```

CHECK_IR2:
##B MOV R7,P2
##D MOV A,R7
##E RRC A
##F RRC A
##1 JNC RECHECK_IR1
##12 SJMP CHECK_IR2
CLOSEGATE:
##14 MOV P1,##5H
##17 SETB P3.5
##19 CLR P3.1
##1B ACALL DELAY
##1D CLR P3.5
##1F CLR P3.1
##21 SJMP CHECK_IR2
RECHECK_IR1:
##23 MOV R5,P2
##25 MOV A,R5
##26 RRC A
##27 JNC RECHECK_IR1
  
```

P0.7 1 Display-select Decoder CS/DAC WR
P0.6 1 Keypad Column 2
P0.5 1 Keypad Column 1
P0.4 1 Keypad Column 0
P0.3 1 Keypad Row 3
P0.2 1 Keypad Row 2
P0.1 1 Keypad Row 1
P0.0 1 Keypad Row 0
P1.7 0 LED 7(Seg. dp)DAC DB7/LCD DB7
P1.6 0 LED 6(Seg. g)DAC DB6/LCD DB6
P1.5 0 LED 5(Seg. f)DAC DB5/LCD DB5
P1.4 0 LED 4(Seg. e)DAC DB4/LCD DB4
P1.3 0 LED 3(Seg. d)DAC DB3/LCD DB3
P1.2 0 LED 2(Seg. c)DAC DB2/LCD DB2
P1.1 0 LED 1(Seg. b)DAC DB1/LCD DB1
P1.0 0 LED 0(Seg. a)DAC DB0/LCD DB0
P2.7 1 SW 7(ADC DB7)
P2.6 1 SW 6(ADC DB6)
P2.5 1 SW 5(ADC DB5)
P2.4 1 SW 4(ADC DB4)
P2.3 1 SW 3(ADC DB3)
P2.2 1 SW 2(ADC DB2)
P2.1 0 SW 1(ADC DB1)
P2.0 0 SW 0(ADC DB0)
P3.7 1 ADC RD/Comparator Output
P3.6 1 ADC WR
P3.5 1 Motor Sensor
P3.4 1 Display-select Input 1
P3.3 1 AND Gate Output(Display-se...t 0
P3.2 1 ADC INTR
P3.1 0 Motor Control Bit 1/Ext. UART Rx
P3.0 0 Motor Control Bit 0/Ext. UART Tx

DI LD AND Gate Disabled Key Bounce Disabled Standard

U No Parity 8-bit UART @ 4800 Baud Rx Reset Tx Send

0.0 V output Scope DAC

Error Function set not called

11111111 ADC

MAX MIN Motor Enabled

2) IR SENSOR 2 has detected the train .the gate should not be opened now because we have to confirm that train has completely left.

3)ir sensor 1 is unblocked .

The screenshot displays the Proteus simulation environment for an 8051 microcontroller project. The top section shows the register window with various registers (R0-R7, ACC, PSW, IP, IE, PCON, DPH, DPL, SP) and their current values. The code editor in the center contains assembly code for the 8051, including labels like RECHECK_IR1, RECHECK_IR2, and OPENGATE, with instructions such as MOV, RRC, JNC, CLR, SETB, ACALL, and LJMP. The right side of the interface shows a pin list for the 8051, mapping pins to specific functions like Display-select Decoder CS, Keypad Column, Keypad Row, LED, SW, ADC, and Motor Control. The bottom section shows the hardware components, including a display, keypad, and motor, with a status bar indicating 'Error Function set not called'.

4) ir sensor 2 is unblocked ,we have ensured that train has completely left .so,the gate is opened.

INFERENCE:

Designed an automatic railway gate control system which automatically controls the gate on arrival and departure of train .

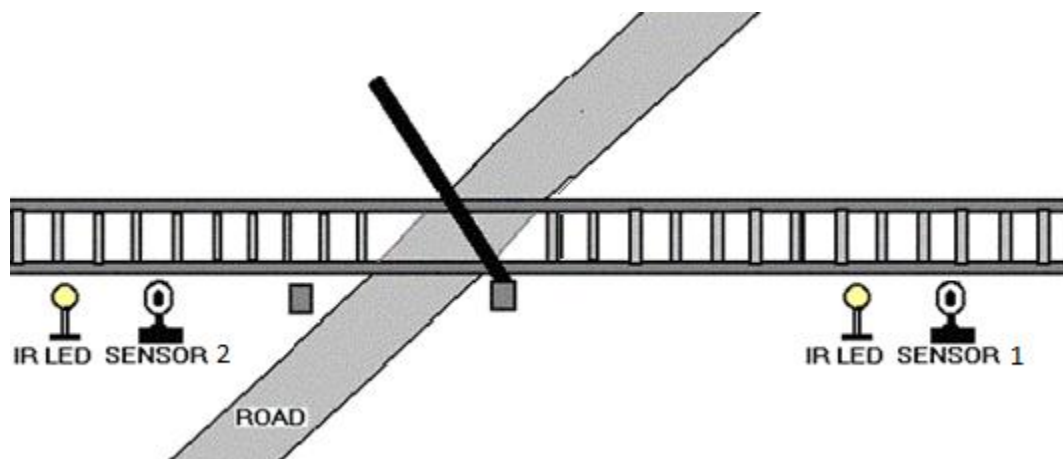
ADVANTAGES:

- 1.Reduces chances of human error
- 2.less time consuming

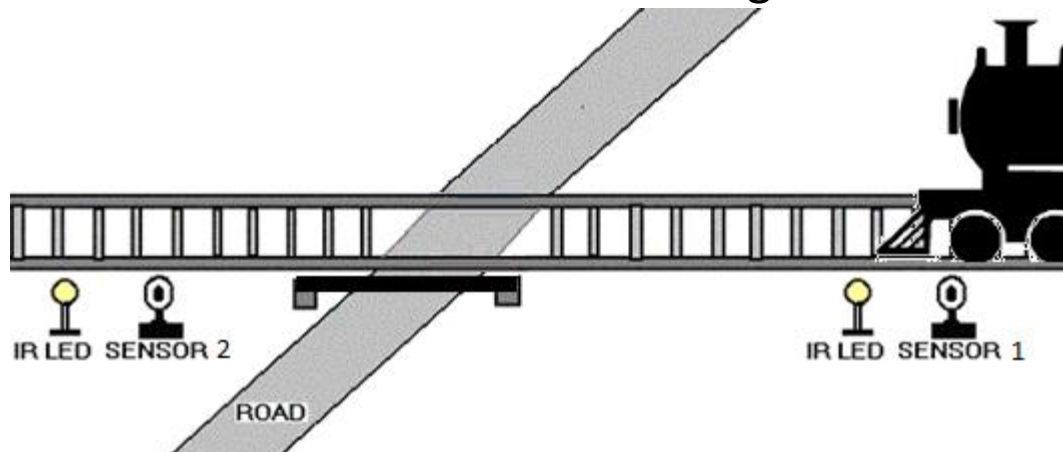
Limitations

- The system can be implemented more efficiently by incorporating more efficient sensor network.

- SCHEMATIC DIAGRAM



- If the sensor 1 detects the arrival of the train, microcontroller starts the motor with the help of motor driver in order to close the gate.



- The gate remains closed as the train passes the crossing.
- When the train crosses the gate and reaches second sensor, it detects the train and the microcontroller will open the gate.

